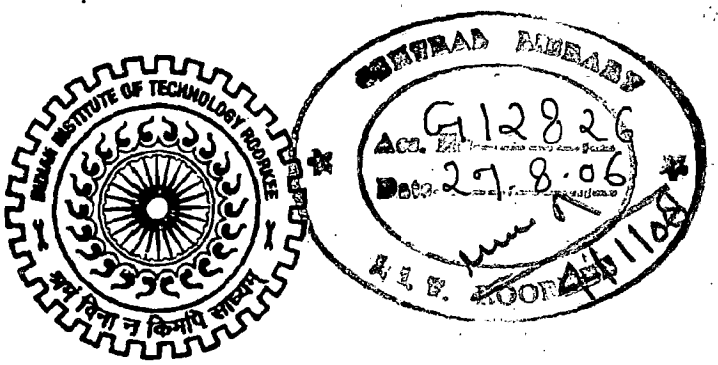
IMPLEMENTATION OF MPEG SYSTEM FOR IMAGE PROCESSING USING FPGA

A DISSERTATION

Submitted in partial fulfilment of the
requirements for the award of the degree
of
MASTER OF TECHNOLOGY
in
ELECTRICAL ENGINEERING
(With Specialization in Systems Engineering and Operations Research)

By

PRASHANT TAK



DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE-247 667 (INDIA)

JUNE, 2006

ID NO. MT-359/2006-54/PT-IG-MK

INDIAN INSTITUTE OF TECHNOLOGY ROORKEE-247667

CANDIDATE'S DECLARATION

I hereby declare that the work, which is being presented in this dissertation entitled **"Implementation of MPEG System for Image Processing Using FPGA"** in the partial fulfillment of the requirement for the award of the degree of **Master of Technology in Electrical Engineering** with specialization in **System Engineering and Operation Research,** submitted in the **Department of Electrical Engineering, Indian Institute of Technology Roorkee,** Roorkee, is an authentic record of my own work carried out during July 2005 to June 2006 under the supervision of **Dr. Indra Gupta,** Assistant Professor and **Prof. M. K. Vasantha,** Professor, Department of Electrical Engineering, IIT Roorkee, Roorkee.

I have not submitted the matter embodied in this dissertation for award of any other degree.

Date: **(Prashant Tak)**

Place: Roorkee

CERTIFICATE

This is to certify that the above statement made by the candidate is true to the best of my knowledge and belief.

Prof. M. K. Vasantha
Professor,
Department of Electrical Engineering,
Indian Institute of Technology Roorkee.
Roorkee – 247667.

Dr. Indra Gupta
Asstt. Professor,
Department of Electrical Engineering,
Indian Institute of Technology Roorkee.
Roorkee – 247667.

Acknowledgements

I would like to take this opportunity to express my deepest sense of gratitude to my supervisors ***Dr.*** **(Ms.) Indra Gupta** and **Prof. M. K. Vasantha**, *Department of Electrical Engineering, I.I.T. Roorkee, for their invaluable support, guidance and suggestions at various stages of this Dissertation. I feel privileged to be associated under them and it was a great pleasure in learning the practical aspects of digital design and verification under their aegis and guidance. I am very much thankful to them for giving me an opportunity to work on a topic which was very much interesting and challenging for me. I remember with great emotion, the constant encouragement and help extended to me by him that went even beyond the realm of academics.*

The department of Electrical Engineering of this institute provided me with all kinds of necessary facilities for carrying out my work. My sincere thanks are due to ***Prof. S. P. Gupta****, Head of Electrical Engineering department and* ***Prof. H. O. Gupta****, Ex-head of the department, for making the opportunities available all through. My sincere thanks go to all the faculty members of the department for the voluntary help, direct and indirect, extended to me during the course of the work.*

I am also thankful to ***Dr. D.K. Mehra, HOD,*** *Department of Electronics and Computers, IIT Roorkee, for his invaluable discussion and suggestions, regarding various issues related to field of communication and image processing.*

My sincere regards to staff and my friends at the Department who have directly and indirectly helped me in completing this Report. I am grateful to my hostel mates and my colleagues, for helpful and fruitful discussions and for the good time we spent together.

Last, but not least by any means, I wish to acknowledge my family members for giving me the moral strength and constant encouragement. The work could never reach its present status without their constant support and love.

Dated:

Place: Roorkee

Prashant Tak

M. Tech (System Engineering & Operations Research)

Department of Electrical Engineering, I. I. T. Roorkee.

Abstract

This Dissertation describes the methodology followed in the implementation of Motion Pictures Expert Group (MPEG) standard for multimedia image processing using VHDL to be implemented over Field Programmable Gate Array (FPGA). As we know MPEG is a popular multimedia image compression standard used in various multimedia applications ranging from data storage to transmission of live video, thus, requiring high speed processing.

Presently, implementation of the standard is mostly done using high level languages, thus requiring central processing unit (CPU) to perform the task of brute force calculation on a continuous stream of data. This makes the speed of scheduled task on a multipurpose processor to slow down. This dissertation proposes a solution to this problem by transferring the task of implementing various algorithms under MPEG standard to be performed on a standalone hardware and thus, setting CPU free of any overhead.

FPGA being a flexible device from design point of view makes it possible to implement various modifications in the design as the up graded versions of standard comes in and thus, proves to be an economic alternative to presently popular CPLD device. This dissertation gives a detail flow of algorithms stated in ISO recommended MPEG standard implemented in hardware compatible very high speed integrated circuit hardware descriptor language. Here, an attempt has been made to remove three major type of redundancies namely Psycho Visual Redundancy, Inter Pixel Redundancy and Coding Redundancy.

Content

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

Signal processing is one of the areas of wide research for decades. It has been widely used in areas ranging from data communication to biomedical engineering to acoustics to robotics to consumer electronics to instrumentation and many others. Signal processing hardware have wide requirement ranging from military systems to industrial applications.

Initially, signal processing was mainly done for on continuous signals which had its own limitations. But with the introduction of digital processors it became possible to implement different signal processing techniques on discrete samples of data. This lead to various advantages ranging from robustness to errors, mass storage, etc.

Signal processing done in two dimension is used for image processing. It finds wide application in the area of image compression, bio medical imaging, satellite imaging, aerial topography, etc. [1] As storage and processing of image requires large amount of space and computing power. For instance a 512 X 512 size of gray level image requires 256 Kbytes of memory. Thus, a need is felt to compress the image data so as to make it possible to transmit as well as store the large amount of data in a minimum memory space .This give rise to need for data compression but the main problem we face here is the deterioration in the quality of image.

So, we consider the redundancies that are inherently present in the image. They are listed as follows:

1. Within a single image or a single video frame, there exists significant correlation or redundancy among neighboring samples or pixels. This correlation is referred as spatial correlation or redundancy or inters pixel redundancy.[2]

2. Data obtained from multiple sensors such as in satellite imaging, stereophonic data transfer there may exist significant correlation or redundancy among samples obtained from various sensors. This correlation or redundancy is called spectral correlation or redundancy.

3. In the temporal data obtained from successive frames of video sequence, there is significant correlation or redundancy among pixels of successive video frames. This is referred to as temporal correlation or redundancy.

4. If all the pixel values are coded using same sized code, then it gives rise to coding redundancy.

5. Human eyes are more sensitive to brightness in the signal as well as to those frequencies which fall at the lower side of the spectrum. Thus, when whole lot of data in transmitted it gives rise to psychological redundancies.

Thus, due to the above mentioned reasons we go for compressing the image. In order to reconstruct the image or video data the process is reversed at the decoder. In compression systems, the term 'compression ratio' is used to characterize the compression capability of the system.

$$Compression\,Ratio = \frac{Source\ coder\ input\ data\ size}{Source\ coder\ output\ data\ size} \qquad \text{.....Eq'n 1.1}$$

For a still image, size could be the bits needed to represent the entire image. For video, size could refer to the bits needed to represent one frame of video, i.e., one second of video.

1.2 Classification of Compression Schemes [2]

The classification of compression schemes can be done in the following manner :

Lossless vs. Lossy compression:

A compression scheme in which reconstructed image obtained by compression and then decompression is similar to the original image is known as lossless compression. This type of compression scheme can only achieve a modest amount of compression. On the other hand, lossy schemes is one which is capable of achieving much higher compression at the rate of quality deterioration but under normal viewing conditions no visible loss is perceived. Some popular lossy compression schemes are differential pulse code modulation (DPCM), pulse code modulation (PCM), vector quantization (VQ), Transform and Subband coding. [3]

Predictive vs. Transform coding:

Predictive coding is a process in which already available information is used to predict the future values and the difference between them is coded. This is done in the image or spatial domain which makes it some what simple to implement and readily adaptable to local image characteristics. The DPCM is one particular example of predictive coding. Transform coding first transforms the image from its spatial domain representation to known transforms such as DCT, DWT, etc and then codes the transformed values (coefficients).

This method provides greater data compression compared to predictive methods as transforms use energy compaction properties to pack an entire image or a video frame into fewer transform coefficients. Most of the high frequency coefficients become insignificant after applying quantization ensuring less data to be transmitted. In predictive coding, the differences between the original image or video frame samples and the predicted ones remain significant even after applying quantization, thus, requiring more data to be transmitted compared to transform coding.

Predictive coding scheme is basically of two types, namely intra frame predictive coding in which prediction difference between pixels in the same frame is coded to

remove spatial redundancy in the frame and inter frame predictive coding in which difference between pixels in the successive frames is coded to remove temporal redundancy. This requires different motion estimation techniques to be employed.[1]

Subband Coding:

The fundamental concept followed in subband coding is to split the frequency band of a signal (image in our case) in various subbands. To code each subband, we use a coder and bit rate accurately matched to the statistics of the subband. Still image compression can produce a video compression to a figure of five whereas in moving picture this value can be further improved considering the fact that there is a large correlation between successive frames of images.

There are various standards for video compression. The two major standards followed are H.263 and its variants and the second one followed is Motion Pictures Expert Group (MPEG) and its variants. Both of these standards have various similarities and can be implemented on a single chip with proper resource planning but requires utmost care to be taken. The MPEG standard has fixed syntax for decoding algorithms but there is no such standard for encoders and change in buffer size, motion vector estimation, variation of scaling factor for quantization coefficient etc depends totally on the kind of application it has to be applied for and kind of constraints it has to be designed under.

1.3 Organization of the thesis:

This thesis is organized under following topics

Chapter 1 consist of general introduction to development of signal and image processing, causes of redundancy, different types of compression and introduction to the different sections of thesis.

Chapter 2 gives certain basic definition used in this thesis followed by different mathematical components applied in compression standards and finally a brief introduction to JPEG standard which forms the base of MPEG standard.

Chapter 3 gives a detail description of MPEG standard and its different variants. It also discusses about historical phase in the development of standard and various organizations involved in its development. It gives detail of different applications where MPEG standard is implemented and benefits obtained out of its implementation.

Chapter 4 gives a description of Lossy compression scheme applied under MPEG compression standard and its implementation flow over FPGA. This includes determining DCT of the input grey level pixel values and then quantizing them to zero the high frequency values.

Chapter 5 details about lossless coding schemes employed in order to further compress the coefficients by taking advantage of their statistical properties. In this chapter run length and Huffman coding schemes are discussed. This chapter also brief about the interface provided to download image into the memory, reading data from the memory and finally, displaying reconstructed image on to raster.

Chapter 6 gives device introduction and different features available on video application kit. It discusses some of the standard results obtained on Modelsim Simulator and gives a brief discussion of the results obtained.

Chapter 7 provides a summary of work followed by conclusion and finally, briefs about the future scope of work. It is followed by appendix which gives excerpts of standard tables followed under ISO recommendation for implementing the MPEG compression standard.

Appendix A gives excerpts of different tables employed for implementing ISO recommended standard.

Chapter 2

Image Coding and Compression Standards

In this chapter we will undergo a comprehensive study of different features and mathematical components involved in popularly known compression standards. But before we proceed on with the description, let us state certain standard definitions. [3],[4]

AC coefficient: Any DCT coefficient for which the frequency in one or both dimensions is non-zero.

B-field picture: A field structure B-Picture.

B-frame picture: A frame structure B-Picture.

B-picture; bi directionally predictive-coded picture: A picture that is coded using motion compensated prediction from past and/or future reference fields or frames.

Backward compatibility: A newer coding standard is backward compatible with an older coding standard if decoders designed to operate with the older coding standard are able to continue to operate by decoding all or part of a bit stream produced according to the newer coding standard.

Backward motion vector: A motion vector that is used for motion compensation from a

reference frame or reference field at a later time in display order.

Backward prediction: Prediction from the future reference frame (field).

Base layer: First, independently decodable layer of a scalable hierarchy

Bit stream; stream: An ordered series of bits that forms the coded representation of the data.

Bit rate: The rate at which the coded bit stream is delivered from the storage medium to the input of a decoder.

Block: An 8-row by 8-column matrix of samples, or 64 DCT coefficients (source, quantized or dequantised).

Bottom field: One of two fields that comprise a frame. Each line of a bottom field is spatially located immediately below the corresponding line of the top field.

Byte aligned: A bit in a coded bit stream is byte-aligned if its position is a multiple of 8-bits from the first bit in the stream.

Channel: A digital medium that stores or transports a bit stream constructed according to this specification.

Chrominance format: Defines the number of chrominance blocks in a macroblock.

Chrominance component: A matrix, block or single sample representing one of the two

color difference signals related to the primary colors in the manner defined in the bit stream. The symbols used for the chrominance signals are Cr and Cb.

Coded B-frame: A B-frame picture or a pair of B-field pictures.

Coded frame: A coded frame is a coded I-frame, a coded P-frame or a coded B-frame.

Coded I-frame: An I-frame picture or a pair of field pictures, where the first field picture is an I-picture and the second field picture is an I-picture or a P-picture.

Coded P-frame: A P-frame picture or a pair of P-field pictures.

Coded picture: A coded picture is made of a picture header, the optional extensions

immediately following it, and the following picture data. A coded picture may be a coded

frame or a coded field.

Coded video bit stream: A coded representation of a series of one or more pictures as defined in the specification.

Coded order: The order in which the pictures are transmitted and decoded. This order is not necessarily the same as the display order.

Coded representation: A data element as represented in its encoded form.

Coding parameters: The set of user-definable parameters that characterizes a coded video bit stream. Bit streams are characterized by coding parameters. Decoders are characterized by the bit streams that they are capable of decoding.

Component: A matrix, block or single sample from one of the three matrices (luminance and two chrominance) that make up a picture.

Compression: Reduction in the number of bits used to represent an item of data.

Constant bit rate coded video: A coded video bit stream with a constant bit rate.

Constant bit rate: Operation where the bit rate is constant from start to finish of the coded bit stream.

Data element: An item of data as represented before encoding and after decoding.

Data partitioning: A method for dividing a bit stream into two separate bit streams for error resilience purposes. the two bit streams have to be recombined before decoding.

DC coefficient: The DCT coefficient for which the frequency is zero in both dimensions.

DCT coefficient: The amplitude of a specific cosine basis function.

Decoder input buffer: The first-in first-out (FIFO) buffer specified in the video buffering verifier.

Decoder: An embodiment of a decoding process.

Decoding (process): The process is defined as one that reads an input coded bit stream and produces decoded pictures samples.

Dequantization: The process of rescaling the quantized DCT coefficients after their

representation in the bit stream has been decoded and before they are presented to the inverse DCT.

Digital storage media; DSM: A digital storage or transmission device or system.

Display aspect ratio: The ratio height/width (in SI units) of the intended display.

Frame: A frame consists of three rectangular matrices of integers; a luminance matrix (Y), and two chrominance matrices (Cb and Cr).

Field: A field consists of every other line of samples in the three rectangular matrices of integers representing a frame. A frame is the union of a top field and a bottom field. The top field is the field that contains the top-most line of each of the three matrices. The bottom field is the other one.

Picture types: There are three types of pictures that use different coding methods.

An **Intra-coded (I) picture** is coded using information only from itself.
A **Predictive-coded (P) picture** is a picture which is coded using motion compensated prediction from a past reference frame or past reference field.
A **Bi directionally predictive-coded (B) picture** is a picture which is coded using motion compensated prediction from a past and/or future reference frame(s).
Macroblock: A **macroblock** contains a section of the luminance component and the spatially corresponding chrominance components. The term macroblock can either refer to source and decoded data or to the corresponding coded data elements. A skipped macroblock is one for which no information is transmitted .There are three chrominance formats for a macroblock, namely, 4:2:0, 4:2:2 and 4:4:4 formats. The orders of blocks in a macroblock shall be different for each different chrominance format and are illustrated below:

4:2:0 Macroblock consists of 6 blocks. This structure holds 4 Y, 1 Cb and 1 Cr Blocks.
4:2:2 Macroblock consists of 8 blocks. This structure holds 4 Y, 2 Cb and 2 Cr Blocks.
4:4:4 Macroblock consists of 12 blocks. This structure holds 4 Y, 4 Cb and 4 Cr Blocks.

Block: The term "**block**" refers either to source and reconstructed data or to the DCT coefficients or to the corresponding coded data elements. When "block" refers to source and reconstructed data it refers to an orthogonal section of a luminance or chrominance component with the same number of lines and samples. There are 8 lines and 8 samples in the block.
Slice: A consecutive series of macro blocks which are all located in the same horizontal row of macro blocks.
SNR scalability: A type of scalability where the enhancement layer (s) contains only coded refinement data for the DCT coefficients of the lower layer.
Run: The number of zero coefficients preceding a non-zero coefficient, in the scan order.

The absolute value of the non-zero coefficient is called "level".

Scalability: Scalability is the ability of a decoder to decode an ordered set of bit streams to produce a reconstructed sequence. Moreover, useful video is output when subsets are
decoded. The minimum subset that can thus be decoded is the first bit stream in the set
which is called the base layer. Each of the other bit streams in the set is called an enhancement layer. When addressing a specific enhancement layer, "lower layer" refer to
the bit stream which precedes the enhancement layer.

Now, as we are through with these basic definitions we will discuss some of the basis mathematical processes that form the backbone of the various compression standards.
The first to discuss is the Discrete Cosine Transform

2.1 Discrete Cosine Transform (DCT) Based Coding Scheme [1],[2],[4],[5]

Discrete cosine transform (DCT) transforms the image information from spatial domain to frequency domain to be represented in a more compact form. Its stochastic properties are similar to Fourier transform and consider the input image, audio or video signal to be a time invariant or stationary signal. Let us first discuss the fundamentals of Fourier transform, which forms the basis of DCT.

Fourier Transform (FT)

Fourier Transform (FT) is a reversible transform which allows going reverse and forward between the spatial or time domain and the transformed signals. However, only either of them is available at any given time. This means no frequency information is available in the time-domain signal and vice versa.

FT gives the frequency information of the signal but it does not tell us when in time these frequency components exist. This information is not required when the signal is stationary, that is, signal whose frequency content does not change in time. In

that case, one does not need to know at what times frequency components exist since all frequency components exist at all times.[2],[5]

The FT and inverse FT for a continuous signal are defined as:

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-2j\pi ft} dt \qquad \text{.....Eq'n 2.1}$$

$$x(t) = \int_{-\infty}^{\infty} X(f) e^{2j\pi ft} df \qquad \text{.....Eq'n 2.2}$$

FT can be used for non-stationary signals and are used if we are only interested in what spectral components exist in the signal, but not interested where these occur. If the information about spectral components is needed then Fourier transform is not the right transform to use.

In order to obtain discrete Fourier transform (DFT), we simply replace the integration in FT's mathematical expression by summation and calculate it over finite samples.

The k^{th} DFT coefficient of a length N sequence or samples {x(n)} is defined as :

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn} \quad , k=0,\ldots\ldots\ldots, N-1 \qquad \text{.....Eq'n 2.3}$$

Where, $W_N = e^{-j2\Pi/N} = \cos(2\pi/N) - j \sin(2\pi/N)$ is the Nth root of unity. The original

sequence {x(n)} can be retrieved by the inverse discrete Fourier transform (IDFT) is

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{-kn} \quad , n=0,\ldots\ldots\ldots, N-1 \qquad \text{.....Eq'n 2.4}$$

Definition and Properties of DCT

The forward N-point one-dimensional DCT and inverse DCT can be defined as follows:

Forward N-point DCT = X(k) = $\frac{2}{N}$ c_k $\sum_{n=0}^{N-1} x(n)\cos\left[\frac{(2n+1)k\pi}{2N}\right]$, k= 0,1,.....,N-1

Inverse N-point DCT = x(n) = $\frac{2}{N}\sum_{k=0}^{N-1} c_k X(k)\cos\left[\frac{(2n+1)k\pi}{2N}\right]$, n= 0,1,.....,N-1

where $c_k = \begin{cases} 1/\sqrt{2}, k = 0 \\ 1, k \neq 0 \end{cases}$ .....Eq'n 2.5

In case of image and video compression standards such as baseline JPEG and MPEG, 8-point DCT and IDCT are used. The image or each motion video frame of size NXN pixels is divided into two-dimensional non-overlapping blocks often called sub-images or basis functions of size 8x8 (having 64 pixels each) and 2-D DCT is applied on the encoder side, while on the decoding side 2-D IDCT is applied to recover the original data.

The 8-point 2-D DCT and IDCT to generate 8x8 data matrices are calculated as:

2-D DCT = $X_{k,l} = \frac{c(k)c(l)}{4} \sum_{m=0}^{7}\sum_{n=0}^{7} x_{m,n} \cos\left(\frac{(2m+1)k\pi}{16}\right)\cos\left(\frac{(2n+1)l\pi}{16}\right)$

where k,l = 0,1,.....,7

2-D IDCT= $x_{m,n} = \sum_{k=0}^{7}\sum_{l=0}^{7} \frac{c(k)c(l)}{4} X_{k,l}\cos\left(\frac{(2m+1)k\pi}{16}\right)\cos\left(\frac{(2n+1)l\pi}{16}\right)$

where m,n = 0,1,.....,7

and c(k),c(l) = $\begin{cases} 1/\sqrt{2}, k \,\&\, l = 0 \\ 1, otherwise \end{cases}$ .....Eq'n 2.6

The performance of DCT is similar to the statistically optimal KLT because of its nice decorrelation and energy compaction properties. DCT is data independent and many fast algorithms exist for its fast calculation so it is extensively used in multimedia compression standards.

2.2 Subband Coding

In this compression method the signal is divided into multiple frequency bands in order to take advantage of statistical characteristics and visual significance of each band. Thus, this can be considered as a super set of transform coding discussed above in section 2.1. In this coding scheme filters are employed to separate different frequency components in the form of bands which are then separately encoded, multiplexed and transmitted. As this method is very expensive to implement providing almost same advantages as being provided by DCT transform coding so they were not practically used in the past but with improvement in hardware the filters required for the process of band separation which requires maximum computation time are being implemented successfully.

2.3 Entropy Coding

Entropy coding performs the function of allocating codes to data according to its statistical characteristic without increasing the overall entropy of the signal. Most commonly used entropy coding techniques are Run length coding and Huffman coding.[1], [2],[3] The ISO standard given in reference [3] gives a detail description of the standard applied on data.

2.4 Vector Quantization

As we know that in scalar quantization sample values are separately quantized thus causing inter sample redundancy. To remove this redundancy we take a group of samples and quantize them as a single vector. For this the image is partitioned into N pixel blocks which are input to a Vector quantizer which then generates an index from a pre defined LUT (Look Up Table) accordingly and send it to the receiver side. But there are some limitation to this, primarily for better performance we require value of N to be large but this is limited by memory requirement and computation complexity. Secondly, LUT may not be able to make a perfect match.

2.5 Quality measures of Image Coding

In order to measure the quality of the image or video data at the output of the decoder, mean square error (MSE) and peak to signal to noise ratio (PSNR) ratio are often used. The MSE is often called Quantization error variance σ_q^2. The MSE between the original image ***f*** and the reconstructed image ***g*** at decoder is defined as: [2], [18]

$$\mathbf{MSE} = \sigma_q^2 = \frac{1}{N} \sum_{j,k} (f[j,k] - g[j,k])^2 \qquad \text{..Eq'n 2.7}$$

Where the sum over j, k denotes the sum over all pixels in the image and N is the number of pixels in each image. The PSNR between two images having 8 bits per pixel or sample in terms of decibels (dB's) is given by:

$$\mathbf{PSNR} = 10 \log_{10} \left(\frac{255^2}{MSE} \right) \qquad \text{.....Eq'n 2.8}$$

Generally when PSNR is 40 dB or greater, then the original and the reconstructed images are virtually indistinguishable by human observers.

Signal to noise ratio (SNR) is also a measure, but it is mostly used in telecommunications. SNR for an image in terms of decibels (dB's) is given as:

$$\mathbf{SNR} = 10 \log_{10} \frac{\text{Encoder input image energy or variance}}{\text{Noise energy or variance}} \qquad \text{.....Eq'n 2.9}$$

2.6 Image Compression/Decompression in Joint Picture Experts Group (JPEG)

JPEG is one of the first digital compression standard for still images in both monochrome and color. Baseline JPEG is the simplest of all JPEG implementations based upon sequential processing mode (image is scanned from left to right and top to bottom) such that it uses 8-bit sample or pixel inputs for DCT based compression of 2-D 8 x 8 non-overlapping blocks in an image, followed by quantization of the DCT coefficients and entropy coding of the result.[1],[2],[7]

RGB color images are converted into a luminance component Y and two chrominance components U and V before undergoing process of compression. The luminance component contains the shades of gray and is a monochrome image. Two chrominance components together contain the color information. Encoding and decoding operations in the baseline JPEG are performed for luminance and chrominance components. [5]

The source image is first partitioned into non-overlapping 2-D blocks of 8x8, which are processed sequentially in a raster scan fashion. The pixels in each block are level shifted by subtracting a value of 128. Since we use the pixel values of range 0 to 255 (8-bit unsigned integer), applying DCT will generate AC coefficients in the range −1023 to +1023 and these may be represented by an 11-bit signed integer. [2]

The DC coefficient generated will be in the range 0 to 2040, and this unsigned 11-bit. The value of 128 is subtracted from each pixel prior to DCT. This subtraction has no effect on the AC coefficients but shifts the DC coefficients into the same range so that all can be represented as 11-bit signed integers.

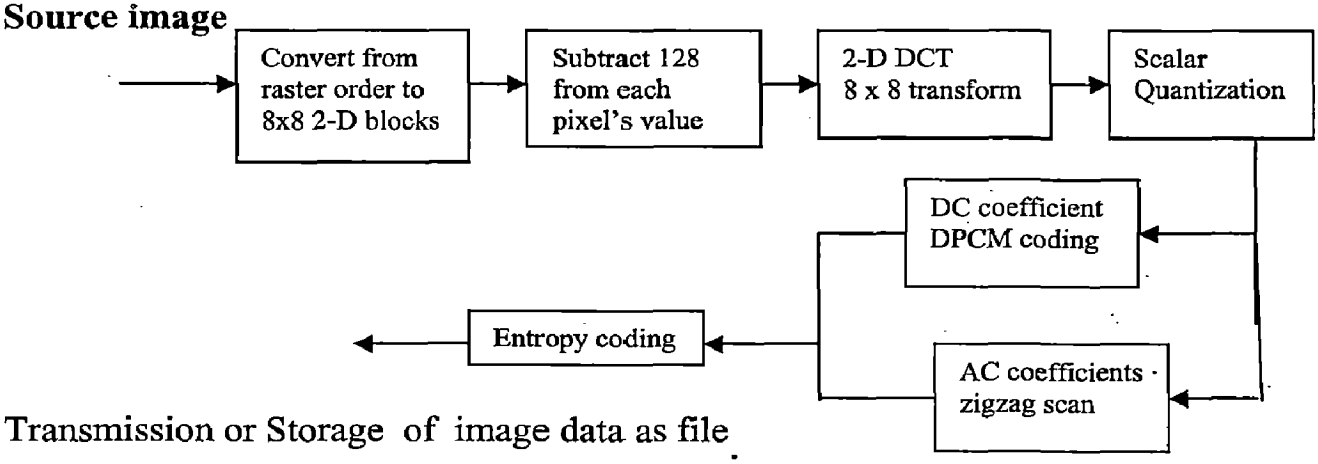


Figure2.2(a): Baseline JPEG encoder for image compression

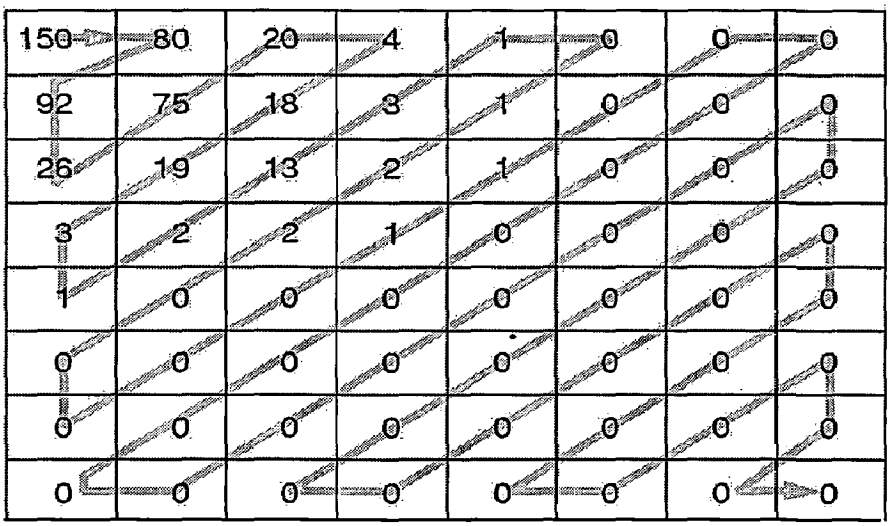


Figure 2.2(b): Zigzag scan of DCT coefficients at encoder

On each of these 2-D block of 8x8, DCT is in order to generate an array of 2-D transformed coefficients. The coefficient with lowest spatial frequency and highest variance value is considered as DC coefficient and it is proportional to the average brightness of the whole 2-D 8x8 spatial block. The rest are considered AC coefficients. Ideally, DCT introduces no loss to the source samples but it merely transforms them to a domain in which they can be more efficiently encoded.

The 2-D DCT array of coefficients are quantized using uniform scalar quantizer, ensuring that each coefficient is quantized individually and independently. The quantization is based on the properties of human visual system which is less sensitive to higher frequency coefficients and more to low frequency coefficients. The quantization table can be scaled to provide a variety of compression levels to achieve desired bit rate and quality of the images. The quantization of the AC coefficients produces many zeros, especially at higher frequencies. The rounding process used in quantization is lossy but it gives lot of compression.

This 2-D DCT array of quantized coefficients is reordered using a zigzag pattern to form a 1-D sequence. This rearranges the coefficients in approximately decreasing order of their average energies and increasing spatial frequencies so as to create large runs of zero values. The DC coefficient is separated from the AC coefficients and the sequence of DC coefficients is first coded using DPCM. The final step at the encoder is to use entropy coding such as Huffman coding on both AC and DC

(DPCM coded before) coefficients to achieve extra compression and making them more immune to error.

On the decoder side once the encoded bit stream is entropy decoded, the 2-D array of quantized DCT coefficients is recovered using de-zigzag, reordered and each coefficient is inverse quantized. The resulting array of coefficients is transformed back using 2-D inverse DCT and adding 128 to each pixel value to yield an approximation of the original 8x8 block or sub-image. The same quantization and entropy coding tables are used at the encoder and decoder side.

Each chrominance component of the color image is encoded in the same way as luminance except that it is down sampled (decimated) by the factor of 2 or 4 in both horizontal and vertical directions prior to DCT. At the decoder side, the reconstructed chrominance component is bilinear.

Thus, in this way compression is achieved for still images. In the next chapter discussion is done on how compression is achieved for a motion picture used in various multimedia applications.

Chapter 3
THE MPEG STANDARDS

To meet the need of TV broadcasting and multimedia imaging the Moving Picture Experts Group (MPEG) was formed for developing coding standards. MPEG-1 and MPEG-2 video coding standards have attracted much attention with an increasing number of very large scale integration and software implementations of these standards becoming feasible. We review the basic concepts and techniques that are relevant in the context of the MPEG video-compression standards and outline MPEG-1 and MPEG-2 video-coding algorithms. [6]

3.1 Development of MPEG Standard

Modern image and video compression techniques offer the possibility to store or transmit the vast amount of data necessary to represent digital images and video in an efficient and robust way. From the beginning of the 1980s on, a number of international video and audio standardization activities started within the International Telephone Consultive Committee (CCITT), followed by the International Radio Consultive Committee (CCIR), and the International Organization of Standardization/International Electro technical Commission (ISO/IEC).

MPEG was established in 1988 in the framework of the Joint ISO/IEC Technical Committee (JTC 1) on Information Technology with the mandate to develop standards for coded representation of moving pictures, associated audio, and their combination when used for storage and retrieval on digital storage media with a bit rate at up to about 1.5 M bit /s. In 1992 MPEG 1 standard was released.

The MPEG-2 standard was released in 1994. Emerging applications, such as digital cable TV distribution, networked database services via asynchronous transfer mode

(ATM), digital video tape recorder (VTR) applications, and satellite and terrestrial digital broadcasting distribution, were seen to benefit from the increased quality expected to result from the emerging MPEG-2 standard.

One key factor for the success of the MPEG standards is the generic structure supporting a wide range of applications and applications-specific parameters. To support the wide range of applications profiles, a diversity of input parameters including flexible picture size and frame rate can be specified by the user. Another important factor is the fact that the MPEG group only standardized the decoder structures and the bit-stream formats.

The MPEG group officially initiated a new MPEG-4 standardization phase in 1994 with the mandate to standardize algorithms and tools for coding and flexible representation of audio-visual data to meet the challenges of future multimedia applications and applications environments. In particular, MPEG-4 addresses the need for universal accessibility and robustness in error-prone environments, high interactive functionality, coding of natural and synthetic data, as well as high compression efficiency. Bit rates targeted for the MPEG-4 video standard are between 5-64 kbits/s for mobile or public switched telephone network (PSTN) video applications and up to 4 Mbits/s for TV/film applications.

3.2 Fundamentals of MPEG Video Compression Algorithms

Video sequences contain a significant amount of statistical and subjective redundancy within and between frames. The main goal of video source coding is the bit-rate reduction for storage and transmission by exploring both statistical and subjective redundancies and to encode a minimum amount of information using entropy coding techniques. Main aim here is to balance between compression quality and design complexity.[8]

MPEG mainly constitute lossless and lossy compression. The aim of lossless coding is to reduce image or video data for storage and transmission while retaining the quality of the original images the decoded image quality is required to be

identical to the image quality prior to encoding. In contrast, the aim of "lossy" coding techniques and this is relevant to the applications envisioned by MPEG-1, MPEG-2, and MPEG-4 video standards is to meet a given target bit rate for storage and transmission.

Important applications comprise transmission of video over communications channels with constrained bandwidth and the efficient storage of video. In these applications high video compression is achieved by degrading the video quality and reconstructed images. The smaller the target bit rate of the channel the higher the necessary compression of the video data and usually the more coding artifacts become visible.

The MPEG digital video-coding techniques are statistical in nature. Video sequences usually contain statistical redundancies in both temporal and spatial directions. The basic statistical property upon which MPEG compression techniques depends is interpel correlation, including the assumption of simple, correlated translatory motion between consecutive frames. The temporal correlation between pels in nearby frames is small or even vanishes the video scene then assembles a collection of uncorrelated still images. In the case intra frame coding techniques are employed to explore spatial correlation to achieve efficient data compression.

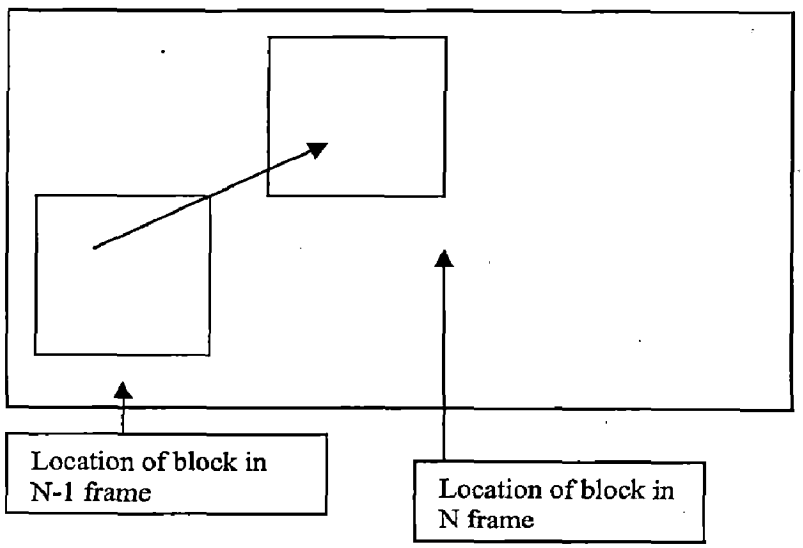


Figure 3.1 Temporal motion estimation in MPEG [6]

In MPEG video-coding schemes an adaptive combination of both temporal motion-compensated prediction followed by transform coding of the remaining spatial information is used to achieve high data compression (hybrid DPCM/DCT coding of video).

3.3 Sub sampling and Interpolation

The basic concept of sub sampling is to reduce the dimension of the input video (horizontal dimension and/or vertical dimension) and thus the number of pels to be coded prior to the encoding process. In case of some applications video is also subsampled in the temporal direction to reduce frame rate prior to encoding at the receiver images are interpolated for display. This technique makes use of specific physiological characteristics of the human eye and thus removes subjective redundancy contained in the video data-i.e., the human eye is more sensitive to changes in brightness than to chromaticity changes.

Therefore, the MPEG coding schemes first divide the images into YUV components (one luminance and two chrominance components). Next the chrominance components are sub sampled relative to the luminance component with a Y:U:V ratio specific to particular applications (i.e., with the MPEG2 standard a ratio of a 4:1:1 or 4:2:2 issued).

3.4 Motion-Compensated Prediction

It is a powerful tool to reduce temporal redundancies between frames. It is used extensively in MPEG-1 and MPEG-2 video-coding standards as a prediction technique for temporal DPCM coding. Motion compensation is based on the estimation of motion between video frames, that is if the elements in a video scene are approximately spatially displaced, the motion between frames can be described by a limited number of motion parameters (i.e. by motion vectors for translatory motion of pels). For example, the best prediction of an actual pel is given by a motion-compensated prediction pel from a previously coded frame. Normally both

prediction error and motion vectors are transmitted to the receiver. As the spatial correlation between motion vectors is often high, it is can be assumed that one motion vector is representative for the motion of a "block" of adjacent pels. So, the images are usually separated into disjoint blocks of pels (i.e. 16 x 16 pels in both MPEG-1 and MPEG-2 standards) and only one motion vector is estimated, coded, and transmitted for each block (as shown in Fig 3.1).

3.5 The MPEG-1 Standard

MPEG-1 was primarily targeted for multimedia CDROM applications, requiring additional functionality supported by both encoder and decoder. The video-compression technique developed by MPEG-1 covers many applications. To support the wide range of applications, a diversity of input parameters including flexible picture size and frame rate can be specified. MPEG has recommended a constraint parameter *set:* every MPEG- 1 compatible decoder must be able *to* support at least video-source parameters up to TV size: including a minimum number of 720 pixels per line, a minimum number of 576 lines per picture, a minimum frame rate of 30 frames per second, and a minimum bit rate of 1.86 Mbits/s.[9],[10]

3.6 The Basic MPEG- 1 Inter frame Coding Scheme

The MPEG-1 coding algorithm encodes the first frame in a video sequence in intraframe coding mode (I-picture). Each subsequent frame is coded using interframe prediction (P-pictures), and only data from the nearest previously coded I- or P-frame is used for the prediction. The algorithm processes the frames of a block-based video sequence. Each color input frame in a video sequence is partitioned into non overlapping "macro-block'" structure. Each macro-block contains blocks of data from both luminance and co-sited chrominance bands, there are four luminance blocks and two chrominance blocks (U, V), each with a size of 8

x 8 pels. For the Intra frame the process for encoding frame is similar to that adopted for JPEG standard. It is observed with reference to Fig. 3.2 the DCT coefficients most likely to appear are concentrated around the DC coefficient with decreasing importance.

For coding P-pictures, the previously I- or P-picture frame N- 1 is stored in a frame store (FS) in both the encoder and decoder. Motion compensation is performed on a macro-block basis, only one motion vector is estimated between frame *N* and frame *N* - 1 for a particular macro-block to be encoded. These motion vectors are coded and then transmitted to the receiver

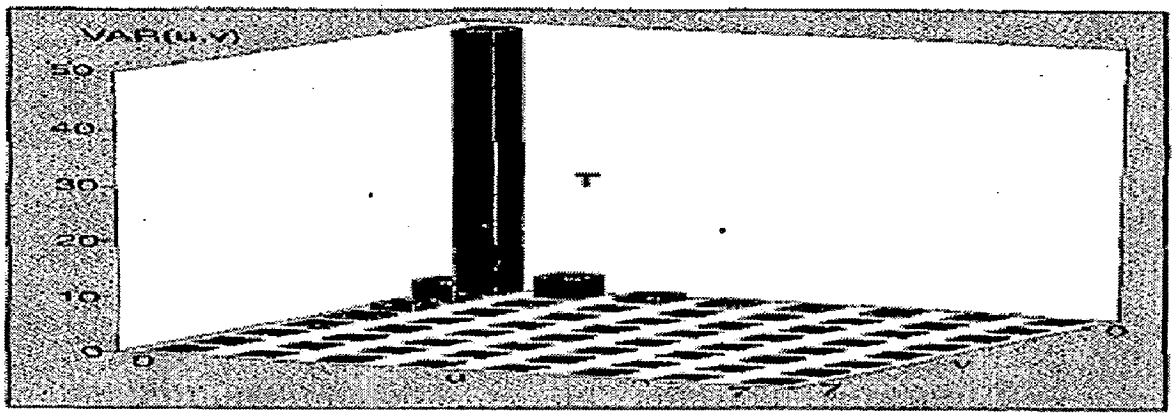


Figure 3.2 Energy variance at various frequency post DCT [6]

The motion compensated prediction error is calculated by subtracting each pel in a macro-block with its motion-shifted counterpart in the previous frame. A video buffer (VB) is needed to ensure that a constant target bit-rate output is produced by the encoder. The quantization step size can be adjusted for each macro-block in a frame to achieve the given target bit rate and also to avoid buffer overflow and underflow. The decoder uses the reverse process to reproduce a macro-block of frame

After decoding the variable-length words are contained in the video buffer. The pixel values of the prediction error are then reconstructed. The motion-compensated

pixels from the previous frame, N - 1, contained in the frame store are added to the prediction error so as to recover the particular macro-block.

3.7 Conditional Replenishment

The macro-block information at the decoder can be updated if the content of the macro-block has changed in comparison to the content of the same macro-block in the previous frame (conditional macro-block replenishment). The key for efficient coding of video sequences at lower bit rates is the selection of appropriate prediction modes to achieve conditional replenishment. The MPEG standard distinguishes mainly between three different macro-block coding types (MB types):

1. SkippedMB: Prediction from previous frame with zero motion vectors. No information about the macro-block is coded nor transmitted to *the* receiver.
2. Inter AB: Motion-compensated prediction from the previous frame is used. The MB type, the MB address and, if required, the motion vector, the DCT coefficients, and quantization step size are transmitted.
3. Intra MB: No prediction is used from the previous frame (intraframe prediction only). Only the MB type, the MB address, and the DCT coefficients and quantization step size are transmitted to the receiver.

3.8 Rate Control

In the MEPG-1 encoding algorithms there is also a possibility to tailor the bit rate (and thus the quality of the reconstructed video) to specific applications requirements by adjusting the quantizer step size for quantizing the DCT coefficients. Coarse quantization of the DCT coefficients enables the storage or transmission of video with high compression ratios, but, depending on the level of quantization, may result in significant coding artifacts. The MPEG-1 standard allows the encoder to select different quantizer values for each coded macro-block, which enables a high degree of flexibility to allocate bits in images where it is

needed to improve image quality. Furthermore, it also allows the generation of both constant and variable bit rates for storage or real-time transmission of the compressed video.

3.9 Specific Storage Media Functionalities

The MPEG-1 video-compression algorithm also supports important functionalities such as random access and FF and FR playback functionalities for accessing video from storage media. To incorporate the requirements for storage media and to further explore the significant advantages of motion compensation and motion interpolation, the concept of B-pictures (bi directional predicted/bi-directional interpolated pictures) was introduced by MPEG-1.

Intra pictures (I-pictures) are coded without reference to other pictures contained in the video sequence, I-pictures allow access points for random access and FF/FR functionality in the bit stream but achieve only low compression.

Since, P-pictures are usually used as reference for prediction for future or past frames, they provide no suitable access points for random access functionality or editability. Bi-directional predicted/interpolated pictures (B-picture$) require both past and future frames as references. To achieve high compression, motion compensation can be employed based on the nearest past and future P-pictures or I-pictures. B-pictures themselves are never used as references.

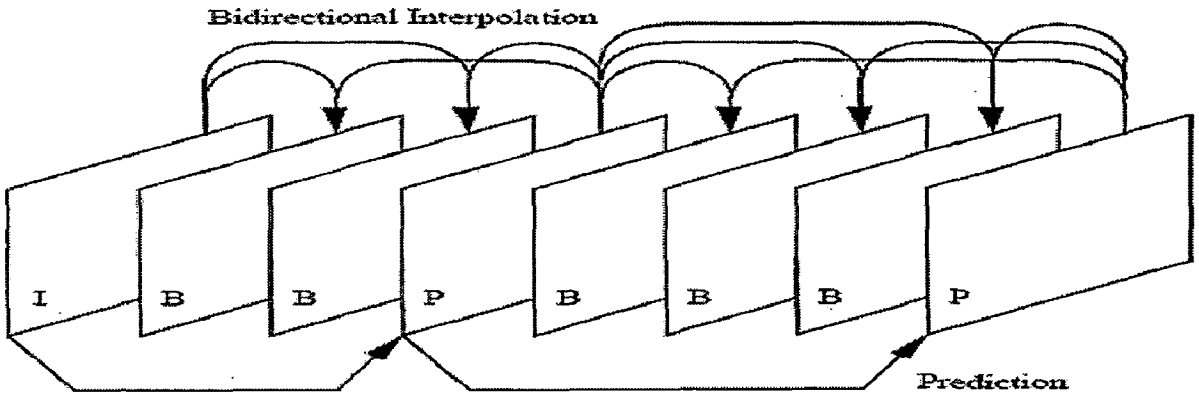


Figure 3.3 Frame transmission sequence in MPEG[4]

The picture types can be arranged in a video sequence with a high degree of flexibility to suit diverse applications requirements. As a general rule, a video sequence coded using I-pictures only (I I I I I I . . .) allows the highest degree of random access, FF/FR, and editability, but achieves only low compression. A sequence coded with a regular I-picture update and no B-pictures (i.e., I P P P P P P I P P P P . . .) achieves moderate compression and a certain degree of random access and FF/FR functionality. Incorporation of all the pictures types, as shown in Figure3.3 [4] for example, (I B B P B B P B B I B B P. ..), may achieve high compression and reasonable random access and FF/FR functionality but also increases the coding delay significantly. 'This delay may not be tolerable for applications such as video telephony or video conferencing.

3.10 Coding of Interlaced Video Sources

The standard video input format for MPEG-1 is non-interlaced. However, coding of interlaced color television with both 52s and 625 lines at 29.97 and 2s frames per second, respectively, is an important application for the MPEG-1 standard. Based on the conversion of the interlaced source to a progressive intermediate format, a suggestion to code Rec.601 digital color television signals has been made by MPEG- 1. In essence, only one horizontally sub-sampled field of each interlaced video input frame is encoded, i.e., the sub-sampled top field. At the receiver the even field is predicted from the decoded and horizontally interpolated odd field for display.

3.11 The MPEG-2 Standard

MPEG-2 provided a video coding solution for applications not originally covered or envisaged by MPEG-1 standard. MPEG-2 provides video quality not lower than NTSC/PAL and up to CCIR 601 quality. The standard aims to facilitate the bit-stream interchange among different applications and transmission and storage media. Basically, MPEG-2 can be seen as a superset of the MPEG-1 coding

standard and was designed to be backward compatible to MPEG-l. New coding features were added by MPEG-2 to achieve sufficient functionality and quality, thus prediction modes were developed to support efficient coding of interlaced video. In addition, scalable video coding extensions were introduced to provide additional functionality, such as embedded coding of digital TV and HDTV, and graceful quality degradation in the presence of transmission errors. MPEG-2 has introduced the concept of pro files' and "Levels" to stipulate conformance between equipment not supporting the full implementation.

Profiles and Levels provide means for defining subsets of the syntax and, thus, the decoder capabilities required to decode a particular bit stream. A Level specifies the range of the parameters that are supported by the implementation (i.e., image size, frame rate, and bit rates). The MPEG-2 core algorithm at MAIN Profile features non scalable coding of both progressive and interlaced video sources. It is expected that most MPEG-2 implementations will at least conform to the MAIN Profile at MAIN Level, which supports non scalable coding of digital video with approximately digital TV parameters, a maximum sample density of 720 samples per line and 576 lines per frame, a maximum frame rate of 30 frames per second, and a maximum bit rate of 15 Mbit/s.

3.12 MPEG-2 Non scalable Coding Modes

The MPEG-2 algorithm defined in the MAIN Profile is an extension of the MPEG-1 coding scheme to accommodate coding of interlaced video retaining the full range of functionality provided by MPEG- 1. The MPEG-2 coding algorithm is also based on the general hybrid DCT/DPCM coding scheme, incorporating a macro-block structure, motion compensation, and coding modes for conditional replenishment of macro-blocks. The concept of I-pictures, P-pictures, and B-pictures is fully retained in MPEG-2 to achieve efficient motion prediction and to assist random access functionality.

The algorithm defined with the MPEG-2 SIMPLE Profile is basically identical with the one in the MAIN Profile, except that no B-picture prediction modes are allowed at the encoder. Thus, the additional implementation complexity and the additional frame stores necessary for the decoding of B-pictures are not required for MPEG-2 decoders only conforming to the SIMPLE Profile.

3.13 Field and Frame Pictures

MPEG-2 has introduced the concept of frame pictures and field pictures along with particular field prediction and frame prediction modes to accommodate coding of progressive and interlaced video. For interlaced sequences it is assumed that the coder input consists of a series of odd (top) and even (bottom) fields that are separated in time by a field period. Two fields of a frame may be coded separately. In this case each field is separated into adjacent non overlapping macro blocks and the DCT is applied on a field basis. Alternatively, two fields may be coded together as a frame (frame pictures) similar to conventional coding of progressive video sequences. Here, consecutive lines of top and bottom fields are simply merged to form a frame.

3.14 Field and Frame Prediction

New motion-compensated field-prediction is assumed to contain only three field pictures and no B- pictures. In field prediction, predictions are made independently for each field by using data from one or more previously decoded fields, i.e., for a top field a prediction may be obtained from either a previously decoded top field (using motion-compensated prediction) or from the previously decoded bottom field belonging to the same picture.

Generally, the inter field prediction from the decoded field in the same picture is preferred if no motion occurs between fields. An indication to which reference field

is used for prediction is transmitted with the bit stream. Within a field picture all predictions are field predictions. Frame prediction forms a prediction for a frame picture based on one or more previously decoded frames. In a frame picture, either field or frame predictions may be used or the particular prediction mode preferred can be selected on a macro block-by-macro block basis. MPEG-2 has introduced new motion-compensation modes to efficiently explore temporal redundancies between fields, namely the "Dual Prime" prediction and motion compensation based on 16x8 block.

3.15 MPEG-2 Scalable Coding Extension

The MPEG-2 coding algorithm is intended to provide interoperability between different services and to flexibility support receiver with different display capabilities. Receivers either not capable or willing to reconstruct the full resolution video can decode subset of the layered bit stream to display video at lower spatial or temporal resolution or with lower quality. Another important purpose of scalable coding is to provide a layered video bit stream that is amenable for prioritized transmission.

Other important applications for scalable coding include video database browsing and multi resolution playback of video in multimedia environments. Here two layers are provided, each layer supporting video at a different scale, i.e., a multi resolution representation can be achieved by downscaling the input video signal into a lower resolution video (down sampling spatially or temporally). The downscaled version is encoded into a base-layer bit stream with reduced bit rate. The up scaled reconstructed base-layer video (up-sampled spatially or temporally) is used as a prediction for the coding of the original input video signal.

MPEG-2 has standardized three scalable coding schemes: signal to noise ratio (SNR) scalability (quality), spatial scalability, and temporal scalability, each of which are targeted to assist applications with particular requirements. The

scalability tools provide algorithmic extensions to the non scalable scheme defined in the MAIN profile. It is possible to combine different scalability tools into a hybrid coding scheme; i.e., interoperability between services with different spatial resolutions and frame rates can be supported by means of combining the spatial scalability and the temporal scalability tool into a hybrid-layered coding scheme. Interoperability between HDTV and SDTV services can be provided along, with a certain resilience to channel errors by combining the spatial scalability extensions with the SNR scalability tool

3.16 SNR Scalability

It provides graceful degradation (quality scalability) of the video quality in prioritized transmission media. If the base layer can be protected from transmission errors, a version of the video with gracefully reduced quality can be obtained by decoding the base layer signal only. The algorithm is based on a frequency (DCT domain) scalability technique. At the base layer the DCT coefficients are coarsely quantized to achieve moderate image quality at reduced bit rate. The enhancement layer encodes the difference between the non-quantized DCT coefficients and the quantized coefficients from the base layer with finer quantization step size. The method is implemented as a simple and straightforward extension to the MAIN Profile MPEG-2 coder and achieves excellent coding efficiency. It is also possible to use this method to obtain video with lower spatial resolution at the receiver.

3.17 Spatial Scalability

Spatial scalability supports displays with different spatial resolutions at the receiver; lower spatial resolution video can be reconstructed from the base layer. This functionality is useful for many applications including embedded coding for HDTV/TV systems, allowing a migration from a digital TV service to higher spatial resolution HDTV services. The algorithm is based on a classical pyramidal approach for progressive image coding. Spatial scalability can flexibly support a

wide range of spatial resolutions but adds considerable implementation complexity to the MAIN Profile coding scheme.

3.18 Temporal Scalability

The temporal scalability is similar to spatial scalability; a stereoscopic video can be supported with a layered bit stream suitable for receivers with stereoscopic display capabilities. Layering is achieved by providing a prediction of one of the images of the stereoscopic video (i.e., left view) in the enhancement layer based on coded images from the opposite view transmitted in the base layer.

3.19 Data Partitioning

Data partitioning assists with error concealment in the presence of transmission or channel errors in ATM, terrestrial broadcast, or magnetic recording environments. It can be entirely used as a post processing and preprocessing tool to any single-layer coding scheme. Similar to the SNR scalability tool, the algorithm is based on the separation of DCT coefficients and is implemented with very low complexity compared to the other scalable coding schemes. To provide error protection, the coded DCT coefficients in the bit stream are simply separated and transmitted in two layers with different error likelihood.

In this chapter a detail description of MPEG codec is discussed. In the next chapter implementation of different sub blocks is described.

Chapter 4

Implementation Details for Psycho visual Redundancy Removal

In this chapter we will see how major algorithms forming the implementation blocks of MPEG standard for spatial redundancy removal are implemented in VHDL according to the standard flow over FPGA device. We will also discuss possible scope for modifying the implemented structure to make it compatible for streaming image data coming from a real time source. Before we download data into FPGA we first convert it into grey levels varying in values from 0 to 256 levels being represented by eight bits.

4.1 Discrete Cosine Transform (DCT)

This is the first block through which data has to pass through. It is a lossy compression scheme transforms an 8x8 block from spatial to DCT domain and decomposed input pixel values into DCT coefficient in a two dimensional matrix. Due to inherent property of DCT to be periodic and even symmetric for a finite duration sequence, the low frequency components at the left top of the matrix carry the maximum energy whereas high frequency components at the bottom right of the matrix carry minimum energy .This is in contrast with human psycho visual effect according to which human is not able to see the high frequency components and thus, such coefficients can be quantized.[5],[11,(c)]

For most of the compression standards we take N = 8 as an 8x8 block does not have large memory requirement and by increasing the size does not lead to any significant improvement. For this each picture is divided into macro block of 16x16 order consisting of four such blocks to which DCT is performed using parallel architecture. Equation used for calculating DCT is given as follows

$$f(u,v)=\frac{1}{4}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(x,y)C(u)\ C\ (v)\ Cos\ (\frac{(2x+1)\pi u}{2M})\ Cos\ (\frac{(2y+1)\pi v}{2N})$$

.....Eq'n 4.1

This can be rewritten into two dimensional constants given below

$$C = K * Cos(\frac{(2*colnumber+1)*(rownumber*\pi)}{2*M})$$

.....Eq'n 4.2

$$K=\sqrt{\frac{1}{N}}\ for\ row=0 \qquad K=\sqrt{\frac{2}{N}}\ for\ row\neq 0$$

$$C' = K * Cos(\frac{(2*rownumber+1)*(colnumber*\pi)}{2*N}) \qquad \text{.....Eq'n 4.3}$$

$$K=\sqrt{\frac{1}{M}}\ for\ col=0 \qquad K=\sqrt{\frac{2}{M}}\ for\ col\neq 0$$

Thus, we first perform one dimensional DCT followed by a two buffer storage RAM followed by second dimension.

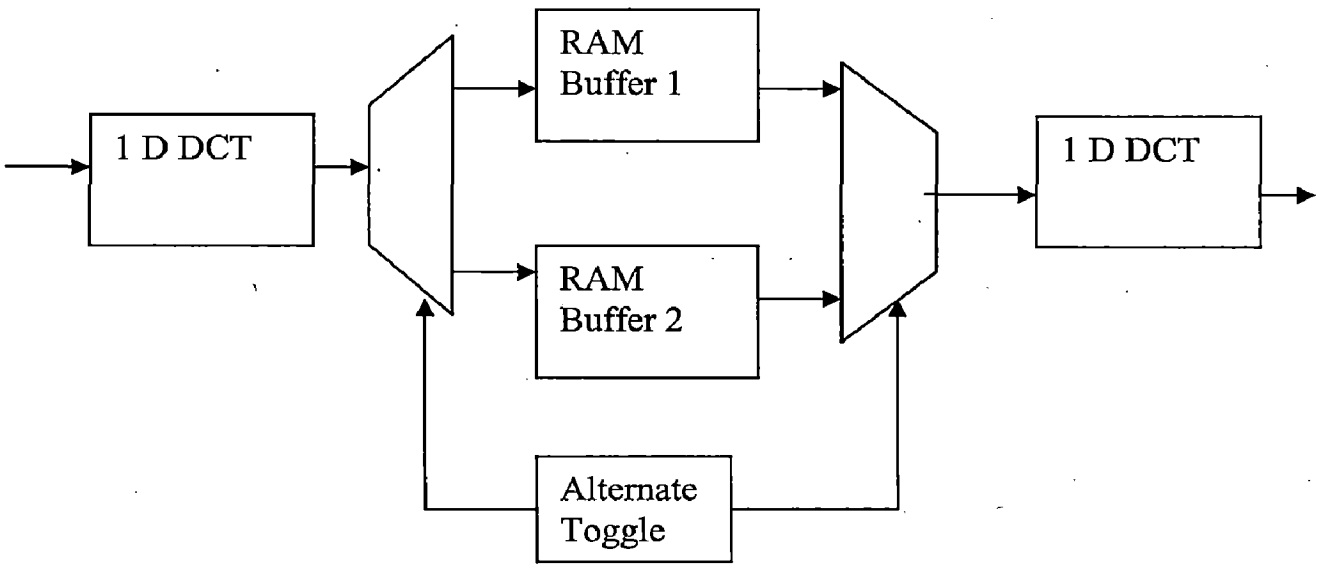


Figure 4.1 2 D DCT using vector processing

Here, we apply parallel multipliers to implement one dimensional DCT known as vector processing. The advantage obtained out of this is simple implementation with balanced and less complex structure. For input signal given as X vector we have two dimensional implementation given as

$$Y = C\ X\ C^T$$

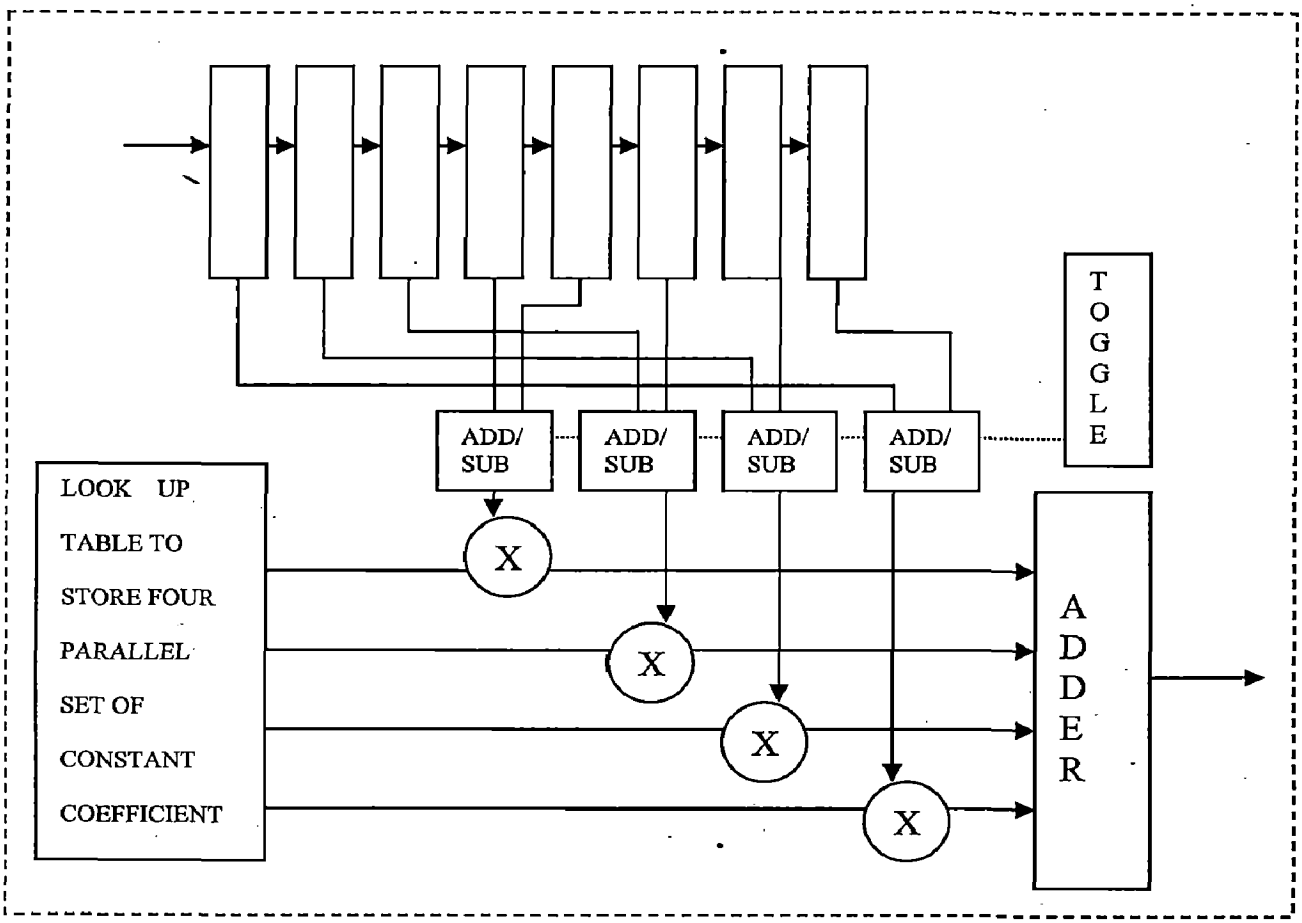


Figure 4.2 DCT implementation using vector implementation

Here, C and C^T are constants of DCT coefficient and inverse DCT coefficient. Thus, applying one dimensional DCT we follow the parallel structure due to are observation that the constant coefficient have values such that they can be multiplied with incoming signal in a set of four alternate addition and subtraction obtained by taking 8 input pixel values in set of (0,7);(1,6);(2,5);(3,4) and then adding them using quad input adder.

As shown in figure 4.3, first eight values of input are taken using shift registers are taken and then they are made to undergo above mentioned procedure to obtain one dimensional DCT values. Similar procedure is adopted for second one dimensional implementation to obtain final DCT output.

In RGB color space, the average value of all the color components is 128. This is converted to YCbCr color space where the average value of Y is 128, but the value of Cr and Cb is bias zero. The image pixels are preprocessed before going into the DCT coder to provide uniform processing. The preprocessing makes the average value to be zero by subtracting 128 from each pixel value. This value is added back after the inverse DCT operation. The 1D DCT values are first calculated and stored in a RAM. The second 1D-DCT is done on the values stored in the RAM. For each 1D implementation, inputs are loaded into an adder/ subtractor.

The output of the adder/ subtractor is fed into a multiplier. The constant coefficient multiplication values are stored in a ROM and fed into the second input of the multiplier. The equations for Z and Y show the even column values are obtained by adding the inputs, and the odd column values are obtained by subtracting the inputs. Thus, for every other clock an addition is done at the inputs. This control is achieved by using a simple toggle flip-flop with the output toggling High or Low to select an adder or a subtractor. The outputs of the four multipliers are added together resulting in the intermediate coefficients. The intermediate coefficients are stored in a RAM. The values stored in the intermediate RAM are read out one column at a time.

4.2 Quantization

Quantization is the process of selectively discarding visual information without significant loss in the visual effect. [4]

Quantization is the process of reducing the number of bits needed to store an integer value by reducing the precision of integer. For this each DCT coefficient is divided

by a fixed quantization coefficient from a standard quantization table and then rounded to the nearest integer value. Larger the Quantizer size smaller the DCT coefficient becomes after quantization. On the decoder side fractional bits are reduced to zero thus leading to loss in precision. [11(d)]

DCT divides the image into spatial domain ensuring that maximum energy gets confined to low frequency region as human visual capability is sensitive to sharp contrast in this region. Quantization further uses a threshold value to convert the DCT value to nearest coefficient by dividing it with a pre determined coefficient.

In MPEG standard the intra-frame AC coefficients (i.e., all coefficients other than (0,0)) are quantized without a dead-zone using an intra-quantizer matrix. The P (predictive) and B (bidirectional) frames are quantized by uniform quantizer with a dead-zone around zero. A dead zone is described as a quantized to zero area larger than the step size with decreasing sensitivity to input noise. A dead zone results in a string of zeroes at the output. For this non-intra quantizer matrix is used in this design.

The output of a 2D-DCT is read out one value at a time. Each DCT coefficient is divided by a corresponding quantization matrix value supplied by the quantization matrix. The encoder use the default matrix. There are three formats in image

In a 4:4:4 format there is no decimation of chrominance components. For every one Y or luma component, there is one Cb and one Cr component.
In 4:2:2 format, both chrominance components are decimated by two horizontally, making the number of chrominance components along a given line be half the number of luminance components.
In a 4:2:0 format, both chrominance components are decimated horizontally as well as vertically where the number of Cr or Cb components horizontally and vertically is half the number of luminance (Y) component.

With 4:2:0 data, two Q matrices are used, one for intra macro blocks and one for non-intra macro blocks. For 4:2:2 and 4:4:4 data, four matrices are used, two sets (intra and non-intra) for luminance and two sets (intra and non-intra) for chrominance. The default matrices corresponding to the luminance portion are loaded first. For 4:2:2 and 4:4:4 data, a different user-defined chrominance matrix can be loaded later. When we receive a sequence header code in the video bit stream all the Qmatrix values are set to default value. The Q matrix has 64 unsigned, 8 bit unsigned values. We use only Intra Quantizer matrix.

Intra Quantizer Matrix

$$\begin{bmatrix} 8 & 16 & 19 & 22 & 26 & 27 & 29 & 34 \\ 16 & 16 & 22 & 24 & 27 & 29 & 34 & 37 \\ 19 & 22 & 26 & 27 & 29 & 34 & 34 & 38 \\ 22 & 22 & 26 & 27 & 29 & 34 & 37 & 40 \\ 22 & 26 & 27 & 29 & 32 & 35 & 40 & 48 \\ 26 & 27 & 29 & 32 & 35 & 40 & 48 & 58 \\ 26 & 27 & 29 & 34 & 38 & 46 & 56 & 69 \\ 27 & 29 & 35 & 38 & 46 & 56 & 69 & 83 \end{bmatrix}$$

Non Intra Quantizer Matrix

$$\begin{bmatrix} 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \end{bmatrix}$$

Figure 4.3 Qmatrix for MPEG standard [4]

$$QDCT(i,j) = \frac{1}{2}\left[\frac{32 \times DCT(i,j)}{Qmatrix(i,j) \times QuantizerScale} + k\right] \qquad \text{Eq'n 4.4}$$

where k = 0 for intra block

k = sign of DCT coefficient for non-intra block

i.e. 1 for DCT (i, j) >0

0 for DCT (i, j) =0

-1 for DCT (i, j) <0

here, value zero is prohibited in denominator. The quantization of D.C. component is given as

$$Quantized\ Coefficient\ (0,0) = \frac{DCT(0,0)}{k} \qquad \text{.....Eq'n 4.5}$$

here, k is constant given by DCT precision selected:
k = 1 for eleven bit precision;

Each value in the quantization matrix can be pre-scaled when multiplied by a single integer value, known as the quantizer scale code. The quantizer scale code is a 5-bit unsigned integer in the one to 31 range. The encoder / decoder uses this value until another quantizer scale code is encountered either in a slice or a macroblock. The value zero is forbidden. Each of the 31 values of the quantizer scale code are mapped to two sets of integers, called "quantizer_scale", ranging from one to 112.(Refer Table 7-6,pp-84 [4])

In this implementation a fixed quantizer_scale code is taken and is multiplied with Qmatrix coefficient to finally obtain a quantizing coefficient for multiplying with input DCT value. The quantizer scale type is used to select between MPEG1 and MPEG2 standard based quantizer_scale. Thus, performing a fine tuning parameter for controlling bit rate at macroblock level. This ensures maximum DCT coefficient are reduced to zero.
Here, it is taken to default value '0' for MPEG1 compliance.

Inverse quantizer output for an MPEG system is given by

$$DCT = \frac{(2\times QDCT(i,j)+k)\times Qmatrix(i,j)\times quantizer\,scale}{32}$$

.....Eq'n 4.6

where k = 0 for intra block
k = sign of DCT coefficient for non-intra block
i.e. 1 for DCT (i, j) >0
0 for DCT (i, j) =0
-1 for DCT (i, j) <0

Implementation

Firstly, fixed value of 16 / Quantizer_scale is taken from the standard table and multiplied with inverse of default Qmatrix to obtain quantization coefficient. This coefficient is then multiplied with input DCT values to finally obtain quantized values which mostly consist of high frequency components being reduced to zero and low frequency components carrying a significant value. In the case of non intra block we can go for quantizer matrix with dead zone across zero. Similarly, for MPEG 2 standard we can select user defined Qmatrix but for this design flow end user will not get these applications as they are predefined at designer level.

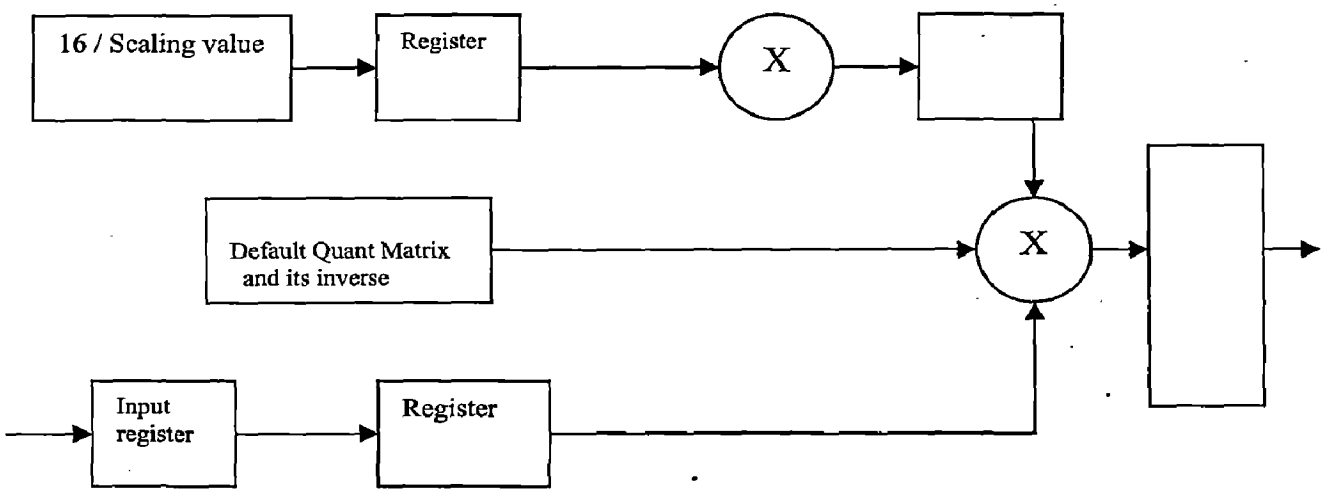


Figure 4.4 Implementation flow of Quantization

Similarly, we can apply inverse quantization by following the reverse flow. But here we have to take care of the mismatch error. This error occurs due to the fact that DCT coefficients are quantized.

Thus, in this chapter methods for lossy compression are discussed so as to remove psychological redundancy. We have also discussed possible block diagram for DCT generation and quantization of input pixel values. A detail VHDL code for the above implementation is included in the CD. In the next chapter different coding method employed for removal of inter pixel and coding redundancy will be discussed.

Chapter 5

Implementation Details for Coding and Intra Pixel Redundancy Removal

This chapter discusses about coding techniques applied to the quantized coefficient so as to perform lossless compression of the data by employing statistical features present in the input coefficient. The process includes three successive steps followed in a fixed order so as to achieve maximum compression. Further method for downloading data into the FPGA based board, its processing and finally display on the raster is discussed.

5.1 Variable Length Code

It is the final loss less stage in an MPEG system which is constituted of Zigzag scanning followed by Run length code further followed by Huffman Code.

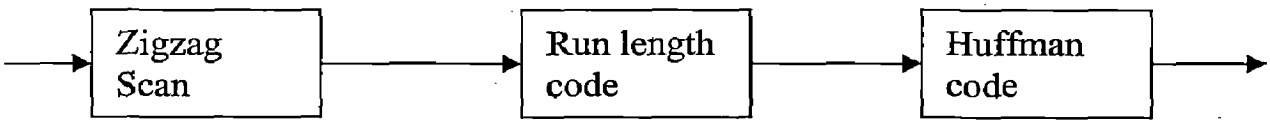


Figure 5.1 Variable Length Code

5.1.1 Zigzag Scan

In zigzag scanning, the quantized coefficients are read out in a zigzag order. This is done to arrange the coefficients in such a way that RLE and Huffman coding can be done to further compress the data. The scan puts the high frequency components together. These components are usually zeroes. [11(j)]

For MPEG2 interlaced video, a top field and a bottom field consist of a frame frozen at different instances of time. If there is a moving object in this frame, the intra-coded blocks containing this moving object will exhibit high vertical frequencies. These high frequencies are less likely to quantize to zero. In this case, an alternate zigzag scanning pattern is used so that the non-zero high frequencies are picked up sooner. This scanning mode is called the alternate scan and is used as an additional choice in MPEG2. MPEG1 uses only the regular zigzag scanning order.

0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

Increasing vertical frequency

Increasing horizontal frequency

0	4	6	20	22	36	38	52
1	5	7	21	23	37	39	53
2	8	19	24	34	40	50	54
3	9	18	25	35	41	51	55
10	17	26	30	42	46	56	60
11	16	27	31	43	47	57	61
12	15	28	32	44	48	58	62
13	14	29	33	45	49	59	63

Increasing vertical frequency

Increasing horizontal frequency

Fig.5.2 Zigzag scan orders allowed in MPEG1 & MPEG2 (top), Alternate scan used in MPEG2 (bottom) (table 7-2,7-3 ,pp 81 ref.[4])

5.1.2 Run Length Encoding

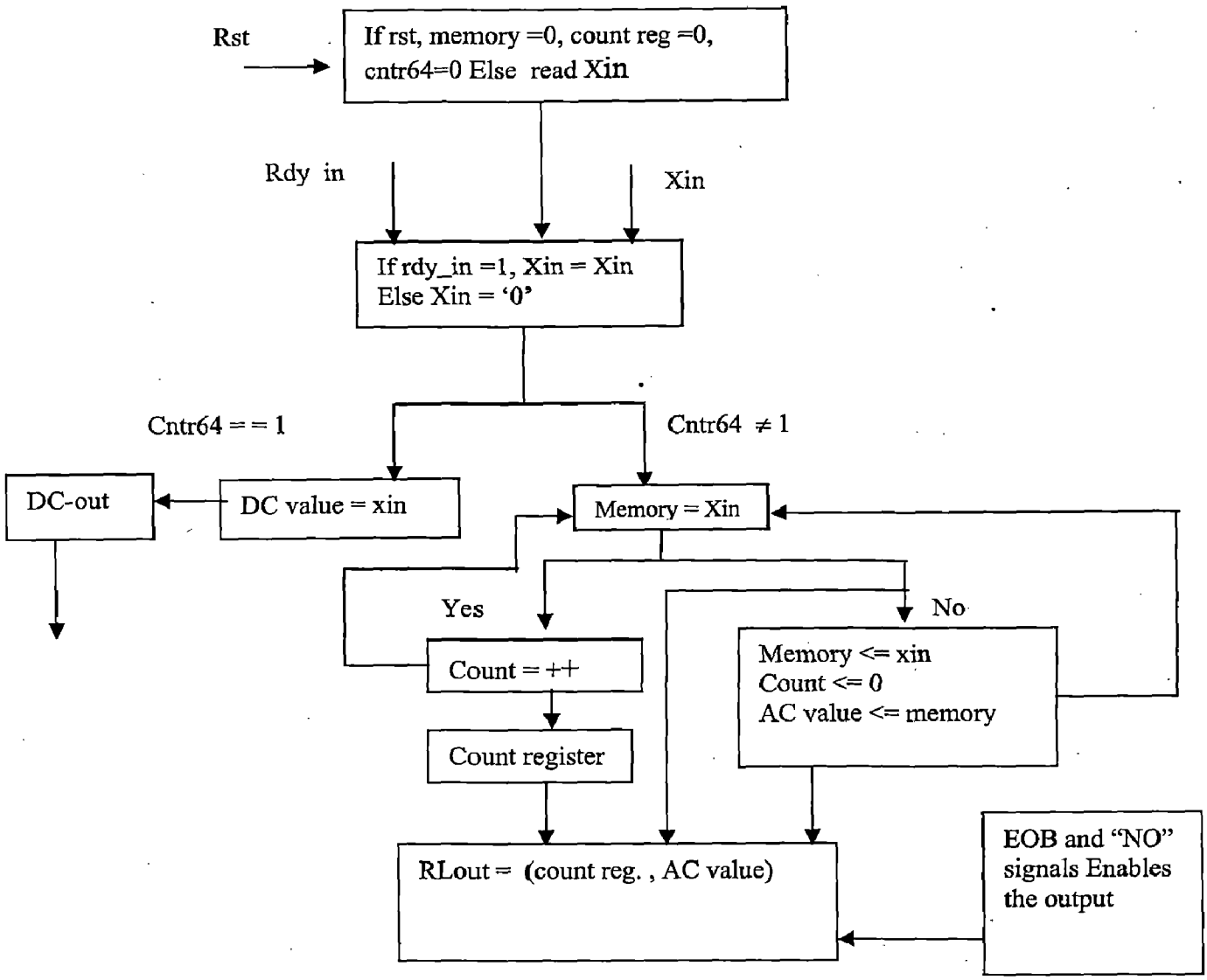
Basics

The quantized coefficients are read out in a zigzag order from DC component to the highest frequency component. RLE is used to code the string of data from the zigzag scanner. Run length encoding codes the coefficients in the quantized block into their run length (or number of occurrences) and their level or amplitude. For example, to transmit 3 coefficients of value 10 we transmit as follows: {25, 25, 25}. By using RLE, we transmit the level i.e., 10 and the run of the value 10, which is 3. Thus by using RLE, we transmit {3, 25} and the amount of data transmitted is reduced. Typically RLE encodes a run of symbols into two bytes, a count and a symbol.

To define an 8x8 block without RLE, we need 64 coefficients. We can make use of the fact that many of the quantized coefficients in the 8x8 block are zero to further compress the data. Coding can be terminated when there are no more non-zero coefficients in the zigzag sequence. Using the "end-of-block" code does termination of coding. The compression ratio obtained by RLE is given by

Compression Ratio = original size / compressed size: 1.

RLE is most often used to compress black and white or 8 bit indexed color images where long runs are likely. RLE is not generally used for high color images such as photographs where in general each pixel will vary from the last. RLE cannot achieve high compression ratios compared to other compression methods, but it is easy to implement and is quick to execute. To encode a run in RLE, it is required that there is a minimum of two characters worth of information, otherwise a run of single character takes more space and thus, turns out to be redundant in itself.



EOB	:	End Of Buffer.
MEM	:	Memory Register
Cntr64	:	64 state counter

Figure 5.3 RLE Implementation [11]

In this implementation input data is received under a count of 64.For this first input which is D.C. value is taken separately and from then onwards data are taken and compared with the previous data stored in a memory register. If same data is received then a count register is increment to keep a count of repetition otherwise count is concatenated with the data value and new data is stored in memory register. Thus, a track of run length and data value is maintained.

5.2 Huffman Coding

Huffman coding is used to code values statistically according to their probability of occurrence.

It is a process of assigning short code words to highly probable values and long code words to less probable values. The output symbols from RLE are assigned binary code words depending on the statistics of the symbol. Symbols which are frequently occurring are assigned short code words whereas rarely occurring symbols are assigned long code words. The resulting code string can be uniquely decoded to get the original output of the run length encoder. The code assignment procedure developed by Huffman is used to get the optimum code word assignment for a set of input symbols. [11(g)]

MPEG follows fixed code table in which code table does not change with the input video sequence. To define the entries in the MPEG-2 Huffman table standardized by ISO many image sequences were coded and the statistics were used. MPEG defines a set of variable-length code (VLC) for each of the probable run/level combinations. The run/level combinations not found in the table are represented by an escape code followed by a six-bit code for the run and an eight or 16-bit code for the level. The end-of block (EOB) code is used when all the remaining coefficients in the 8 x 8 block are zeroes.

Coding of the 8 x 8 block starts from the DC coefficient in the zigzag order. When there are no more non zero coefficients remaining in the zigzag order, the EOB code is used to terminate coding. Since the probability of occurrence of the EOB symbol is high, it is assigned a two-bit code "10".

The VLC tables used in MPEG-2 are not true Huffman codes. They are optimized for a range of bit rates to sample rate ratios. Most of the code words in the MPEG-2 tables were carried over from the H.261 standard. The DCT coefficient tables in MPEG-2 assume equal probability for both positive and negative coefficients.

MPEG-2 VLC uses a new table, (Table B-15, pp166-169[4]). It is better suited for the statistics of intra-coded blocks. The EOB code for Table B-14(pp 162-165,[4]) has two bits but for TableB-15[4], the EOB has four bits. This implies that an intra-coded block can have on average of 24 or 16 non-zero AC coefficients. For non intra-coded blocks, the statistics point to an average of 22 or four non-zero AC coefficients. Both Table b-14[4] and TableB-15[4] have 113 entries.

The last bit "s" of the code denotes the sign of the level with s = 0 for positive and s = 1 for negative. EOB code in the VLC table is used to indicate the status of the rest of the coefficients as zero. The EOB cannot occur as the only code in a block since no coding was done in the block. There are run/level combinations not defined in the VLC tables. When the variable length coder sees an undefined run/level combination, it codes the run into a 6-bit binary value and the level into a 12-bit signed level value. Before coding an undefined run/level pair, a 6-bit "escape code" is used to denote that the next six to12 bits are not from the VLC table.[4]

5.2.1 Encoding DC Coefficients

Due to high redundancy between adjacent quantized DC coefficients of 8 x 8 blocks, the difference in DC values is encoded using VLC. The difference signals range from –255 to 255 in MPEG-1 and from –2047 to 2047 in MPEG-2. The size of the differential DC value (dct_dc_size) is found in Table 1(appendix A). The size denotes the number of bits used to represent a particular value. Two different VLC codes are used for coding the DCT DC coefficient for luma and chroma.

5.2.2 AC Coefficients encoding

To code AC coefficients we use macroblock type and intra_vlc_format value according to Table 7-3(pp-79,[4]). For MPEG-1 coding, only Table B-14[4] is used. For MPEG-2, Table B-15[4] is used for intra-coded blocks and Table B-14[4] for non-intra coded blocks. In Table B-14[4], there are two Huffman codes for the run/level combination of 0/1. The code 1s is used if the 0/1 pair represents the first coefficient or the DC coefficient in the block. Term 11s is used as the Huffman

code for subsequent 0/1 run/level pairs,. The s in the code denotes the sign of the coefficient, "0" for positive and 1 for negative. The run/level pair for DC coefficient of "+1" and the EOB code has the same Huffman code. The differentiation is apparent since the EOB code will not be the first code in the block. In intra coding, since the DC value is coded separately, the first coded symbol is the first AC value. In this case, this first coded value can have a run/level of 0/1. For a run/level pair that is not defined in the VLC Table B-14[4] and Table B-15[4], an escape code is used according to Table B-16(pp 169[4]), followed by a 6-bit run symbol and 12-bit level symbol. The run/level value is used to access the ROM and the corresponding variable length code is read out. [11], [Contributed by Mr. Turney from Xilinx Video Application Center]

5.3 Huffman Implementation

We use standard flow diagram proposed for implementation of Huffman code. Here, first we take the input D.C. coefficient obtained from run length and determine the appropriate size to represent it using Look up table standardized under ISO. From this table we determine the equivalent code followed by number of bits required to represent it. For A.C. coefficient we determine the equivalent code according to run, level coefficient again using standard LUT. After this we perform data count and shift operation to finally obtain output in chunks of sixteen bits through three level register. This process runs according to following tables.

Condition	Cl_sum	Cl_sum_prev (checked)	H.F	F.F
RST	39	0	0	0
Cl_rdy=1(after 2 clk) + cl_sum_prev < 16	39 – cl_sum_prev	Codelength + cl_sum_prev	0	0
16< cl_sum_prev < 32	-do-	Codelength + (cl_sum_prev-16)	1	0
Cl_sum_prev>= 32	-do-	Codelength + (cl_sum_prev - 32)	1	1

Table 5.1 Code length shift at multiplier.

FF : HF	Upper Register (U.R)	Middle Register (M.R.)	Lower Register (L.R.)
RST others	0 U.R.	0 M.R.	0 L.R.
00	Multiply out (M.O.) [38:23] OR U.R [16:1]	M.O. [22:7]	M.O. [16:0] & "000000000"
01	M.O.[38:23] OR M.R.[16:1]	M.O.[22:7]	M.O.[6:0] &"000000000"
11	M.O [38:23] OR L.R.	M.O.[22:7]	"0000000000000000"

Table 5.2 Content of output register for Huffman Implementation

As the output generated by the Huffman coder has to be in the fixed number of bits so a shifting process is adopted as been tabulated in table 5.1 and 5.2. The output are obtained through upper and middle registers when they are filled by 16 bit data each and lower register is used for shifting data to middle and upper register. Two flags namely half flag and full flag are used for indicating status of middle and

upper flag and output is obtained when both or either of them are set high. The VHDL code block for this implementation is given in the CD attached.

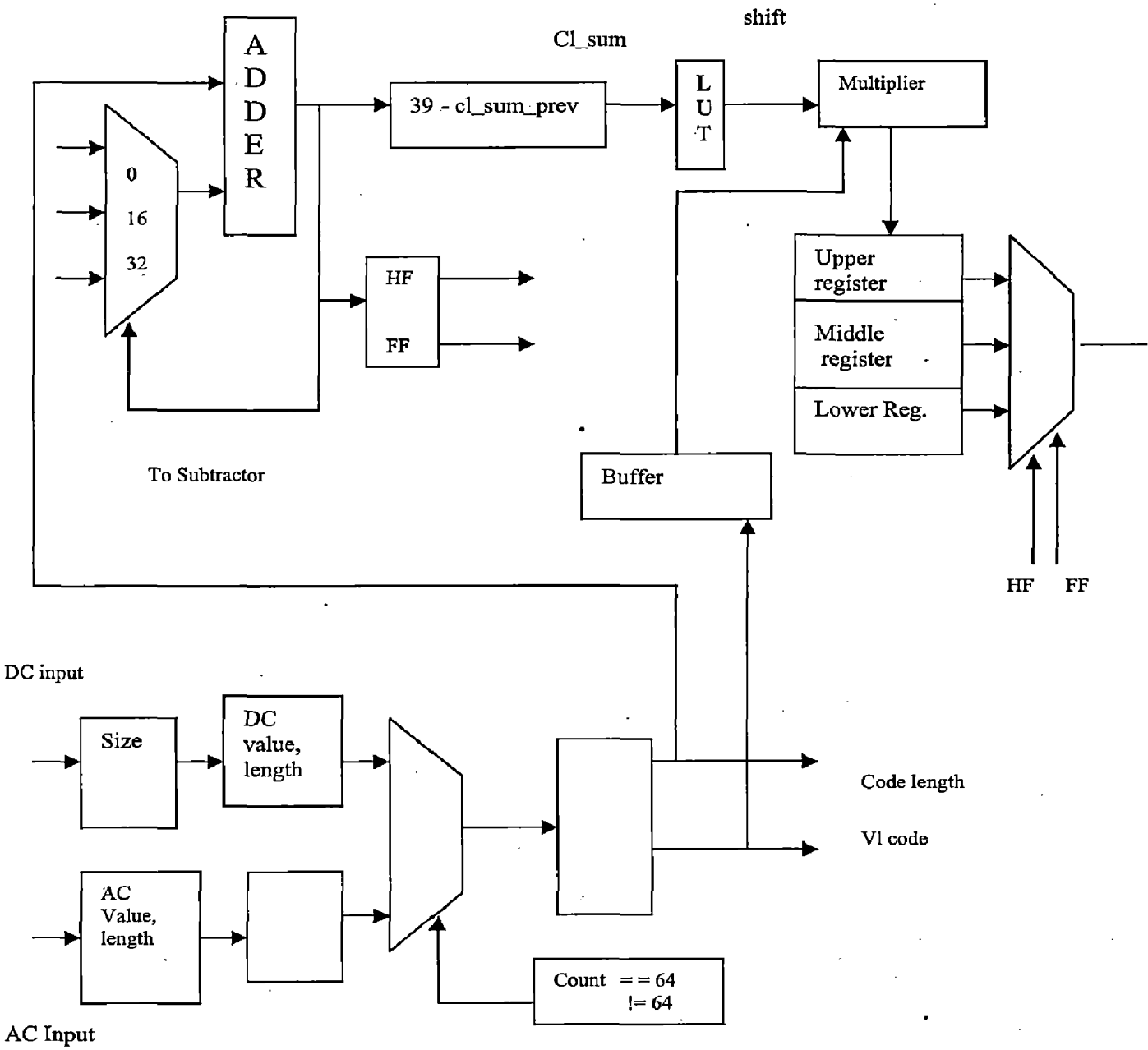


Figure 5.4 Huffman implementation for Intra Frame Coding [11]

5.4 External Interfaces

In this implementation the image data in grey level is first converted into data file with header being removed and then downloaded into the external SRAM provided on the video application card through PCI interface using Slave Impatt mode. Second step is to read the data from SRAM into algorithm perform compression

and decompression algorithms and then store the data back into other SRAM. Finally, compressed-decompressed data is read from SRAM and superimposed over raster pulses. For data to be read or written on to SRAM certain control signals are to be written. They are enlisted below

1. To generate initial addresses for read and write memory location and to perform further increment of address.
2. To generate control signals for read and write enable at correct instant for the different memories.

To achieve this control we write state machine in synchronization with the algorithm and test it for various conditions. The SRAM interface requires initial address to be provided by the designer and then to be incremented in order to read values for one block at a time.

It also requires enable signals to be generated at right time so as to match with latency of the algorithms.

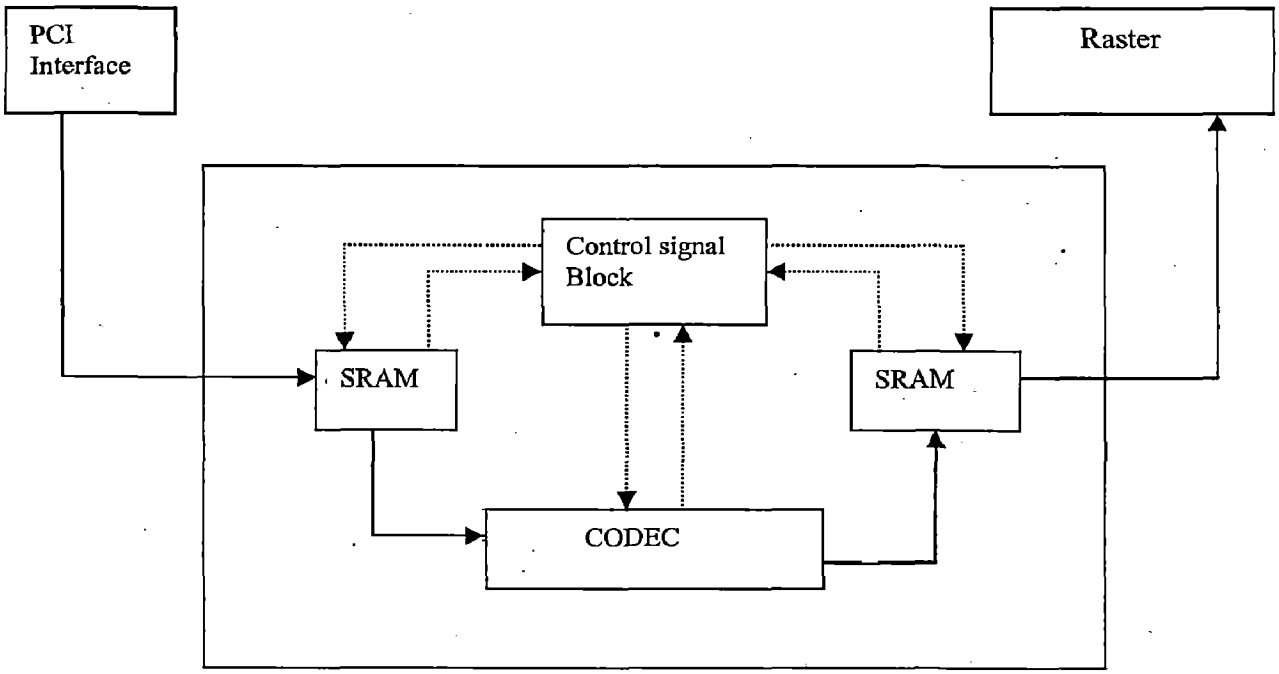


Figure 5.5 Interface block diagram

The complete implementation for MPEG coder with motion estimation will look like one mentioned in figure 5.5

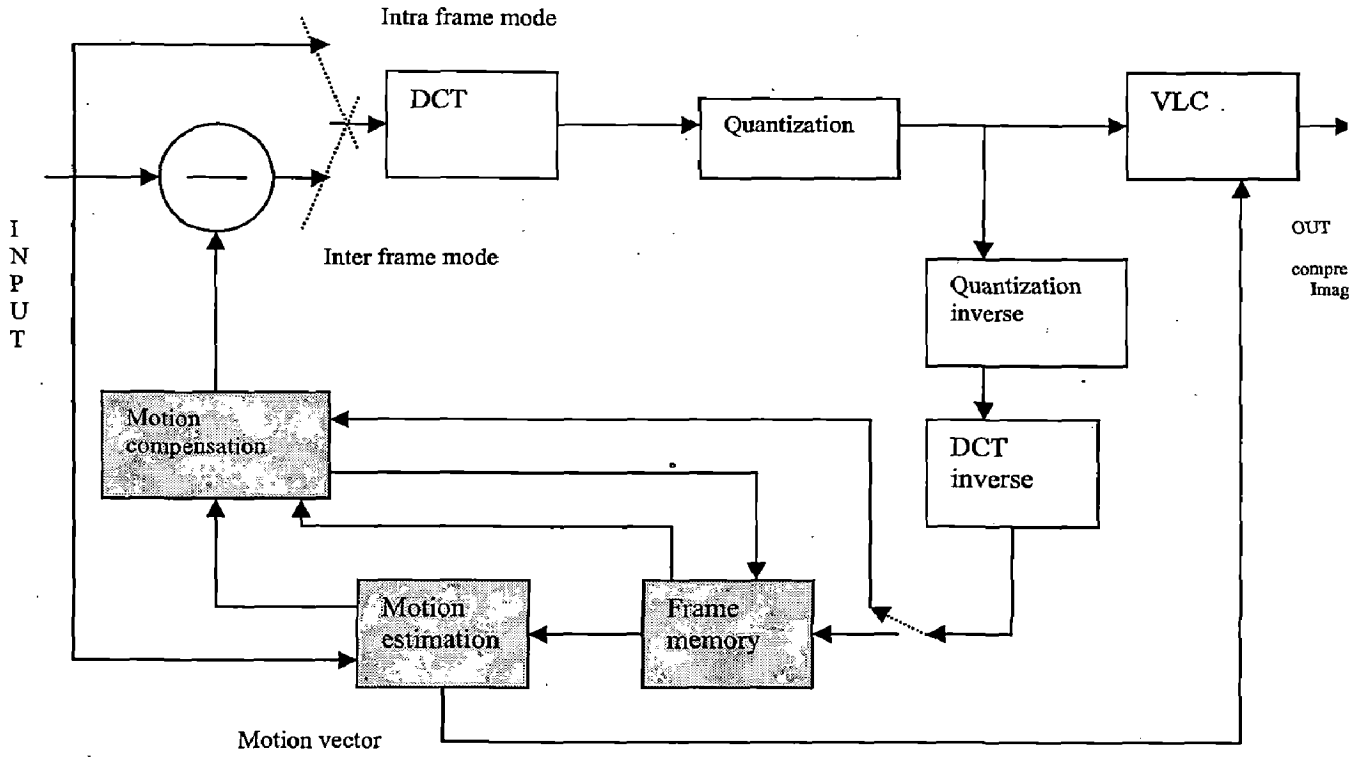


Figure 5.6 MPEG implementation block diagram [1]

In this chapter different coding methods for compression of image have been discussed including run length and Huffman code. A brief description of interface with the CPU and display raster is also discussed.

Chapter 6

Device Introduction, Results and Conclusion

In this chapter the device used for implementing the MPEG coder design will be discuss and various features provided in it will be enlisted. Then results obtained from simulating the design will be displayed and finally, different features envisioned in the simulation are discussed.

6.1 VIRTEX-II PRO PCI VDO CARD

Virtex-II Pro based video processing card with PCI interface offers a cost-effective platform for developing video and multimedia based applications. With on board high speed video ADC, DAC the board supports real time video processing of component video signals of NTSC or PAL standards. The on board SRAM and Flash memories may be used as data/code store for PowerPC or as video coefficient/data buffer(s). The board supports three different modes of FPGA configuration. Namely, configuration through PROM, JTAG Port and PCI. Flash PROM memory can be programmed with code for PowerPC through PCI interface. [17]

This board can also be used as standalone video processing board. In that case RS232 port provided on board will be used to program flash PROM. Further this platform is optimized for experimentation with 32-bit IBM PowerPC™ RISC processor core integrated into the FPGA fabric.

6.1.1 SPECIFICATIONS

- Analog Input: - ADC – 14 bit, 10 MSPS single channel is available using AD9240.

- Video Input: - 3 channels, 8 bit, 30MSPS single Video Input channel is available using Video ADC TLV5734.

- Video Output: - Two Video Output channels provide NTSC, PAL compliant component video signals using 10 bit Video DAC THS8133.

- User I/Os: - 16 IOs are available with XC2VP30 device
  32 IOs are available with XC2VP40 & XC2VP50 device.
  IOs are provided on two 20-pin box type FRC connectors.

- Memory: - 5 fully independent banks of 1M X 16 SRAM (NEC make uPD4416016).
  3 fully independent banks of 512K X 16 Flash PROM (LH28F800).

- Serial Interface: - Two RS232 Channels (MAX3223). One Mil STD 1553 channel (optional).

- PCI Interface: - 32 bit 33 MHz master interface, with facility for DMA transfers

- Drivers: - Windows 98/2000 Driver and APIs for capturing the acquired data on to hard disc or for User application.

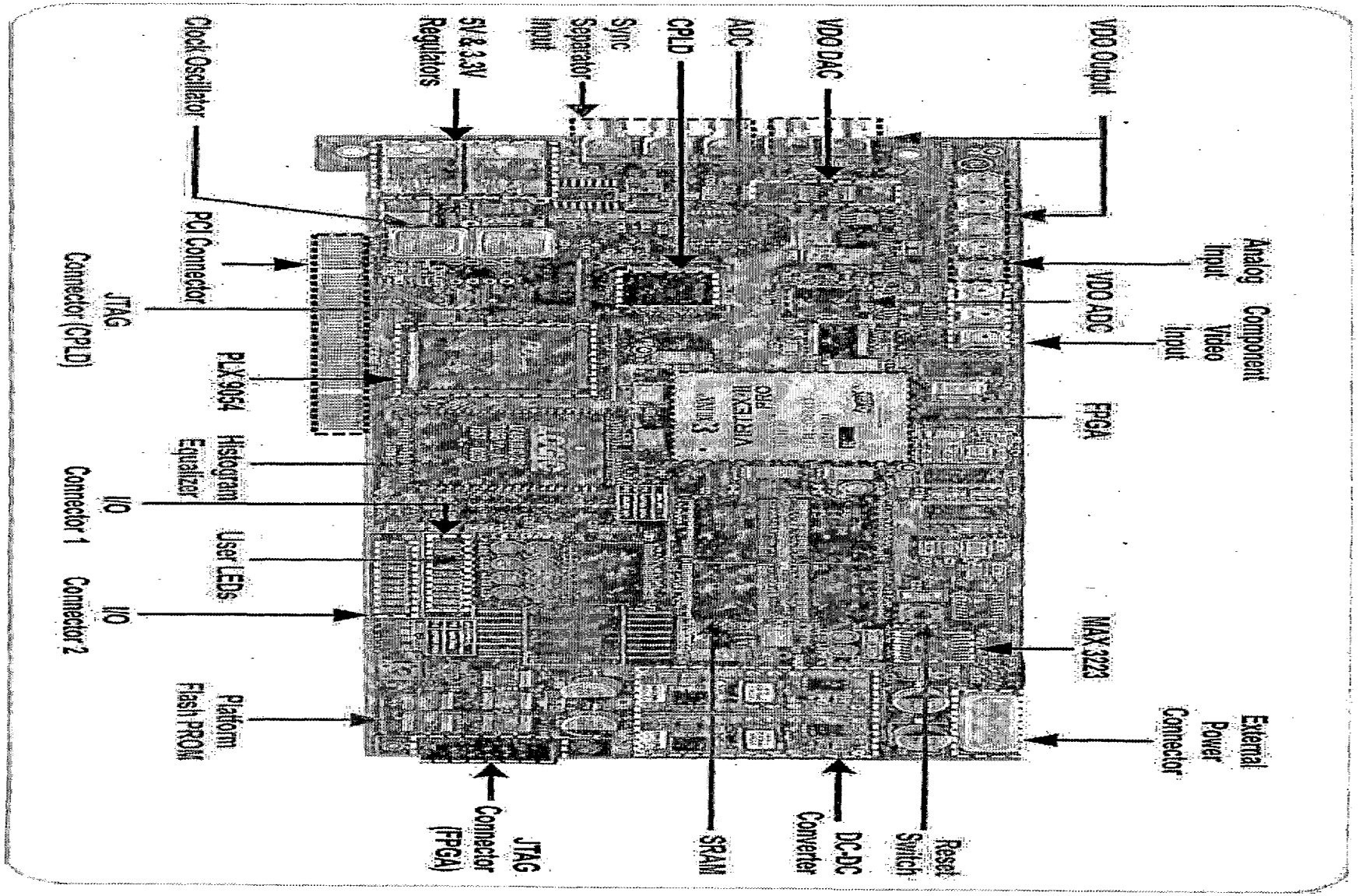


Figure 6.1 Top view of VIRTEX-II PRO PCI VDO CARD[17]

6.1.2 HARDWARE DESCRIPTION

The MXV2PFA-PCIVDO-001 is a PCI bus plug-in card for PCs.

It consists of:

- FPGA – Xilinx Virtex-II Pro XC2VP30 / XC2VP40 / XC2VP50 device in FF1152 package.

  These are platform FPGA's that are based on IP cores and customized modules, optimized for high density and high performance system design. They empower complete solutions for telecommunication, wireless, networking, and Video and DSP applications.[17]

  The family incorporates multi gigabit transceivers using RocketIO technology and 32-bit microprocessors (IBM's Power PC 405 CPU) in the FPGA fabric.

  The PowerPC core is a 0.13 µm implementation of the IBM PowerPC 405D4 core, with the ability to operate at 300+ MHz while maintaining low power consumption. Specially designed interface logic integrates the core with the surrounding CLBs, block RAMs, and general routing resources.

- PowerPC 405 Processor - The PPC405 RISC CPU can execute instructions at a sustained rate of one instruction per cycle. On-chip instruction and data cache reduce design complexity and improve system throughput. Within the Processor Block, there are four components:

1) Embedded IBM PowerPC 405-D5 RISC CPU core

Embedded PowerPC 405 (PPC405) core operating at 300+ MHz maintains low power consumption.

Specially designed interface logic integrates the core with the surrounding CLBs, block RAMs, and general routing resources.

Embedded PowerPC 405 core consists of the following functional blocks

- Instruction & Data Cache units
- Memory Management unit
- Fetch & Decode unit

- Execution unit
- Timers
- Debug logic unit

2) On-Chip Memory (OCM) controllers and interfaces

The OCM controllers (DSOCM & ISOCM) serve as dedicated interfaces between the block RAMs in the FPGA fabric and OCM signals available on the embedded PPC405 core.

ISOCM controller provides an interface to the 64-bit Instruction-Side Block RAM (ISBRAM) while

DSOCM controller provides an interface to the 32-bit Data-Side Block RAM (DSBRAM).

- Clocks – User has option of using 2 different clock sources on board which provide all necessary clocks for User logic and PowerPC

  The clock sources provided on board are as follows

  1. Clock Source1

  32 MHz clock oscillator – supplied as standard and can be used as system clock for PPC.

  2. Clock Source2 Socket for user clock source (foot print compatible with clock sources from 40 to 300 MHz). User can assemble a clock oscillator of his choice if his system frequency is different from 40 MHz. It should be noted that all the clock sources are connected to the global clock inputs. Thus user can use these external clocks as input to the on-chip DCM's.

- Flash PROM - 1.5 MB of Flash PROM is provided as standard, using three 512K X 16 PROMs. Can be upgraded to 3 MB, as these proms are footprint compatible with 1M X 16 PROMs.

  Note - This up gradation is possible only during manufacturing.

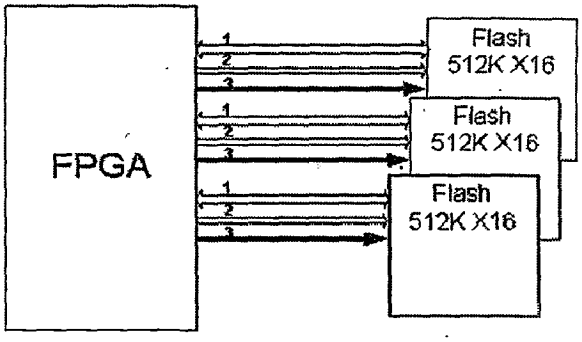


Figure 6.2 Flash PROM

- SRAM - Five 1M X 16 SRAM devices are independently interfaced to the on board FPGA.

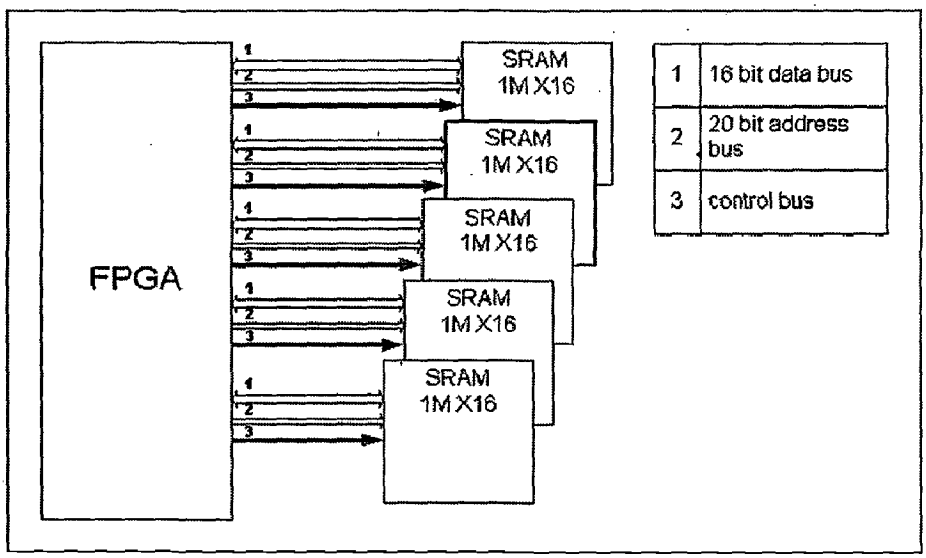


Figure 6.3 SRAM Interface

- RS ,232 Port - RS232C compatible connectivity is provided using device MAX3223. Signals provided are Rx, Tx, RTS and CTS. These signals are terminated on a 10 pin FRC connector on board and a 10 pin FRC to 9-pin D connector interface cable is provided as standard accessory with the board. Is compatible with the UART core provided by the Xilinx EDK.

- PCI Interface – A 32-bit, 33 MHz PCI interface, using PLX-9054 master interface with DMA capability.

- Analog Input – One analog input channel is available with the following specifications.

ADC - AD9240.

- Resolution – 14 bits
- Max Sampling rate – 10 MSPS
- Input range - 0 to 5 Volts, single ended
- Input buffer – using AD8052 op-amp.
- Connector type – SMA.

AD9240 is useful in various applications such as imaging, communications, and medical and data acquisition systems.

- Video Input - One video input channel is available with the following specifications.
  - Video ADC - TLV5734
  - Video signal format – NTSC / PAL compliant RGB /YUV component video signal.
  - Resolution – 8 bit
  - Sampling rate –30 MSPS maximum.
  - Input range - 1 Vp-p.
  - Input buffer – using AD8052 op-amp.
  - Connector type – SMA
  - Selectable clamping function for RGB/YUV applications.
  - Selectable Output Data Format for 4:4:4 (RGB, YUV), 4:2:2 and 4:1:1 (YUV) Format.

- Video Output - Two video output channels are provided on board with the following specifications
  - Video DAC - THS8133
  - Video signal format – NTSC / PAL compliant RGB /YUV component video signal.
  - Resolution – 10 bit
  - Sampling rate –80 MSPS maximum.

- Connector type – SMA

THS8133 provides current output; these current outputs can be converted into NTSC/PAL standard voltage levels by connecting a double terminated 75 ohms load.

- Sync Generation -Using sync, sync_t control signals, video sync signals can be added either on AGY (G/Y) channel or on all three channels with 7:3 video/sync ratios. Depending on the timing control of these signals both horizontal and vertical sync signals can be generated as well as either bi-level negative going or tri level pulses can be generated.
- Blanking Generation - An additional control input BLANK is provided that will fix the output amplitude on all channels to the blanking level. The absolute amplitude of the blanking level with respect to active video is determined by the GBR or YPbPr operation mode of the device.

- Histogram Equalizer - The on board histogrammer LF48410 is capable of generating histograms and cumulative distribution functions of video images. It provides following features:
  - 40 MHz data input and computation rate.
  - 1024 X 24 bit memory array
  - Histograms of images up to 4k X 4k with 10-bit pixel resolution.
  - User programmable modes – Histogram, histogram accumulate mode, look-up table mode, Delay memory, single port memory.

- Sync Separator - Sync separator EL4583 used on board extracts timing from video sync in NTSC, PAL, and SECAM systems.
  - Sync Separator – EL4583
  - Input voltage range – 0.5 V to 2 Vp-p.
  - Output signals –composite sync signal, vertical sync signal, horizontal sync signal, burst signal, odd/even signal, no signal detect output.
  - Connector type – SMA

- Digital I/Os - Maximum of 16 true bi-directional IOs are available when using XC2VP30 device. (32 with XC2VP40 and XC2VP50).

  These IO's are provided on two FRC connectors as follows,

  I/O Connector1

  A 20-pin box type FRC connector– provides 16 single ended, 5V tolerant IOs. These IOs support LVTTL, LVCMOSI_33 signaling standard (3.3V tolerant).

  I/O Connector2

  A 20-pin box type FRC connector– provides 16 single ended, 5V tolerant IOs. These IOs support LVTTL, LVCMOSI_33 signaling standard.

  5V tolerance is achieved by using quick switches for isolating the User I/O's and FPGA I/O's.

  For more information on quick switches visit www.idt.com/quickswitch

- Power Supply - When Board is to be used as PCI add on card then Board can be powered from PC's SMPS. While using the board in standalone mode an external power supply will power the board.

  Required Vccint (Core) and Vcco voltages are generated using two DC-DC converters - PTH05010W. (These parts are recommended by Xilinx.)

  Vccaux and Rocket IO supplies; 2.5 V are generated using LDO (LT1963).

  5VA & 3.3VA required by the analog section (ADC and DAC) is generated using regulators LM317.

- User LEDs - 8 LEDs are provided on board, which can be used by user to monitor signals from his design.
- Reset Switch - Can be used by user as a manual Reset input while verifying his designs.

6.2 Results

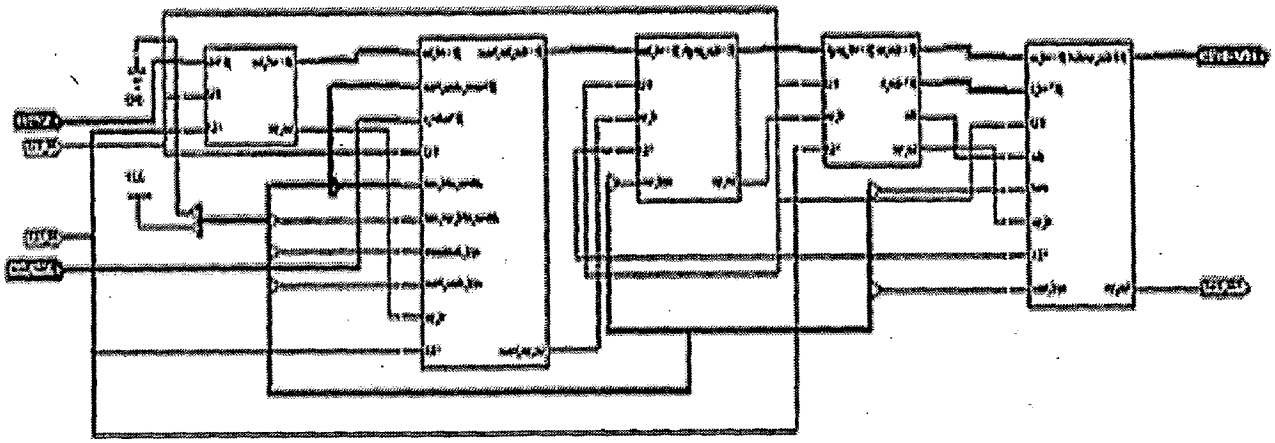


Figure 6.4 RTL Schematic for MPEG ENCODER

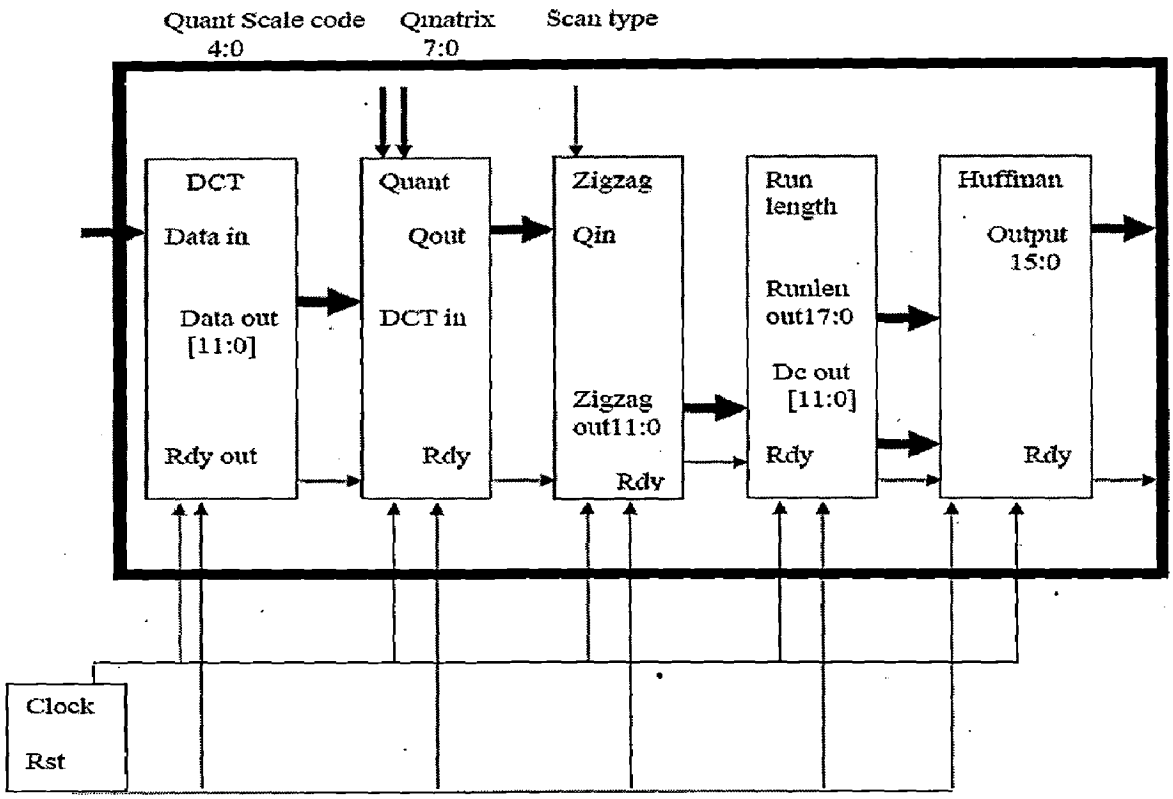


Figure 6.5 Implemented MPEG Encoder Block

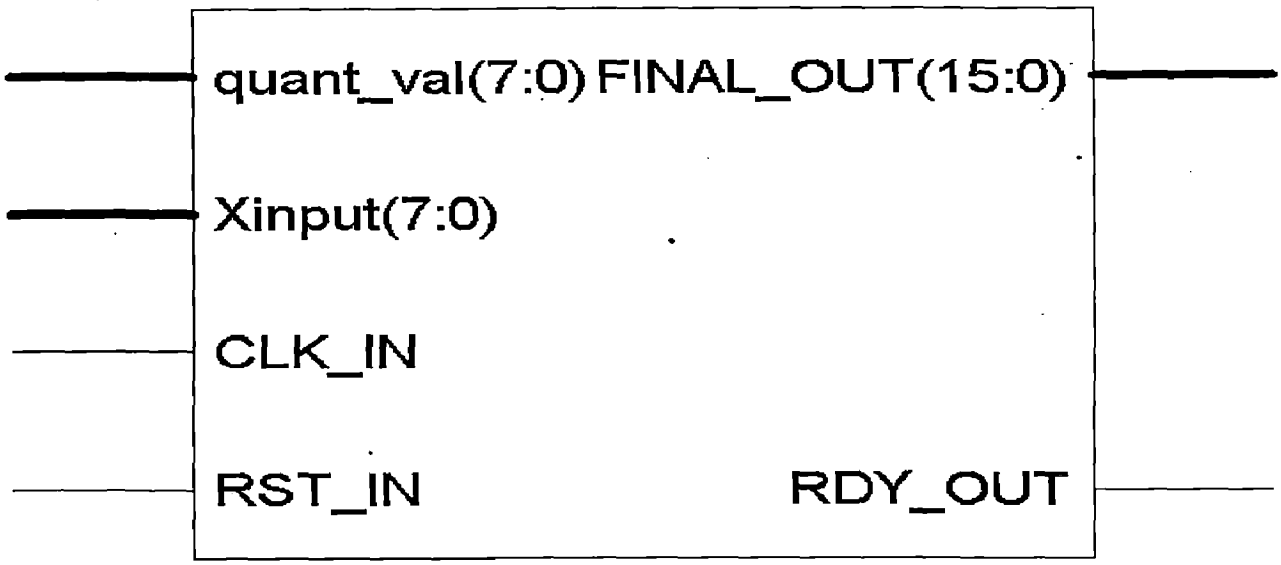


Figure 6.6 MPEG ENCODER BLOCK

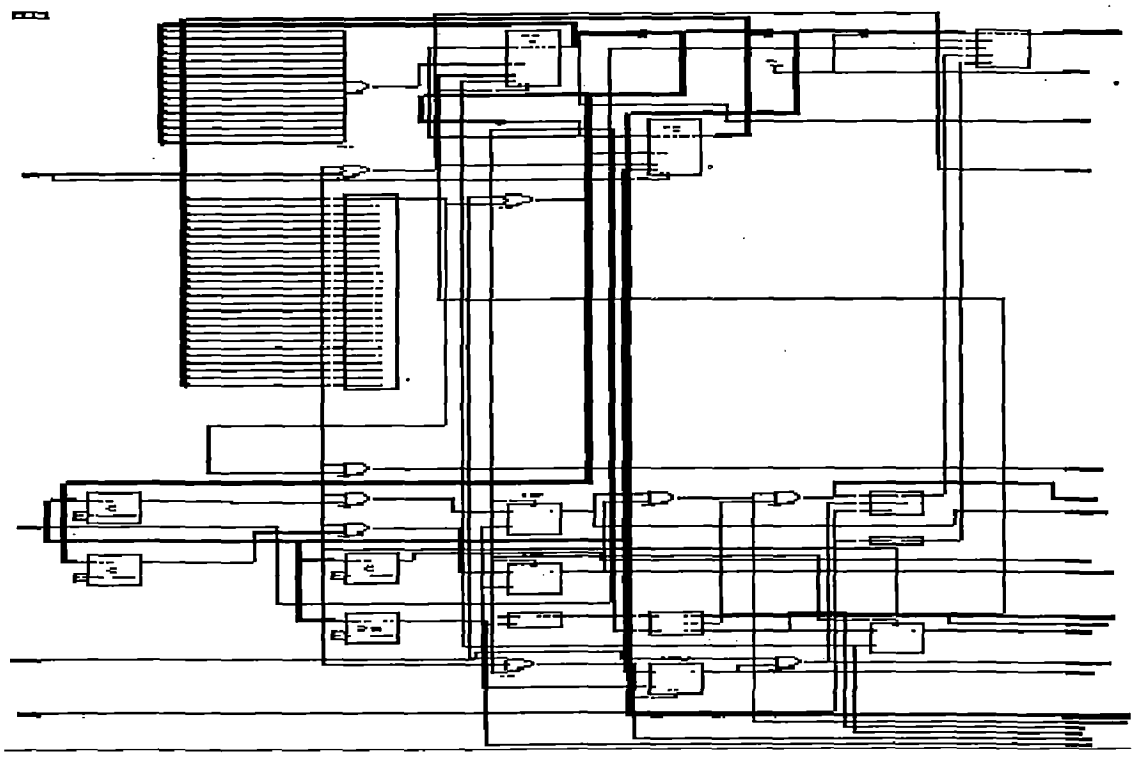


Figure 6.7 TV FPGA INTERFACE RTL Model

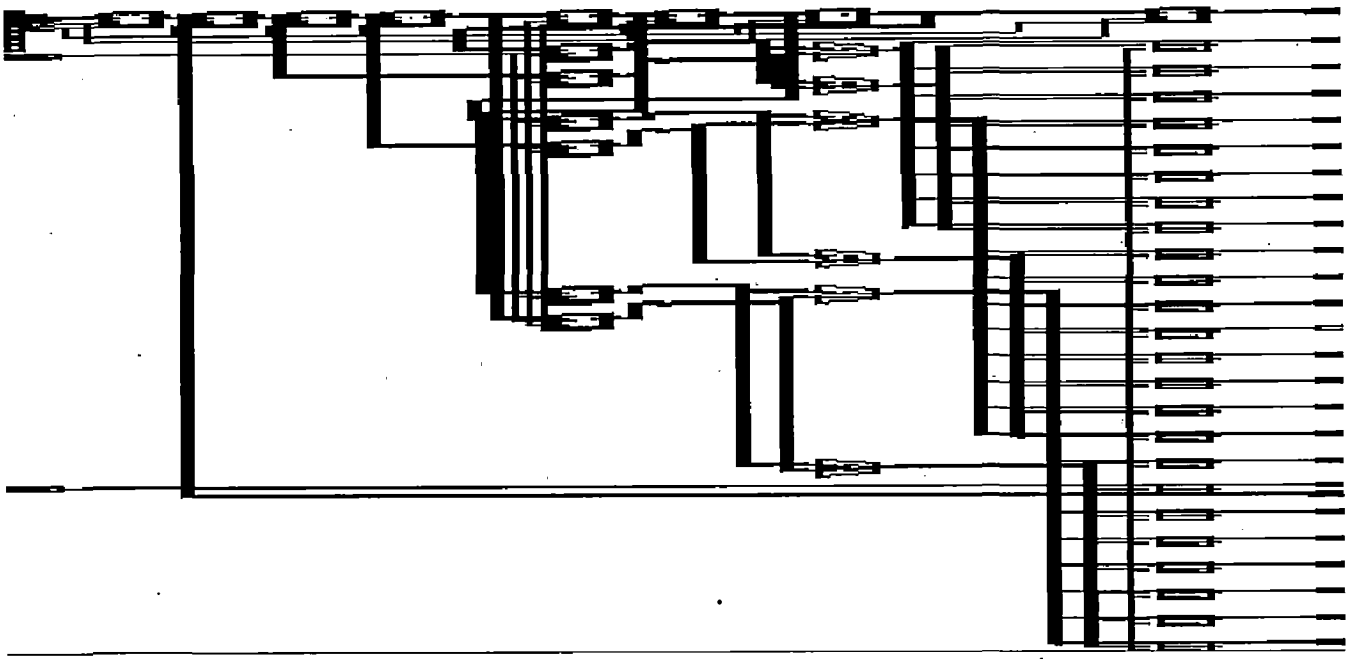


Figure 6.8 DCT Implemented RTL Level

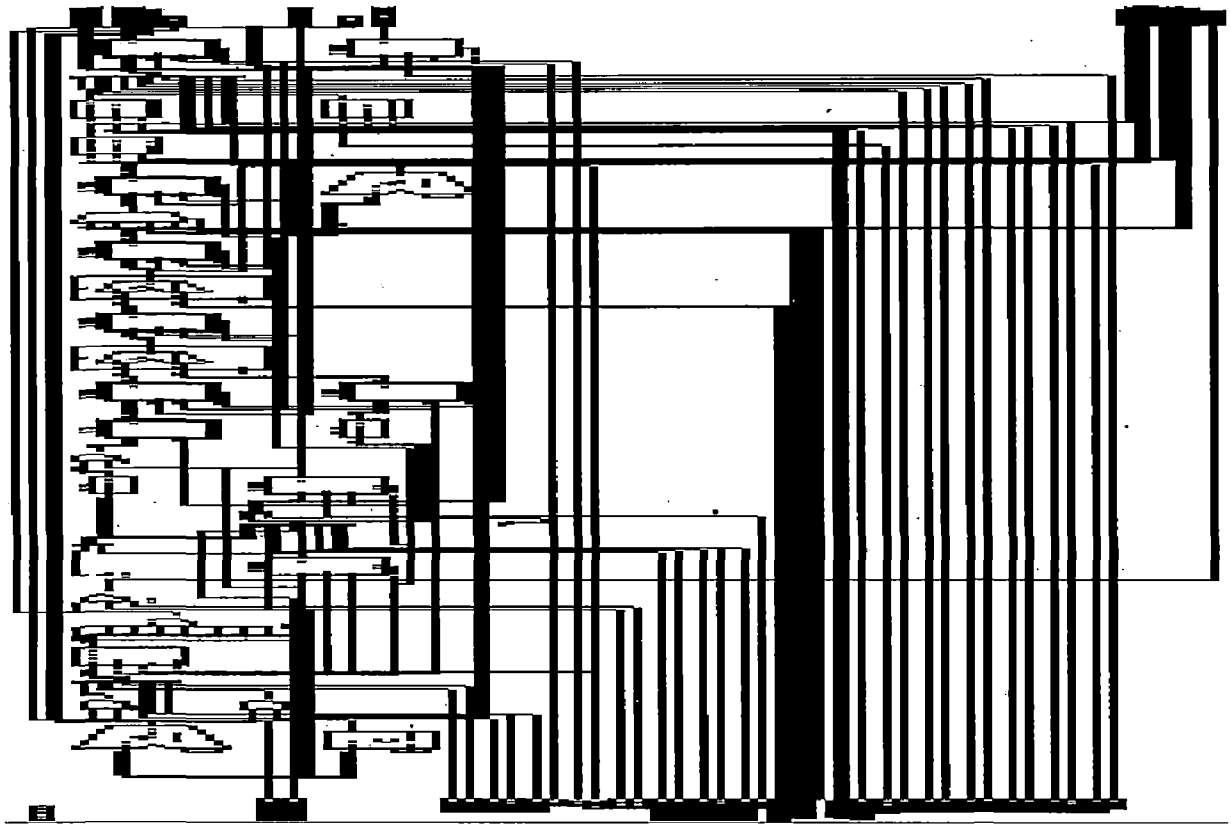


Figure 6.9 QUANTIZATION

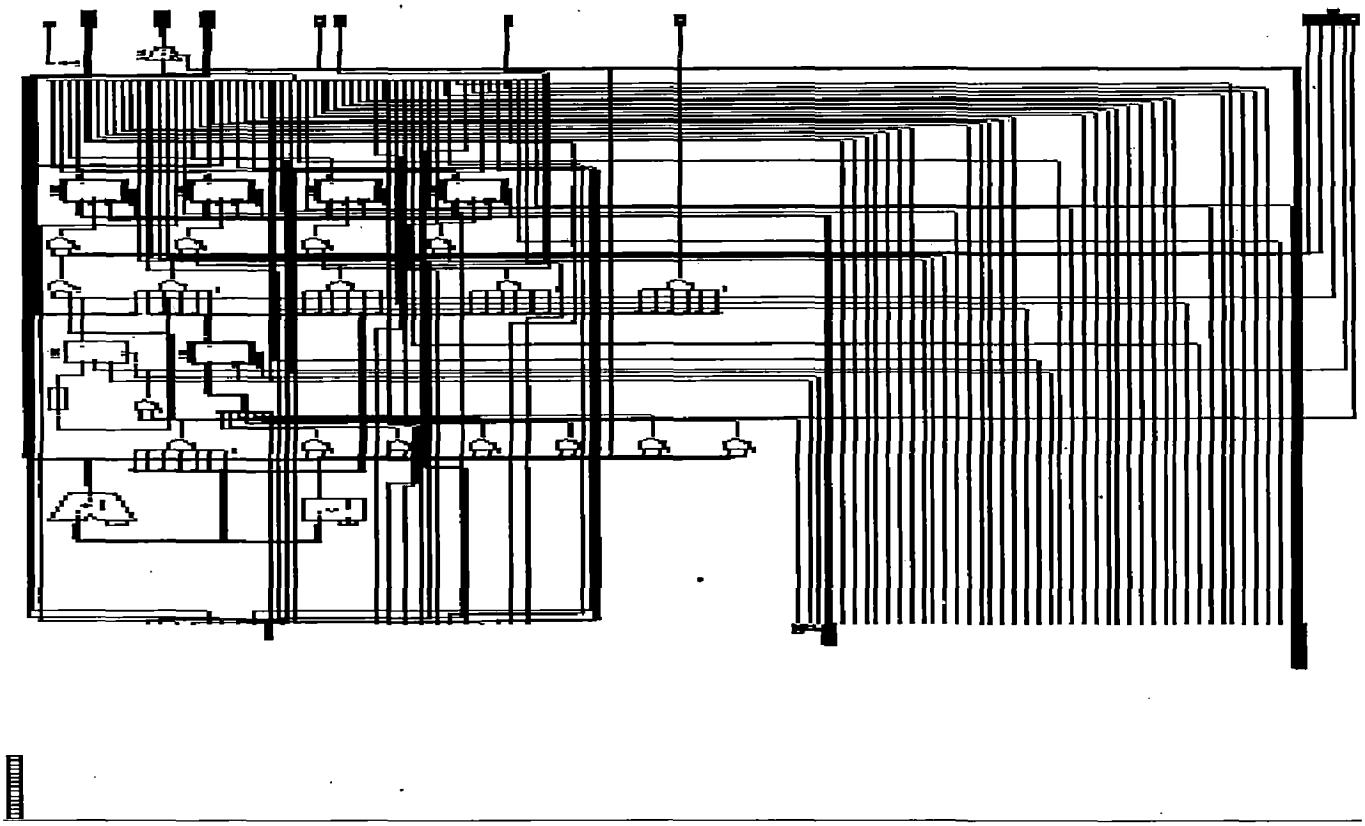


Figure 6.10 Zigzag Scan

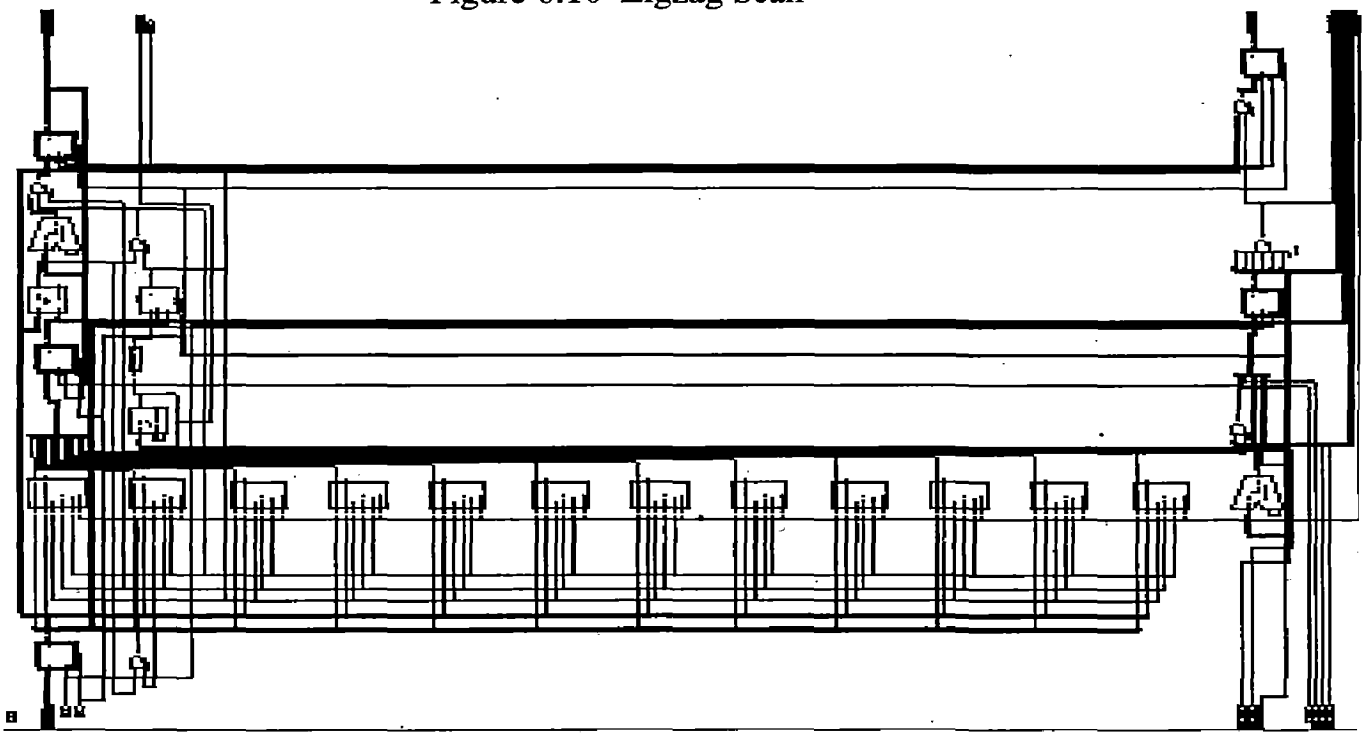


Figure 6.11 Run length

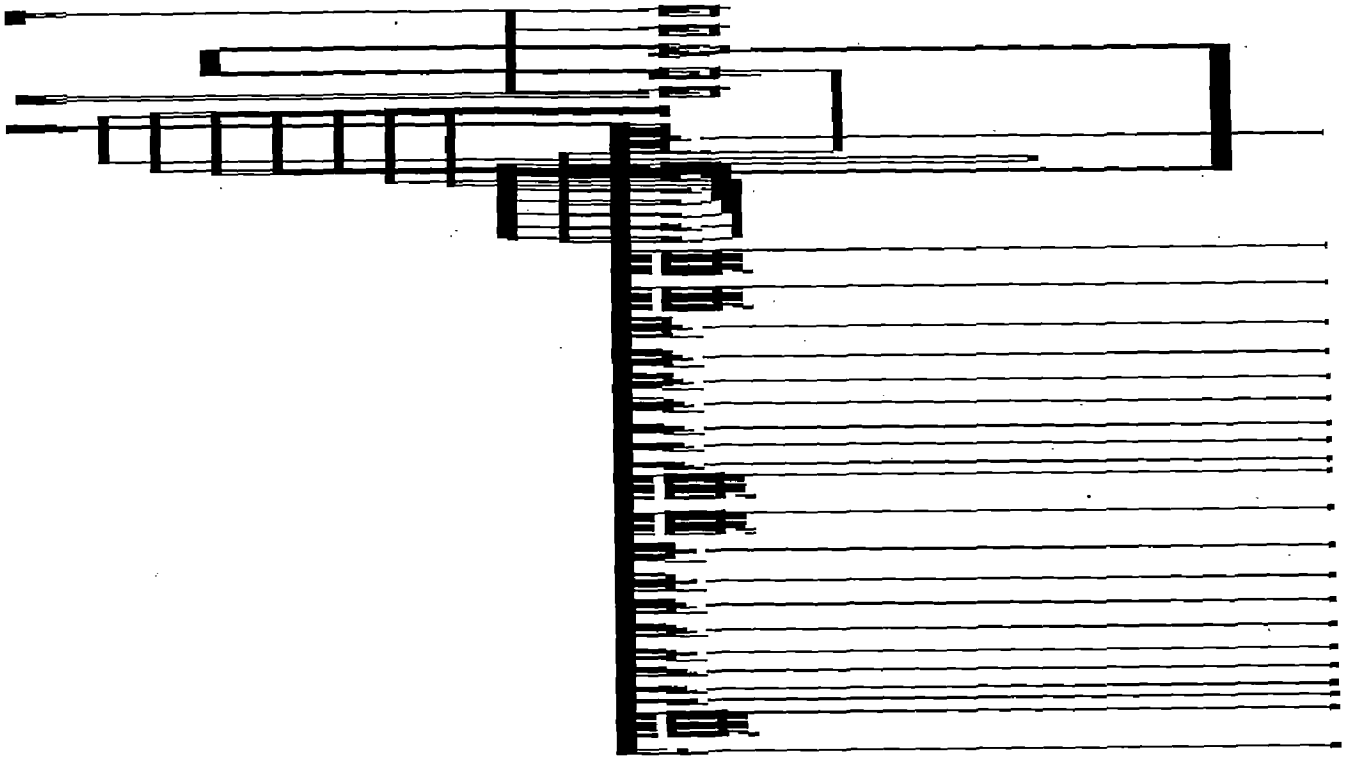


Figure 6.12 Huffman coding RTL Model

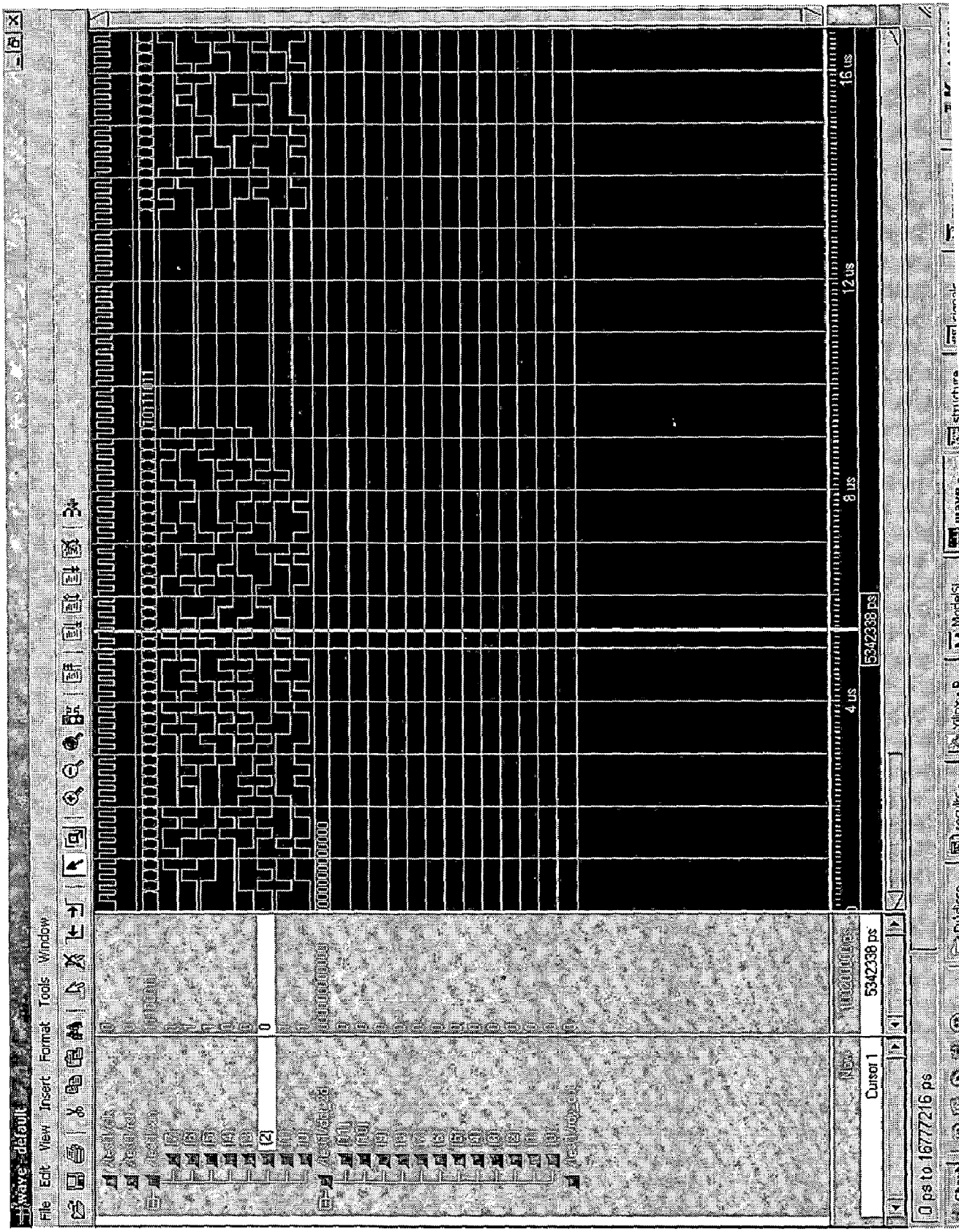


Figure 6.13 Initial Latency in DCT

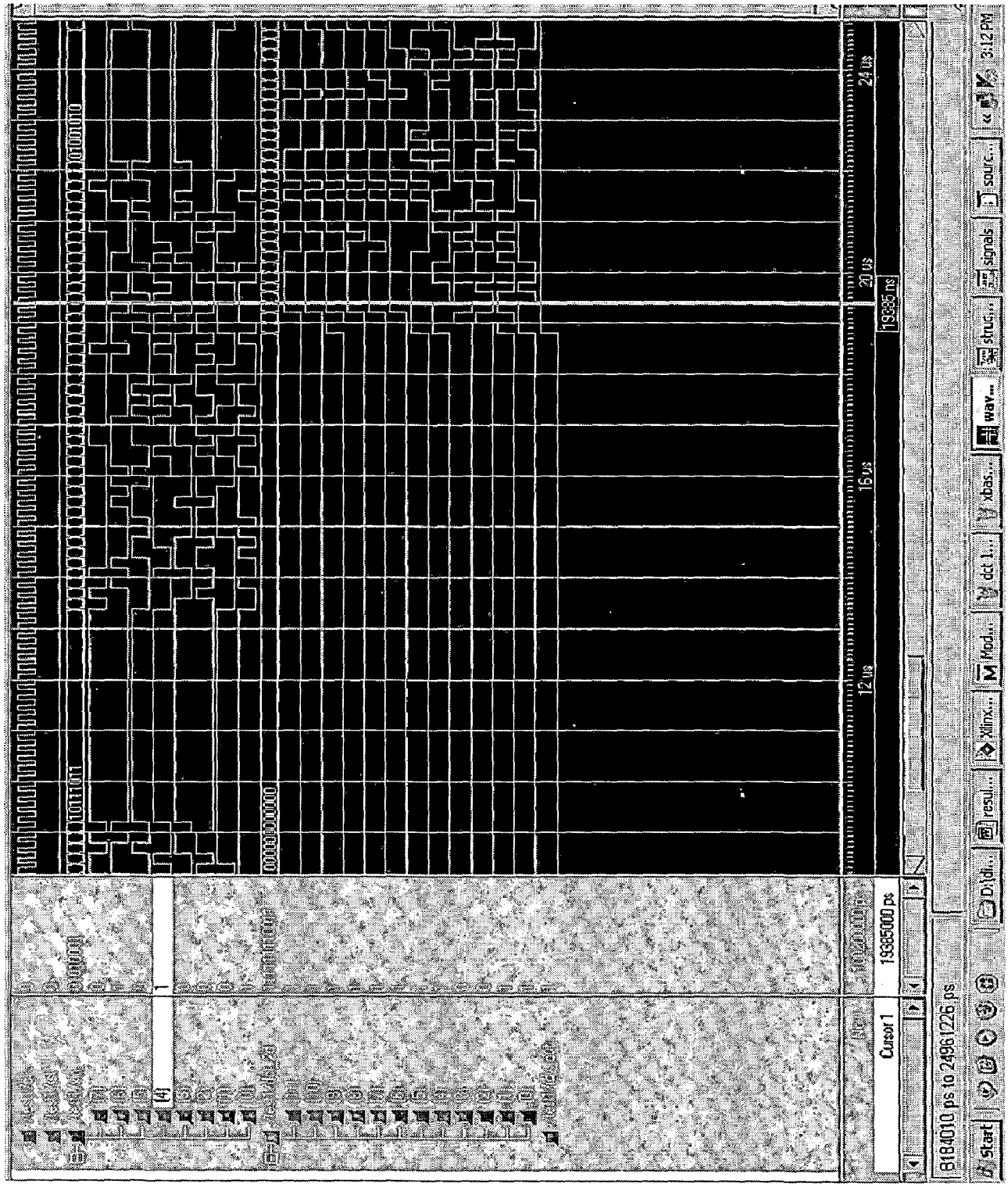


Fig 6.14 Output after initial Latency

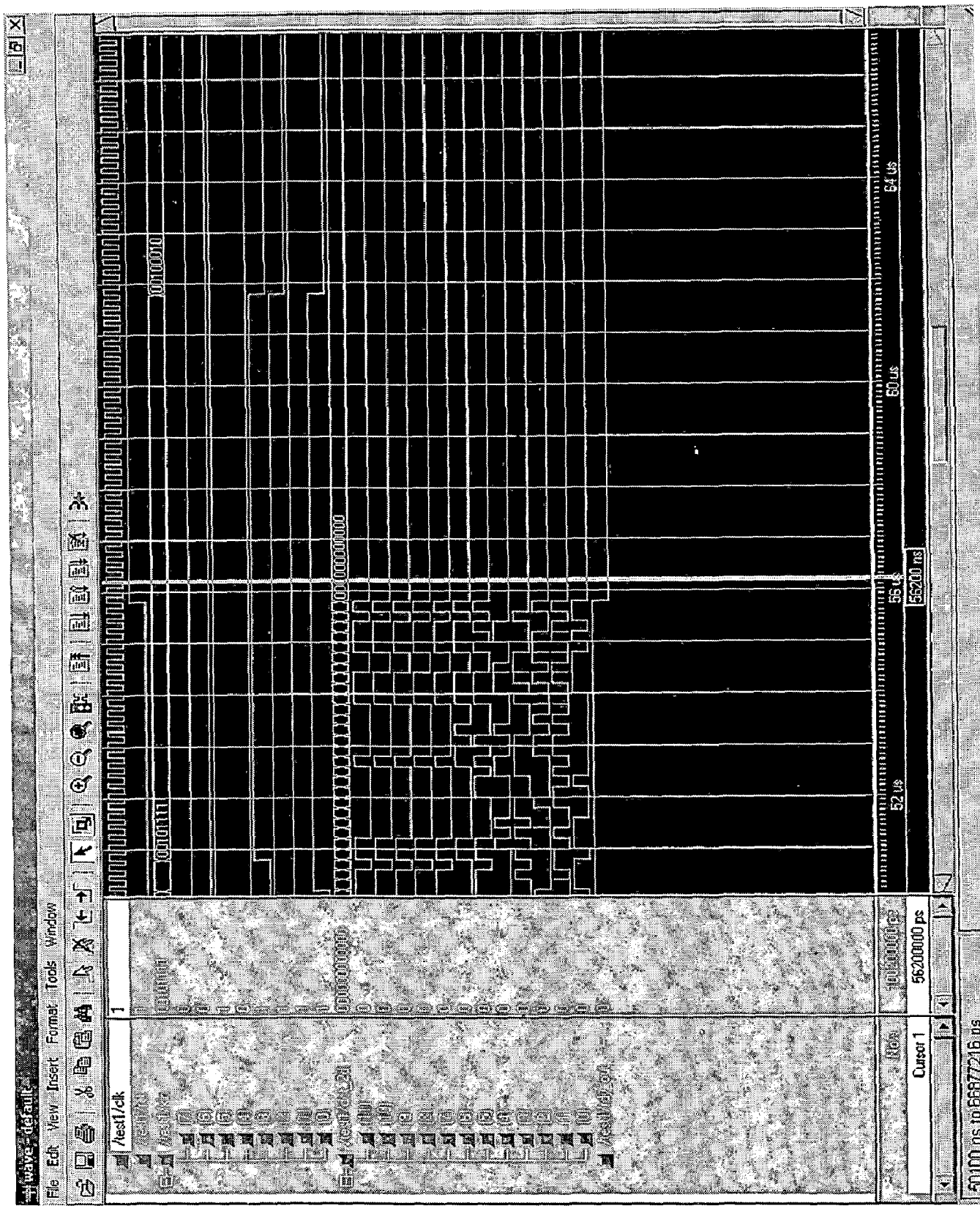


Figure 6.15 Reset signal being Activated

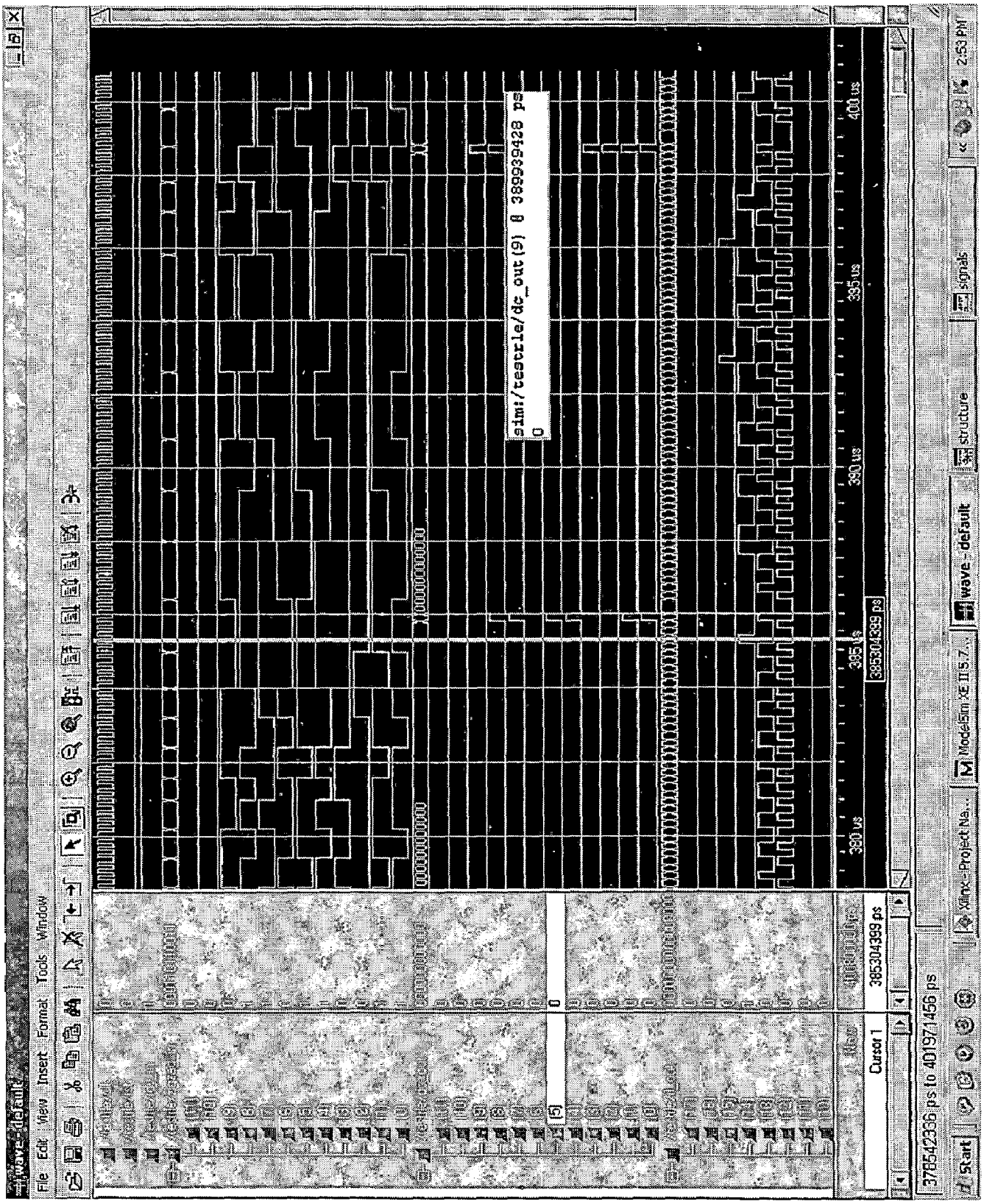


Figure 6.16 Run length Encode

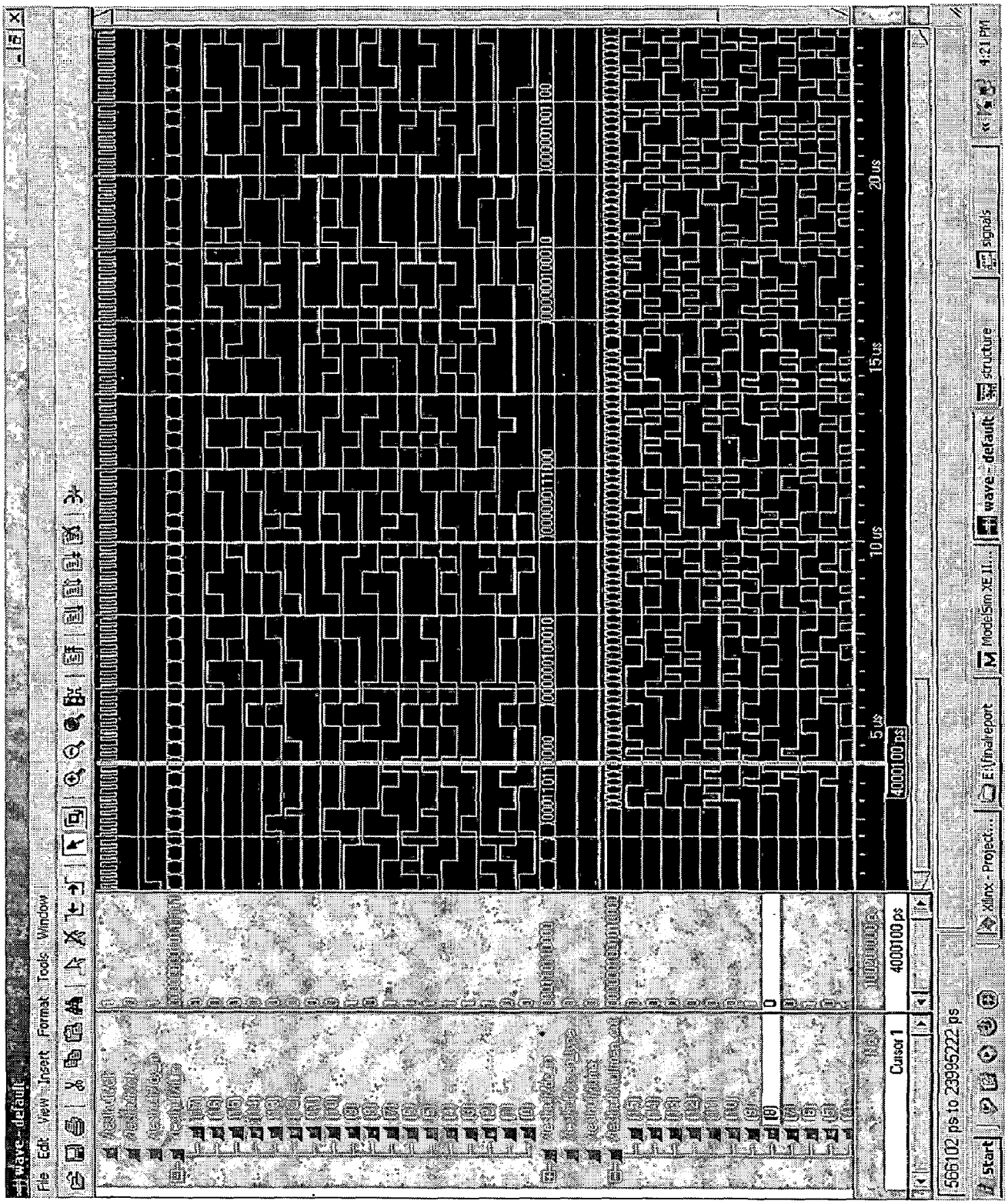


Figure 6.17 Huffman output

Discussion

1. DCT Implementation

1D-DCT is implemented on the input pixels first. The output of this called the intermediate value is stored in a RAM. The 2nd 1D-DCT operation is done on this stored value to give the final 2D-DCT output . The inputs are 8 bit vector wide and the 2d-dct outputs are 9 bit vector wide. The input signals are taken one pixel at a time in the order to follow from 0 to7 along the row from each column and in this way from 8 consecutive rows to read in a matrix form. These inputs are fed into a 8 shift register. The outputs of the 8 shift registers are registered by the 8 count CLK signal. This will enable us to register in 8 pixels (one row) at a time. The pixels are paired up in an adder subtractor in the order {0,7},{1,6},{2,5},{3,4}. For every clock, the adder/subtractor module alternately chooses addition and subtraction. This selection is done by the toggle flop. The output of the add sub is fed into a multiplier whose other input is connected to stored values in registers which act as memory. The outputs of the four multipliers are added at every CLK in the final adder. The output of the adder is the 1D-DCT values given out in the order in which the inputs were read in.

The outputs of the adder are stored in RAMs. Two RAM's are used so that data write can be continuous. The write operation continues for 64 clks . At the 65th clock, since output is continuous valid output data is obtained. These second sets of valid 1D-DCT coefficients are written into RAM2. So at 65th clk, RAM1 goes into read mode for the next 64 clks and RAM2 is in write mode. After this for every 64 clks, the read and write switches between the 2 RAMS.2nd 1D-DCT section. When RAM1 is full, the 2nd one dimensional calculation can start.

2.Quantization Implementation

The 2 dimensional DCT is taken through the quantizer where each of the DCT coefficients is quantized by dividing by the corresponding element in the quantizer

matrix. Quantization is done to zero out those DCT coefficients that correspond to the higher frequency components.

The Q_matrix is stored in a RAM. The ready input signal goes active when the first DCT output is available. The quantization starts when ready goes high and as long as it is high. An internal 64-count clock is used to loop through the 64 elements in the Q_matrix. Default Q matrix is stored in memory with its inverse values. In general quantizer scale code is used to select the quantizer scale. quantizer scale code value can range from 1 to 31. Here we have taken a fixed value for quantizer scale code.

Quantizer scale value for each quantize scale code is found in the MPEG ISO/IEC document. The Q matrix has 64, 8-bit values. The default quantized values are stored in a FIFO in zigzag order.

3. Zigzag Scan

The zigzag module is used to read out the quantized DCT input in a zigzag order. The default scanning mode is the mode used in MPEG1 is chosen in this design to convert two dimensional matrix into one dimension. MPEG2 consist of alternate mode as well. Here we have a latency of 64 clocks as all the coefficients are needed to be in the buffer at the time of scanning.

4. Run length Encode

The RUN LENGTH ENCODING module is used to convert the input stream of 64 coefficients into the run of zeroes and the value. The output of the zigzag module is fed into the RLE module. The counter is used to register the number of zeroes in the input. When an input is a non-zero value, the value is sent out along with the counter value. Counter counts up to 64 reading in each of the input value at every clock. The value read in when Counter content is '1' is the "dc" value. At the 64 count the End of buffer signal goes high showing that one set of coefficient has been processed.

5. Huffman Code

First we find the size for the DC Differential value. The Differential DC size gives the number of bits to be used to denote the variable length code and the corresponding code length for DC differential. Tables given in the appendix gives variable length code and corresponding code length for AC coefficients. Table 4 is used for intra blocks and table 5 is used for non-intra blocks. For a run/level pair not defined in table zero or one, an escape code followed by a 6 bit run symbol and 12 bit level is used. Thus, the maximum code can be 24 bit. Whereas minimum size can be two bits for end of block code.

6. Interfaces

For interfacing we used the FPGA into Slave Impatt mode in which image data is downloaded using PCI slot and PCI clock is used to implement the whole process. To take the data into the algorithm we first read it from external memory into the algorithm and the final result obtained is stored into another memory from which data is finally superimposed over raster horizontal and vertical pulses and displayed on to raster. Here main issue to be considered is the timing and memory address generation and increment.

With these implementation details we get an idea of how different sections are implemented in the VHDL flow to be downloaded over FPGA. In the next chapter we will conclude the thesis discussing the different objectives obtained in this attempt.

Chapter 7

Conclusion and Future Scope of Work

7.1 Conclusion

In this dissertation work an in depth study of the basic principles of image compression and their application to various areas in multimedia storage and transmission are discussed. A description of different mathematical formulation required for forming the base of compression algorithms and their physical significance are also discussed. Different terminologies defined in the standard are defined.

A detail description of MPEG standard and its variants is given, including MPEG 1, MPEG 2 and a brief introduction to MPEG 4 standard. The details included are the basic introduction to the standard, its variants, modifications made in the basic standard to accommodate various applications ranging from DVD storage to High Definition Television. It also describes how different level of coding is done in order to achieve better quality video maintaining backward compatibility with the previously existing standard.

Finally, different algorithms required to built the whole standard and their detail implementation flow in VHDL is discussed. A detail of device used for implementation and interface of the entire algorithm with the external memories and display are also discussed.

1)These algorithms when simulated on ModelSim gives significant improvement as compare to one run in software using CPU cycles, thus, providing a faster implementation of streaming multimedia algorithms for data compression.

2) When implementing motion estimation and prediction multiple frames and thus, multiple memories are required to be taken care of, thus, requiring separate controller dedicated for generating multiple control signals and interrupts making implementation more complex.

3) Test implementation with single frame have been encouraging but with multiple frames there are serious artifacts of synchronization.

7.2 Future Scope of Work

Following are the areas in which further scope can be found

1) These algorithms have been implemented using the simplest parallel pipelining architecture but can be modified to introduce more stages, thus, improving the implementation speed.
2) This implementation is based on receiving data through PCI interface using Slave serial mode but this can be done through dedicated video input and output interfaces provided on the video application board.
3) For multiple frame handling a separate processor can be included in the form Power PC in the system making it a standalone embedded system.
4) With the new support board launched recently for MPEG encoder implementation many new features can be induced to implement higher MPEG 4 standard where separation of Video object planes is required.
5) These algorithms when integrated with a tracking system can be built into a system that can be applied for fast capturing cum processing of real time data.

References

[1] Vijayendra Mohan Agrawal, "A STUDY OF COMPRESSIION USING MPEG STANDARD", M.Tech Thesis, University of Roorkee, February, 1997.

[2] Khurram Zaka Bukhari, "Visual Data Transforms Comparison", Master's Thesis, Delft University, Netherlands, August 2002.

[3] Peter D.Symes, "Video Compression", McGraw Hill, New York.

[4] ISO/IEC 13818-2: 1995 (E), "Recommendation ITU-T H.262 (1995 E)".

[5] A.V. Oppenheim, R.W.Schaffer, John R. Buck, "Discrete Time Signal Processing", Pearson Edition,2004.

[6] "MPEG Digital Video Coding Standard", IEEE Signal Processing Magazine, September, 1997

[7] Gregory K. Wallace ,"THE JPEG STILL PICTURE COMPRESSION STANDARD,"IEEE Transactions on Consumer Electronics, Vol. 38, No. 1, February 1992.

[8] Thomas Sikora," The MPEG-4 Video Standard Verification Model," IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, VOL. 7, NO. 1, FEBRUARY 1997.

[9] Ut-Va Koc and K. J. Ray Liu," DCT-Based Motion Estimation," IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 7,NO. 7, JULY 1998

[10] PO-RONG CHANG AND CHIN-FENG LIN," Wireless ATM-Based Multicode CDMA Transport Architecture for MPEG-2 Video Transmission ," PROCEEDINGS OF THE IEEE, VOL. 87, NO. 10, OCTOBER 1999.

[11] Xilinx Application Notes, (a) "Xapp208", (b) "Xapp256", (c)"Xapp610", (d)"Xapp615", (e) "Xapp131", (f)"Xapp194", (g)"Xapp616", (h)"Xapp682", (i)"Xapp759", (j)"Xapp621", www.xilinx.com (web resource).

[12] Olivier Cantineau Brian Jentz , "Enabling Real-Time JPEG 2000 with FPGA Architectures," Barco Silex, Altera.

[13] "Model Sim," Xilinx Edition II, Tutorial.

[14] Tomas Kratochvil, "VIDEO COMPRESSION HARDWARE IMPLEMENTATION USING PROGRAMMABLE MEDIA PROCESSOR," 2003,4th EURASIP Conference focused on Video Image Processing and Multimedia Communications Zagreb, Croatia.

[15] Simon Haykin, "Communication System", Wiley Eastern, Fourth Edition, 2004.

[16] John Watkinson, "The MPEG-4 Handbook", Focal Press,2002.

[17] "Operation Manual for Virtex II Pro Video Board", Mechatronics Test Equipments.

[18] Rafael C. Gonzales, Richard E. Woods "Digital Image Processing", Pearson Edition, 2003.

Appendix A

Table 1 Differential DC Additional Code [4]

Differential DC	Size	Additional Code
-2047 to -1024	11	00000000000 to 01111111111
-1023 to -512	10	0000000000 to 0111111111
-511 to -256	9	000000000 to 011111111
-255 to -128	8	00000000 to 01111111
-127 to -64	7	0000000 to 0111111
-63 to -32	6	000000 to 011111
-31 to -16	5	00000 to 01111
-15 to -8	4	0000 to 0111
-7 to -4	3	000 to 011
-3 to -2	2	00 to 01
-1	1	0
0	0	
1	1	1
2 to 3	2	10 to 11
4 to 7	3	100 to 111
8 to 15	4	1000 to 1111
16 to 31	5	10000 to 11111
32 to 63	6	100000 to 111111
64 to 127	7	1000000 to 1111111
128 to 255	8	10000000 to 11111111
256 to 511	9	100000000 to 111111111
512 to 1023	10	1000000000 to 1111111111
1024 to 2047	11	10000000000 to 11111111111

Table 4 DCT Coefficient Table Zero (Excerpt) [4]

Variable Length Code (Note 1)	Run	Level
10 (Note 2)	End of Block	
1 s (Note 3)	0	1
11 s (Note 4)	0	1
011 s	1	1
0100 s	0	2
0101 s	2	1
0010 1 s	0	3
0011 1 s	3	1
0011 0 s	4	1
0001 10 s	1	2
0001 11 s	5	1
0001 01 s	6	1
0001 00 s	7	1
0000 110 s	0	4
0000 100 s	2	2
0000 111 s	8	1
0000 101 s	9	1
0000 01 s	Escape	

Table 5 DCT Coefficient Table One (Excerpt) [4]

Variable Length Code (Note 1)	Run	Level
0110 (Note 2)	End of Block	
10 s	0	1
010 s	1	1
110 s	0	2
0010 1 s	2	1
0111 s	0	3
0011 1 s	3	1
0001 10 s	4	1
0011 0 s	1	2
0001 11 s	5	1
0000 110 s	6	1
0000 100 s	7	1
1110 0 s	0	4
0000 111 s	2	2
0000 101 s	8	1
1111 000 s	9	1