

Efficient H.264 Encoder with High Profile for High Frame Rate Video Streams



Submitted by:

Fariaa Faheem	2019-EE-001
Marwa Waseem	2019-EE-007
Hamza Akhtar	2019-EE-012
Filza Shahid	2019-EE-151

Supervised by: Dr. Muhammad Tahir

Department of Electrical Engineering
University of Engineering and Technology Lahore

Efficient H.264 Encoder with High Profile for High Frame Rate Video Streams

Submitted to the faculty of Electrical Engineering
of the University of Engineering and Technology Lahore
in partial fulfillment of the requirements for the Degree of

Bachelor of Science
in
Electrical Engineering.

Internal Examiner

External Examiner

Director
Undergraduate Studies

Department of Electrical Engineering
University of Engineering and Technology Lahore

Declaration

I declare that the work contained in this thesis is my own, except where explicitly stated otherwise. In addition this work has not been submitted to obtain another degree or professional qualification.

Signed: _____

Date: _____

Acknowledgments

The acknowledgements and the people to thank go here, don't forget to include your project advisor...

For/Dedicated to/To my...

Contribution to Sustainable Development Goals

Write a short description of how the work carried out in this thesis has contributed to some of the sustainable development goals listed below.

1. No poverty
2. Zero hunger
3. Good health and well-being
4. Quality education
5. Clean water and sanitation
6. Affordable and clean energy
7. Decent work and economic growth
8. Industry, innovation and infrastructure
9. Sustainable cities and communities
10. Responsible consumption and production
11. Climate action
12. Life below water
13. Life on land
14. Peace, justice and strong institutions

Contents

Acknowledgments	iii
Contribution to Sustainable Development Goals	v
List of Figures	viii
List of Tables	ix
Abbreviations	x
Abstract	xi
1 Introduction	1
1.1 H264 Profiles	1
1.1.1 Baseline Profile	1
1.1.2 Main Profile	2
1.1.3 High Profile	2
1.2 H264 Process	2
1.2.1 Forward Path	3
1.2.2 Reconstruction Path	3
1.3 H264 Major Components	3
1.3.1 Prediction	3
1.3.2 Intra-Prediction	4
1.3.3 Inter-Prediction	4
1.3.3.1 Motion Vector Prediction	4
1.3.4 Transform	5
1.3.5 Quantization	5
1.3.6 Entropy Encoding	5
1.4 H264 syntax	6
2 Motivations and Problem Statement	8
2.1 Motivations	8
3 Prediction, Transformation, Quantization and Encoding	9
3.0.1 Intra-Prediction	9
3.0.1.1 4x4 Luma Prediction	9
3.0.1.2 8x8 Chroma Prediction	11
3.0.2 Inter-Prediction	11
3.0.2.1 Block Matching Technique	11

4	Hardware Architecture for Inter-Prediction	13
5	Thesis Structure	15
A	Introduction to Latex	17
A.1	Learning \LaTeX	17
A.1.1	A (not so short) Introduction to \LaTeX	17
A.1.2	A Short Math Guide for \LaTeX	17
A.1.3	Common \LaTeX Math Symbols	18
A.1.4	Figures	18
A.1.5	Typesetting mathematics	19
A.2	Sectioning and Subsectioning	20
	References	21

List of Figures

1.1	Top Level Diagram for H.264 Encoder	3
1.2	Intra-Prediction	4
1.3	Inter-Prediction	4
1.4	Motion Vector Prediction	5
1.5	Overall syntax for h264 encoder	6
3.1	Reference samples for 4x4 Luma	10
3.2	Major Modes for 4x4 Luma Prediction	10
3.3	Major Modes for 8x8 Chroma Prediction	11
3.4	Motion Estimation using Block Matching	12
4.1	265 PE VBS ME Hardware Architecture	14
A.1	The UET Lahore logo.	18

List of Tables

Abbreviations

LAH List Abbreviations **Here**

Abstract

The Thesis Abstract is written here (and usually kept to just this page). The page is kept centered vertically so can expand into the blank space above the title too...

Chapter 1

Introduction

Digital video refers to a sequence of images displayed on a screen at a predetermined rate. The process of transferring video content from one medium to another depends on both the duration of the video and the size of the encoded bits. **Video compression** techniques are commonly utilized to reduce the overall size of the video, which consequently leads to a decrease in the amount of data required for transmission and storage of digital video signals.

The most current video coding standard is known as **H.264/AVC or MPEG-4 Part-10**. This standard was jointly developed by the **ITU-T Video Coding Experts Group** and **ISO/IEC JTC 1 Moving Picture Experts Group**. Compared to typical video coding standards, H.264/AVC provides significantly higher efficiency, capable of reducing bit rate requirements by up to 50% while maintaining the same level of video quality. It is designed to cover a wide range of video resolutions, from QCIF to HDTV.

1.1 H264 Profiles

The H.264 family of standards includes various capabilities. These profiles are mainly used to reduce the frame count by implementing motion prediction and temporal compression. The most common ones are:

- Baseline Profile
- Main Profile
- High Profile

1.1.1 Baseline Profile

In the realm of video encoding, baseline profiles are commonly employed in applications that require low-power consumption and cost-efficiency. These profiles are capable of achieving an impressive compression ratio of 1000:1, resulting in a streamlet of 1 Gbps being compressed down to approximately 1 Mbps. Baseline profiles utilize a 4:2:0

chrominance sampling method, whereby color information is sampled at half the vertical and horizontal resolution of the black-and-white information. This technique enables the reduction of data without significantly impacting the overall quality of the video.

Furthermore, Universal Variable Length Coding (UVLC) and Context Adaptive Variable Length Coding (CAVLAC) are employed as the primary entropy encoding techniques within this profile. Such encoding methods contribute significantly to the efficient compression of video data, while ensuring that the encoded video stream is in compliance with relevant standards.

1.1.2 Main Profile

Significant enhancements were made to the Baseline Profile through the introduction of advanced frame prediction algorithms, resulting in the development of the Main Profile. This updated profile is primarily utilized for standard-definition digital TV broadcasts in MPEG-4 format.

However, it should be noted that the Main Profile is not employed in high-definition broadcasts. Rather, alternate profiles are used in such scenarios to ensure optimal video quality and compatibility with relevant standards.

1.1.3 High Profile

Introduced in the year 2004, the High Profile is considered to be the most efficient and powerful profile within the H.264 family. It is primarily utilized in high-definition television applications such as Blu-ray Disc storage and DVB HDTV broadcast services. This profile is capable of achieving an exceptional compression ratio of 2000:1, which is a significant improvement over previous encoding standards. It utilizes an adaptive transform method that allows for the selection of either 4x4 or 8x8 pixel blocks. This enables preservation of video quality while reducing network bandwidth consumption by up to 50 percent.

Furthermore, the application of this compression technique facilitates the compression of a 1 Gbps stream to approximately 512 Kbps, further emphasizing the impressive capabilities of the High Profile.

The overall procedure of H.264 includes various components. The top level block diagram of an H.264 Encoder is shown in Figure 1.1.

1.2 H264 Process

An H264 encoder has a **forward path** and a **reconstruction path**. The forward path uses **intra** and **inter predictions** to encode a video frame to create a bit stream. The reconstruction path is used to decode the encoded frame and to reconstruct the decoded frame. Reconstruction path in encoder ensures that both encoder and decoder make use of similar reference frames for inter and intra prediction. This is to avoid encoder-decoder mismatches.

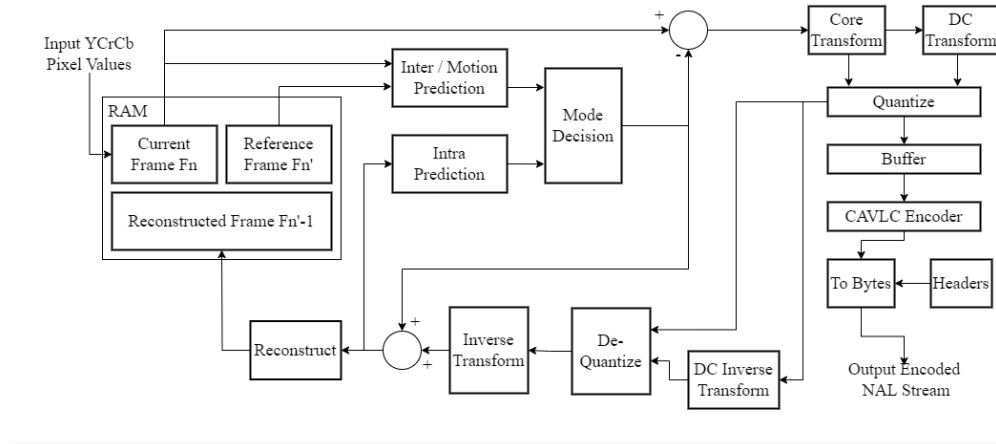


FIGURE 1.1: Top Level Diagram for H.264 Encoder

1.2.1 Forward Path

The input frame is partitioned into **Macro-Blocks (MB)**. These MB are then encoded in intra or inter mode. This depends on mode decision. The current MB is predicted from reconstructed frame. This predicted MB is generated by intra prediction based on **spatial redundancy**, and by inter prediction based on **temporal redundancy**. The mode is chosen based on better quality and bit rate performance of these 2 modes. The Predicted MB is subtracted from current MB to create a **Residual MB**. Residual data is **transformed** (4x4 integer transform), then **quantized**. The obtained coefficient are re-ordered in a **zig-zag order** which are regarded as **entropy encoded**. These coefficients along with header information form the **compressed bit stream**. This stream is forwarded to NAL for storage or transmission.

1.2.2 Reconstruction Path

This path takes quantized transform coefficients and performs inverse quantization and inverse transform. In this way, reconstructed residual data is generated, but they are not identical to original residual data as quantization is a lossy process. In order to create the reconstructed frame, the reconstructed residual data are added to predicted pixels.

1.3 H264 Major Components

1.3.1 Prediction

In order to guarantee a high compression ratio in H.264 encoders, prediction is a technique utilized. In prediction, a 16x16 pixel block known as a macroblock from a previous video frame or the present frame is utilized to forecast macroblocks in the current frame. Detailed information on predictions and its types are given in chapter 3.

There are basically 2 modes for prediction:

1.3.2 Intra-Prediction

Intra prediction is performed without referring to any data outside the current slice i.e prediction from previously coded data in the same slice. It reduces spatial redundancies by exploiting spatial correlation between adjacent blocks in a given picture. There are 3 choices of block size for luma component i.e. 16x16, 8x8 or 4x4. Whereas for chroma component, a single prediction block is generated. Once, the prediction has been made, it is subtracted from current block to make a residual. The overview of the process is shown in figure 1.2.

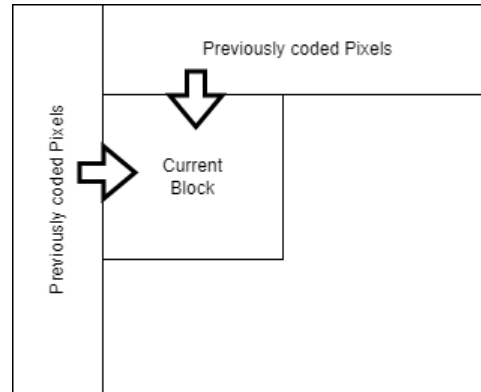


FIGURE 1.2: Intra-Prediction

1.3.3 Inter-Prediction

Inter prediction is the process of predicting a block of luma and chroma samples from a picture that has been previously coded and transmitted i.e reference picture. It uses temporal sampling technique. For this, a prediction region is selected, then a prediction block is generated. After that the prediction block is subtracted from original block of samples to form a residual. The overview of the process is shown in figure 1.3.

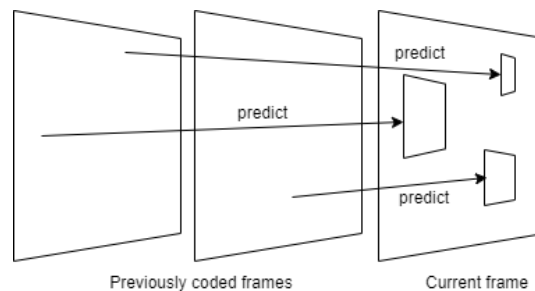


FIGURE 1.3: Inter-Prediction

1.3.3.1 Motion Vector Prediction

In video coding, motion vectors for neighboring partitions are typically closely related, and so each motion vector can be predicted using previously coded vectors from nearby partitions. A predicted vector is created based on these previous motion vectors, and the

difference between the current vector and the predicted vector is encoded and transmitted. The way in which predicted vector is predicted depends on the size of the motion compensation partition and whether there are nearby vectors available for reference.

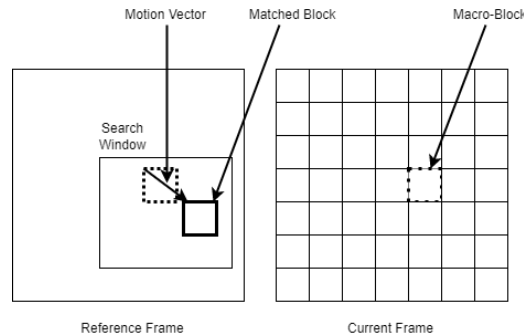


FIGURE 1.4: Motion Vector Prediction

1.3.4 Transform

Once the residual data, which essentially comprises a block of residual coefficients, is obtained, it is subjected to the core transform process. This transform is an integer-based 4x4 or 8x8 transform that provides a scaled approximation to the Discrete Cosine Transform (DCT). In certain cases, a portion of the output from this integer transform is subject to further transformation through a DC Transform, referred to as the Hadamard transform. The same residual data can be reconstructed using the DC inverse transform, which is carried out prior to rescaling. Finally, the rescaled coefficients are inverse transformed using a 4x4 or 8x8 inverse integer transform.

1.3.5 Quantization

The transformed coefficients are subjected to quantization using a non-uniform quantizer. In this process, each coefficient is divided by an integer value, which reduces the precision of the coefficient values as determined by the Quantization Parameter (QP). The use of a non-uniform quantizer results in a smaller number of bits being used to represent each coefficient value, which in turn reduces the amount of data required to represent the video. The quantized transform coefficients of a block are typically scanned in a zig-zag pattern, which is a common approach for video coding standards.

1.3.6 Entropy Encoding

In H.264 stream or file encoding, the symbols are coded in a series. The quantized transform coefficients are efficiently transmitted using the Context-Adaptive Variable Length Coding (CAVLC) method. The statistical distribution of the quantized transform coefficients typically shows larger values for low-frequency components that decrease gradually towards the high-frequency part. As a result, the number of nonzero quantized coefficients (N) and their size and position are coded separately. This enables the receiver to reconstruct the original signal more accurately. The coefficients are scanned in a zig-zag

pattern, and then quantized to reduce their precision using the quantization parameter (QP). The efficient transmission of quantized transform coefficients in H.264 helps in reducing the size of the compressed video file while maintaining the quality of the video. The final Network Abstraction Layer can be seen in figure 1.5

1.4 H264 syntax

H.264 consists of 2 layers: the **Network Abstraction Layer (NAL)** and the **Video Coding Layer (VCL)**. The NAL consists of a series of NAL Units, with **Sequence Parameter Sets (SPS)** and **Picture Parameter Sets (PPS)** being the most common units that signal certain control parameters to the decoder. In the VCL, coded video data is communicated in the form of slices. An **access unit**, which can be a coded frame or field, is made up of one or more slices. Each slice consists of a Slice Header and Slice Data, with the latter being a series of coded **macro blocks (MB)** and skip macro block indicators signaling that certain macro block positions contain no data.

- **MB type:** I/intra coded, P/inter coded from one reference frame
- **Prediction information:** prediction mode for I macro block, choice reference frame and motion vectors for P macro block
- **Coded Block Pattern CBP:** indicates which luma and chroma blocks contain non zero residual co-efficient
- **Quantization Parameter QP:** for macro blocks with CBP not 0
- **Residual Data:** for blocks containing non-zero residual coefficients

The basic H264 syntax can be seen in figure 1.5

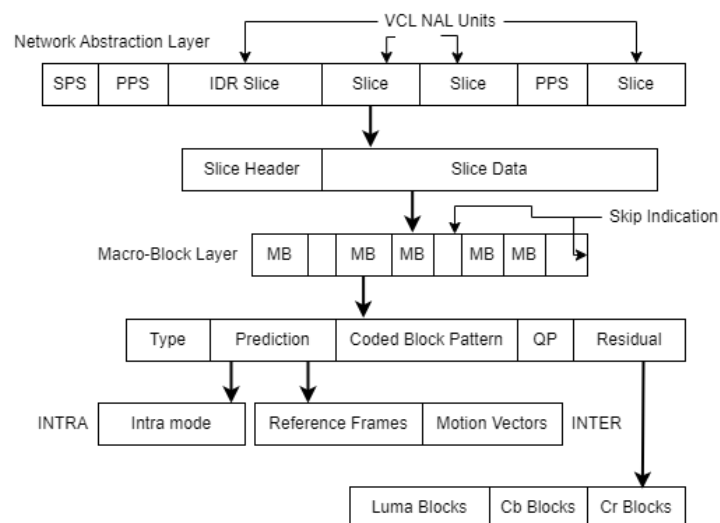


FIGURE 1.5: Overall syntax for h264 encoder

(write something about hardware implementation on FPGA) In this thesis, we developed an FPGA based H.264 intra and inter frame coder hardware targeting **High Profile** (see which one to mention, See from wikipedia)

(write about what is happening in the next chapters)

Chapter 2

Motivations and Problem Statement

2.1 Motivations

(add the relevant info related to our project reason open source camera chip - must , 10x (isp), us (encoder))

With the rapid increase in the development of products and benefits offering full-motion digital video, digital video coding is currently gaining importance and has a considerable monetary impact on computer, imaging and telecommunications technology. Throughout the early phases of technology, there have been many international video coding standards as engines behind the commercial success of digital video compression. In comparison to current existing standards, H264 has many new features that makes it more emphatic. The important features that distinguish H264 from other standards are good video quality at high and low bit rates and network friendliness.

It is proposed that H264 can deliver two times better performance than the previous MPEG-2 coding standard, both in terms of compression efficiency and picture eminence. Moreover, previous H.63 and MPEG-4 implement block-based motion estimation to reduce temporal redundancy between frames. But in H.264, block matching efficiency is improved by some innovative features such as variable block size motion estimation (VBS_ME) and motion vector prediction. The computational complexity of this model is increased by the factor of four due to these features.

Chapter 3

Prediction, Transformation, Quantization and Encoding

Prediction is defined as duplication of the information contained in a macro-block using previously coded data. This duplicated information is subtracted from the macro-block to form a residual. There are 2 types of prediction.

- Intra-Prediction
- Inter-Prediction

3.0.1 Intra-Prediction

Intra-prediction utilizes the space dependency to compress the video. The frames which are intra coded using intra-prediction are called I-frames. Following are the possible prediction modes:

- **4x4 luma:** having 9 directional modes and is suitable for macro blocks that has lot of details
- **8x8 luma:** having 9 directional modes and is for high profiles only.
- **16x16 luma:** having 4 directional modes that is suitable for macro block with smoother area
- **8x8 chroma:** 4 possible prediction modes and used for chrominance components

In our model, 4x4 luma prediction and 8x8 chroma prediction is being implemented.

3.0.1.1 4x4 Luma Prediction

For this type of prediction, each macro block that is of **16x16** (256 pixels each of which is 8 bit wide) is divided into **4x4** block (16 pixels). Figure 3.1 shows the reference samples for 4x4 luma prediction. 4 pixels **A,B,C,D** (adjacent to current block) of block

a on top of current block, pixels **E,F,G,H** of block **b** on top right corner, **I,J,K,L** of block **c** at adjacent left of current block and 1 pixel **M** of **d** block on top left corner are used for prediction of 16 pixels in the current block.

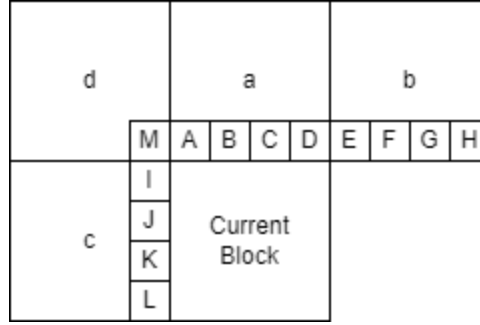


FIGURE 3.1: Reference samples for 4x4 Luma

There are total 9 prediction modes that are supported in this prediction. Major modes that are implemented in our model are as follows:

- **Mode 0 (Vertical):** The predicted block is constructed by using upper samples A,B,C,D of block ‘a’ as shown in figure 3.2. They are extrapolated vertically. It is suitable to predict vertical edges in the block.
- **Mode 1 (Horizontal):** In this mode, left samples I,J,K,L of block c are used. They are extrapolated horizontally and is suitable for horizontal edges. It can be seen in figure 3.2.
- **Mode 2 (DC):** It utilizes average of all adjacent samples (A to D and I to L) to form the prediction of current block. It is suitable for smooth areas. Its process is shown in figure 3.2.

For the details of remaining modes refer to [give ref to richardson book here]. Figure 3.2 display the above 3 prediction modes. To create a predict sample, every color stands for a particular formula. The encoder determines each prediction direction’s cost by finishing processing for all of the prediction directions, then outputs the one with the lowest cost.

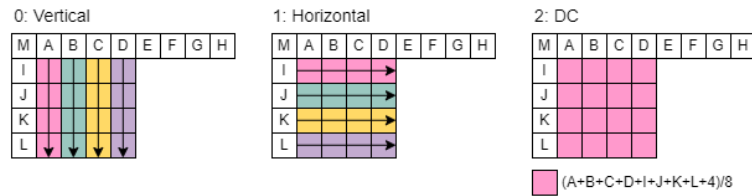


FIGURE 3.2: Major Modes for 4x4 Luma Prediction

3.0.1.2 8x8 Chroma Prediction

This type of prediction applies on chrominance components. It is similar to 16x16 luma prediction which can be referred in [Richardson book] except the block size is 8x8 and there is different order of mode number which are:

- Mode 0: DC
- Mode 1: Horizontal
- Mode 2: Vertical
- Mode 3: Plane

The working of first 3 modes is similar to mode 2,1,0 of 4x4 luma prediction respectively. For details of Plane mode refer to [Richardson book]. The Implemented mode in our model is DC. These modes are shown in the figure 3.3.

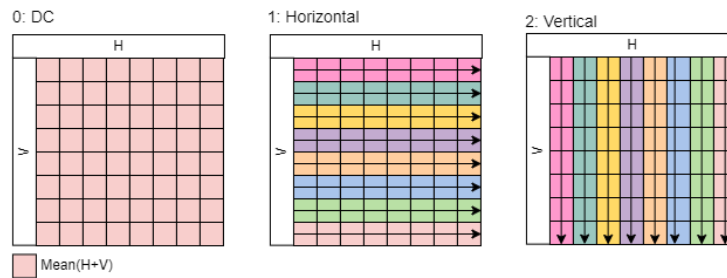


FIGURE 3.3: Major Modes for 8x8 Chroma Prediction

[Explain the hardware architecture of Intra prediction if possible]

3.0.2 Inter-Prediction

The process of predicting a block of luma and chroma samples from a reference picture that has been previously been coded and transmitted i.e. exploits temporal redundancy between successive frames. For this a prediction region is selected, a prediction block is generated and then it is subtracted from original block of samples to form a residual. This is then coded and transmitted. Reference pictures are stored in Decoded Picture Buffer. The offset between position of current block and search region in the reference picture is called motion vector. This prediction is also known as **Motion Estimation** [ref of ppr Low power techniques]. It has the capability of extracting true motion information thus enhancing the quality of displayed images in video enhancement systems. The preferred technique for motion estimation is the **Block Matching (BM)** Technique.

3.0.2.1 Block Matching Technique

This method divides the current frame into non-overlapping NxN macro-blocks and seeks out the block from the reference frame that most closely resembles the current block

within a specified search range. The Sum of Absolute Difference (SAD), which is appropriate for hardware implementations, is the recommended block matching criterion.

In figure 3.4, (x,y) represents the location of the current frame. The search window in the reference frame is in $[-r,r]$ region in both x and y directions. Both current and reference block lies within the range of search window. The SAD value is calculated by accumulating absolute differences of corresponding pixels in both current and reference blocks. A motion vector is the relative motion of current block in reference frame, they are specified in relative coordinates. Thus if $(x+a,y+b)$ is the location of best matching block in reference frame, then (a,b) represents the motion vector. Motion Estimation is performed on luma component and resulting motion vectors are also used for chroma components.

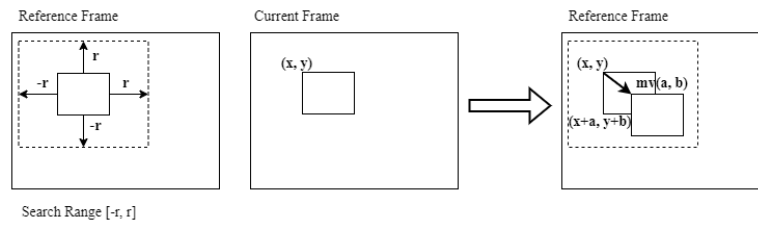


FIGURE 3.4: Motion Estimation using Block Matching

There are several algorithms for Block Matching. Among them mostly used is **Full Search (FS)** algorithm. The SAD values for each search position within a specific search range are calculated by this approach to determine the reference block that most closely resembles the present block. It has the best performance related to other algorithms as it searches all the search locations in a given search range. But its computational complexity is high and its hardware consume a lot of power. For further improvement, instead of **fixed block size (FBS) FS ME** algorithm, **variable block size (VBS) FS ME** algorithm is incorporated. For details of FBS ME algorithm, refer to [that ppr]. VBS FS ME algorithm will be further explained in detail in chapter 4.

Chapter 4

Hardware Architecture for Inter-Prediction

There are many new technologies such as intra prediction, in loop deblocking filter, context based arithmetic coding introduced in the latest H.264/AVC standard. Among all of these amazing technologies, **Variable Block Size Motion Estimation (VBSME)** is one of the powerful techniques. In comparison with the previous Fixed Block Size Motion Estimation (FBSME), VBSME divides one MB into smaller blocks to fit the different motion directions. In this way, the coding performance is proved.

We designed a parallel **265 PE VBS ME Hardware Architecture**. This hardware is implemented in System Verilog. First of all, the pixels in the current MB are stored in a Block Ram (**c_BRAM**). The pixels of the search window are also stored in a block RAM (**s_BRAM**). The architecture is shown in the figure [4.1](#)

In this design, a **2-D systolic PE array** is used. (what is 2d systolic pe array explain). There are **16x16 = 256 PEs** (16 rows and 16 columns) and all of them are made capable of shifting data down, up and left. For a **16x16 MB**, a Motion Vector MV is found in one cycle in a search range of **[-16, 15]** pixels.

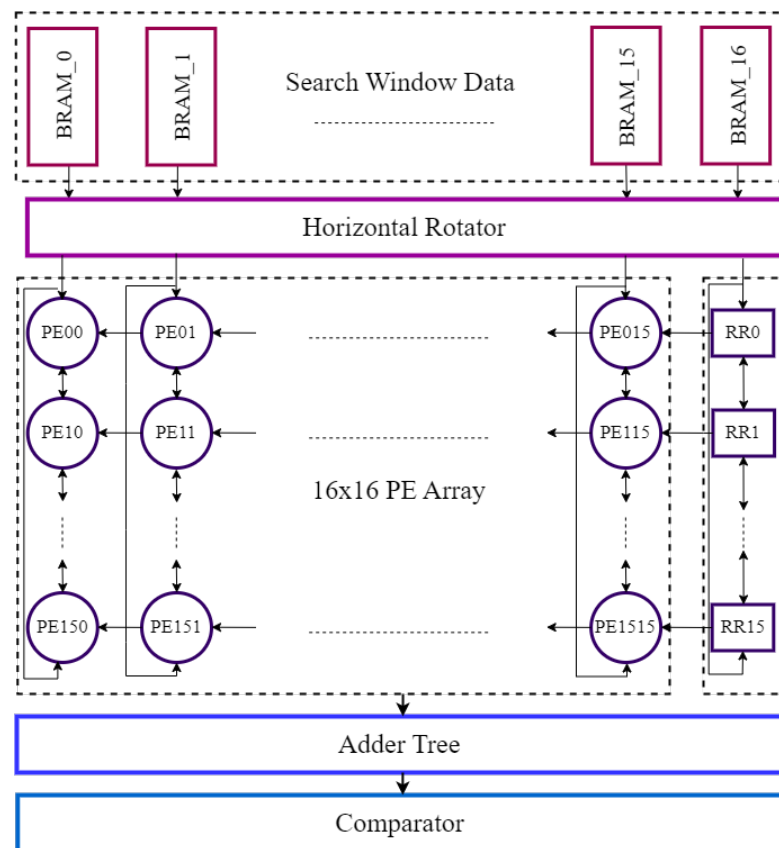


FIGURE 4.1: 265 PE VBS ME Hardware Architecture

Chapter 5

Thesis Structure

The flow of information provided in a thesis depends on its structure. The parameters defining the structure of a thesis are discussed in [2] and are quoted below:

“The structure of a thesis is governed by logic and is invariant with respect to subject. The substance varies with subject, and its quality is determined by the technical knowledge and mastery of essentials exhibited by the student. Style has two components: language and layout. The former deals with the usage of English as a medium of sound technical communication; the latter with the physical presentation of the thesis on paper. All three components structure, substance and style influence one another.”

There is no definite structure for a thesis. The author is the best judge. One possible structure of a thesis can be:

- *Chapter 1*: Introduction
- *Chapter 2*: Motivations, Related Literature and Problem Statement
- *Chapter 3*: Proposed Approach
- *Chapter 4*: Implementation and/or Experimentation
- *Chapter 5*: Conclusions and Future Directions

The purpose of the introduction is to provide an outline of your project in a contextual framework systematically. The introduction should be kept short and to the point.

Depending on the substance, you can break Chapter 2 suggested above into separate chapters. For instance, *motivation and related literature* can be one chapter and *problem statement* can be a separate chapter. This is flexible and is decided while writing the thesis.

The rationale behind the structure selected above is to meet the objective of telling a story as clearly and convincingly as possible. We have adapted the following table from [1] to show the flow in the logic:

Introduction/Aim	What did you do and why?
Materials and Methods	How did you do it?
Observations/Results	What did you find?
Discussion	What do your results mean to you and why?
Conclusions	What new knowledge is extracted from experiment?

There are a number of guiding documents e.g. [5], [4], as well as some documents available online, which can be helpful in writing the thesis. Two such documents outlining some general guidelines are [3], [2].

Finally here is a list of words that you should try avoid while writing the thesis: “very much”, “interesting”, “good”, “fun”, “exciting”, “very”, “too much”.

Appendix A

Introduction to Latex

The material provided in this appendix is taken from
<http://www.sunilpatel.co.uk/thesistemplate.php>

A.1 Learning L^AT_EX

L^AT_EX is not a WYSIWYG (What You See is What You Get) program, unlike word processors such as Microsoft Word or Corel WordPerfect. Instead, a document written for L^AT_EX is actually a simple, plain text file that contains *no formatting*. You tell L^AT_EX how you want the formatting in the finished document by writing in simple commands amongst the text, for example, if I want to use *italic text for emphasis*, I write the ‘\emph{} command and put the text I want in italics in between the curly braces. This means that L^AT_EX is a “mark-up” language, very much like HTML.

A.1.1 A (not so short) Introduction to L^AT_EX

If you are new to L^AT_EX, there is a very good eBook – freely available online as a PDF file – called, “The Not So Short Introduction to L^AT_EX”. The book’s title is typically shortened to just “lshort”. You can download the latest version (as it is occasionally updated) from here:

<http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdf>

It is also available in several other languages. Find yours from the list on this page:

<http://www.ctan.org/tex-archive/info/lshort/>

It is recommended to take a little time out to learn how to use L^AT_EX by creating several, small ‘test’ documents. Making the effort now means you’re not stuck learning the system when what you *really* need to be doing is writing your thesis.

A.1.2 A Short Math Guide for L^AT_EX

If you are writing a technical or mathematical thesis, then you may want to read the document by the AMS (American Mathematical Society) called, “A Short Math Guide for L^AT_EX”. It can be found online here:

<http://www.ams.org/tex/amslatex.html>

under the “Additional Documentation” section towards the bottom of the page.

A.1.3 Common L^AT_EX Math Symbols

There are a multitude of mathematical symbols available for L^AT_EX and it would take a great effort to learn the commands for them all. The most common ones you are likely to use are shown on this page:

<http://www.sunilpatel.co.uk/latexsymbols.html>

You can use this page as a reference or crib sheet, the symbols are rendered as large, high quality images so you can quickly find the L^AT_EX command for the symbol you need.

A.1.4 Figures

There will hopefully be many figures in your thesis (that should be placed in the ‘Figures’ folder). The way to insert figures into your thesis is to use a code template like this:

```
\begin{figure}[htbp]
  \centering
    \includegraphics[width = 1.5in]{./Figures/uet_logo.pdf}
    \rule{35em}{0.5pt}
    \caption{The UET Laore logo.}
    \label{fig:uet_logo}
\end{figure}
```

Also look in the source file. Putting this code into the source file produces the picture of the UET logo that you can see in the figure below.



FIGURE A.1: The UET Lahore logo.

Sometimes figures don’t always appear where you write them in the source. The placement depends on how much space there is on the page for the figure. Sometimes there is not enough room to fit a figure directly where it should go (in relation to the text) and so L^AT_EX puts it at the top of the next page. Positioning figures is the job of L^AT_EX and so you should only worry about making them look good!

Figures usually should have labels just in case you need to refer to them (such as in figure A.1). The ‘`\caption`’ command contains two parts, the first part, inside the square brackets is the title that will appear in the ‘List of Figures’, and so should be short. The second part in the curly brackets should contain the longer and more descriptive caption text.

The ‘`\rule`’ command is optional and simply puts an aesthetic horizontal line below the image. If you do this for one image, do it for all of them.

The L^AT_EX Thesis Template is able to use figures that are either in the PDF or JPEG file format. It is recommended that you read this short guide on how to get the best out of figures in L^AT_EX, available here:

<http://www.sunilpatel.co.uk/texhelp5.html>

Though it is geared more towards users of Mac and OS X systems, much of the advice applies to creating and using figures in general. It also explains why the PDF file format is preferred in figures over JPEG.

A.1.5 Typesetting mathematics

If your thesis is going to contain heavy mathematical content, be sure that L^AT_EX will make it look beautiful, even though it won’t be able to solve the equations for you.

The “Not So Short Introduction to L^AT_EX” (available [here](#)) should tell you everything you need to know for most cases of typesetting mathematics. If you need more information, a much more thorough mathematical guide is available from the AMS called, “A Short Math Guide to L^AT_EX” and can be downloaded from:

<ftp://ftp.ams.org/pub/tex/doc/amsmath/short-math-guide.pdf>

There are many different L^AT_EX symbols to remember, luckily you can find the most common symbols [here](#). You can use the web page as a quick reference or crib sheet and because the symbols are grouped and rendered as high quality images (each with a downloadable PDF), finding the symbol you need is quick and easy.

You can write an equation, which is automatically given an equation number by L^AT_EX like this:

```
\begin{equation}
E = mc^2
\label{eqn:Einstein}
\end{equation}
```

This will produce Einstein’s famous energy-matter equivalence equation:

$$E = mc^2 \tag{A.1}$$

All equations you write (which are not in the middle of paragraph text) are automatically given equation numbers by L^AT_EX. If you don't want a particular equation numbered, just put the command, '`\nonumber`' immediately after the equation.

A.2 Sectioning and Subsectioning

You should break your thesis up into nice, bite-sized sections and subsections. L^AT_EX automatically builds a table of Contents by looking at all the '`\chapter{}`', '`\section{}`' and '`\subsection{}`' commands you write in the source.

The table of Contents should only list the sections to three (3) levels. A '`\chapter{}`' is level one (1). A '`\section{}`' is level two (2) and so a '`\subsection{}`' is level three (3). In your thesis it is likely that you will even use a '`\subsubsection{}`', which is level four (4). Adding all these will create an unnecessarily cluttered table of Contents and so you should use the '`\subsubsection*{}`' command instead (note the asterisk). The asterisk (*) tells L^AT_EX to omit listing the subsubsection in the Contents, keeping it clean and tidy.

References

- [1] R. Barrass. *Scientists must write: a guide to better writing for scientists, engineers and students*. RoutledgeFalmer, 2002.
- [2] R. Chandrasekhar. How to Write a Thesis: *A Working Guide*. <http://ciips.ee.uwa.edu.au/pub/HowToWriteAThesis.pdf>, Last accessed on May 19, 2010.
- [3] S. Easterbrook. How Thesis Get Written. <http://www.cs.toronto.edu/~sme/presentations/thesiswriting.pdf>, Last accessed on May 19, 2010.
- [4] D. Evans and P. Gruba. *How to write a better thesis*. Melbourne University Publishing, 2003.
- [5] R. Murray. *How to write a thesis*. Open Univ Press, 2006.