UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

DANIEL MUNARI PALOMINO

# Algorithm and Hardware Based Architectural Design Targeting the Intra-Frame Prediction of the HEVC Video Coding Standard

Thesis presented in partial fulfillment of the requirements for the degree of Master of Computer Science

Prof. Dr. Altamiro Amadeu Susin
Advisor

Prof. Dr. Luciano Volcan Agostini
Co-Advisor

Porto Alegre, Fevereiro de 2013.

# CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Palomino, Daniel

Algorithm and Hardware Based Architectural Design Targeting the Intra-Frame Prediction of the HEVC Video Coding Standard / Daniel Palomino. – Porto Alegre: PPGC da UFRGS 2012.

94 f.: il.

Orientador: Altamiro Susin; Co-orientador: Luciano Agostini.

Thesis (Master) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação. Porto Alegre, BR – RS, 2013.

1. Arquitetura de Hardware. 2.Codificação de Vídeo. 3. Predição Intra-Quadro. 4. HEVC. 5. Modo de Decisão. 6. Algoritmos Rápidos. I. Susin, Altamiro. II. V. Agostini, Luciano. III. Algorithm and Hardware Based Architectural Design Targeting the Intra-Frame Prediction of the HEVC Video Coding Standard.

# AGRADECIMENTOS

Agradeço primeiramente ao povo brasileiro que financiou meus estudos durante o meu curso de mestrado. Agradeço também, aos colegas de laboratório que estiveram presentes durante o período do meu mestrado, em especial aos colegas Cauane, Cláudio, Débora, Duda, Felipe, Kléber, Léo e Vizzotto por compartilharem várias conversas, risadas, churrascos e viagens a congressos.

Agradeço também aos meus colegas de apartamento Felipe e Diego que além de serem grandes amigos, foram companheiros muito agradáveis no convívio dos meus dois primeiros anos morando fora da casa dos meus pais. Vivenciamos e resolvemos vários problemas juntos nesse tempo, mas também compartilhamos muitos churrascos, jogos, discussões filosóficas e risadas.

Um agradecimento especial aos meus professores orientador e co-orientador, por terem me guiado durante o período do mestrado. Graças aos professores e amigos Altamiro Susin e Luciano Agostini eu fui capaz de concluir a minha dissertação.

Por fim, um agradecimento especial a minha namorada e aos meus pais, por terem me aturado nos momentos mais difíceis durante o período do meu curso de mestrado. A compreensão deles foi muito importante para que eu nunca perdesse o foco.

# CONTENTS

# LIST OF ABBERVIATIONS AND ACRONYMS

| | |
|---|---|
| ASIC | Application-Specific Integrated Circuit |
| AVC | Advanced Video Coding |
| CDR | Complete Decision Reuse |
| CU | Coding Unit |
| DCT | Discrete Cosine Transform |
| FDR | First Decision Reuse |
| FPGA | Field-Programmable Gate Array |
| GoP | Group of Pictures |
| GPB | Generalized P and B Picture |
| HD | High Definition |
| HE | High Efficiency |
| HEVC | High Efficiency Video Coding |
| HM | HEVC test Model |
| HP | Hewlett-Packard |
| IBM | International Business Machines |
| ISO/IEC | International Organization for Standardization/International Electrotechnical Commission |
| ITU-T | International Telecommunication Union |
| JCT-VC | Joint Collaborative Team on Video Coding |
| JVT | Joint Video Team |
| KB | Kilo Byte |
| LC | Low Complexity |
| LCU | Large Coding Unit |
| MDR | Majority Decision Reuse |
| MHz | Megahertz |
| MPEG | Moving Picture Experts Group |
| MSE | Mean Square Error |

| | |
|---|---|
| PDAs | Personal Digital Assistant |
| PSNR | Peak Signal-to-Noise Ratio |
| PU | Prediction Unit |
| QP | Quantization Parameter |
| RDO | Rate-Distortion Optimization |
| RGB | Red Green Blue |
| SAD | Sum of Absolute Differences |
| TMuC | Test Model under Consideration |
| TSMC | Taiwan Semiconductor Manufacturing Company |
| TU | Transform Unit |
| TV | Television |
| VCEG | Video Coding Experts Group |
| YCbCr | Luminance (Y), Chrominance blue (Cb), Chrominance red (Cr) |

# LIST OF FIGURES

# LIST OF TABLES

# RESUMO

Este trabalho apresenta uma arquitetura de hardware para a predição intra-quadro do padrão emergente HEVC de codificação de vídeo. O padrão HEVC está sendo desenvolvido tendo como principal objetivo o aumento em 50% na eficiência de compressão, quando comparado com o padrão H.264/AVC, atual padrão estado da arte na codificação de vídeos. Para atingir este objetivo, várias novas ferramentas de codificação foram desenvolvidas para serem introduzidas no novo padrão HEVC. Embora essas novas ferramentas tenham obtido êxito em aumentar a eficiência de compressão do novo padrão HEVC, elas também colaboraram para o aumento da complexidade computacional no processo de codificação. Analisando somente os avanços na predição intra-quadro, em comparação com o padrão H.264/AVC, é possível perceber que vários novos modos direcionais de codificação foram inseridos no processo de predição. Além disso, existem mais tamanhos de blocos que podem ser considerados pela predição intra-quadro. Nesse contexto, este trabalho propõe o uso de duas abordagens para melhorar o desempenho da predição intra-quadro em codificadores HEVC. Primeiramente, foram desenvolvidos algoritmos rápidos de decisão de modo, baseados em heurísticas, para a predição intra-quadro. Os resultados mostraram que é possível reduzir a complexidade computacional do processo de predição intra-quadro com pequenas perdas na eficiência de compressão (taxa de bits e qualidade visual). No pior caso, a perda foi de 6.9% na taxa de bits e de 0.12dB na qualidade, para uma redução de 35% no tempo de processamento. Em seguida, utilizando um dos algoritmos desenvolvidos, uma arquitetura de hardware para a predição intra-quadro foi desenvolvida. Além da redução de complexidade proporcionada pelo uso do algoritmo desenvolvido, técnicas de desenvolvido de hardware, tais como aumento no nível de paralelismo e uso de pipeline, também foram utilizadas para melhorar o desempenho da arquitetura desenvolvida. Os resultados de síntese da arquitetura para a tecnologia IBM 0,65um mostram que ela é capaz de operar a 500MHz, atingindo uma taxa de processamento suficiente para realizar a predição intra-quadro de mais de 30 quadros por segundo para resoluções como Full HD (1920x1080pixels).

**Palavras-Chave:** Arquitetura de Hardware, Codificação de Vídeo, Predição Intra-Quadro, HEVC, Modo de Decisão, Algoritmos Rápidos.

**Algorithm and hardware based architectural design targeting the Intra-Frame Prediction of the HEVC Video Coding Standard**

# ABSTRACT

This work presents an intra-frame prediction hardware architecture targeting the emerging HEVC video coding standard. The HEVC standard is being developed with the main goal of increase the compression efficiency in 50% when compared to the latest H.264/AVC video coding standard. To achieve such a goal, several new video coding strategies were developed to be used in the HEVC. Although these strategies have increased the compression efficiency of the emerging HEVC standard, it also increased the computational complexity of the encoding process. Looking only to the intra prediction process, several new directional modes are used to perform the prediction. Besides, there are more block sizes that can be supported by the intra prediction process. This work proposes to use two different approaches to improve the HEVC intra prediction performance. First we developed fast intra mode decision algorithms, showing that it is possible to decrease the intra prediction computational complexity with negligible loss in the compression performance (bit-rate and video quality). In the worst case, the bit-rate loss was 6.99% and the PSNR loss was 0.12dB in average allowing reducing the encoding time up to 35%. Then, using the developed fast algorithms as base, this work proposes an intra prediction hardware architecture. The designed architecture was specifically based on one of the developed fast intra mode decision algorithms. Besides, hardware techniques such as increase the parallelism level and pipeline were also used to improve the intra prediction performance. The synthesis results for the IBM 0.65nm have shown that the architecture is able to achieve 500MHz as maximum operation frequency. This way, the architecture throughput is enough to perform the intra prediction process for more than 30 frames per second considering high resolution digital videos, such as Full HD (1920x1080).

**Keywords**: Hardware Design, Video Coding, Intra-Frame Prediction, HEVC, Mode Decision, Fast Algorithms.

# 1 INTRODUCTION

The latest advances in silicon based technology have provided a huge increase in the processing capabilities of modern portable consumer devices. Hard processing applications, such as multimedia, are now widely used, since these advances allowed a large expansion of this market. Cell phones, digital TVs, portable computers, and PDAs are among the most popular consumer devices capable of receiving and displaying high resolution digital videos in real time. A massive number of those devices can also capture and transmit digital videos through wired or wireless networks. Furthermore, the capability of encoding and decoding high resolution digital videos are included in most portable devices with embedded digital cameras.

Due to the high amount of data necessary to represent digital videos, the video compression technique is crucial for the success of applications and devices that deal with this kind of media. As digital video booms as key application, there is an increasing demand for visual quality of these videos. Hence, multimedia applications need to support images with higher resolution and lower distortion to improve visual quality. In this context both academia and industry have focused their efforts on research new algorithms for video coding, with the main goal of enabling efficient storage and transmission of digital videos at high quality and high resolution.

Several video coding standards have been developed along recent years with the main goal of increasing the compression efficiency. Experts groups such as VCEG (Video Coding Experts Group) from ITU-T (International Telecommunication Union) and MPEG (Moving Picture Experts Group) from ISO/IEC (International Organization for Standardization) have been developing these new standards, introducing and developing new coding techniques to enable high compression rates with little to none impacts in the visual quality.

The H.264/AVC (JVT, 2003) is still the current state-of-the-art video coding standard and it was developed by the JVT (Joint Video Team) in 2003. The main goal of the H.264/AVC was to improve the compression performance when compared with previous standards. In order to achieve such a goal, several tools and new algorithms were introduced in the standard. This way, the compression efficiency gain achieved by the H.264/AVC standard was 50% when compared to the previous MPEG-2 standard (RICHARDSON, 2003).

The H.264/AVC was very successful, since the compression efficiency gain achieved by this standard was impressive when compared to previous standards. Then it was defined as the video coding standard for blu-rays and also for the Brazilian System of Digital Television. Even so, the research in video coding algorithms did not stop. The team of experts pursued their research aiming a new coding standard that would be able to surpass the H.264/AVC compression efficiency, in order to address the future generation of digital videos (higher resolutions and higher visual quality).

Thus, a new team of video experts was formed in January of 2010 named JCT-VC (Joint Collaborative Team on Video Coding) and a new call for proposals was launched to gather industry and academia contributions on video coding for the emerging standard. The main goal was again to achieve 50% of compression gain when compared to the latest H.264/AVC standard at the same quality. The proposals were submitted and initially tested in the TMuC (Test Model under Consideration) (ISO/IEC-JTC1/SC29/WG11, 2011) software. This software had two versions before being officially named as HM (High Efficiency Video Coding test model). The HEVC test model (HM) is currently in the version 8.2. The first HEVC standard draft is expected to be launched in the first semester of 2013.

The coding efficiency gain of these latest coding standards is provided by high computational complexity algorithms that are introduced in this process, especially in the encoding step. Previous studies have shown that the H.264/AVC decoding complexity is four times higher than MPEG-2, while in the encoding part, this ratio can be up to eight times higher (SUNNA, 2005).

Although HEVC already achieved the target compression improvement, its computational complexity has increased as well. The HEVC standard includes several new structures, algorithms and coding options when compared to the H.264/AVC. Coding unit (CU) forming quad-trees are the new data structures and the amount of possibilities to encode one block has increased a lot. This way, the HEVC encoder requires a much more complex decision engine than the H.264/AVC, which was already a critical task.

Figure 1.1 shows a graph where the HEVC and H.264/AVC encoding computational complexity (in terms of time) are compared with different resolution videos (from 240p up to 1600p). This evaluation was carried out according to the HM-like configuration, using the H.264/AVC JM reference software 18.3 (JM, 2012). Even considering that both reference software are implemented with a non optimized code in terms of processing time, it is possible to get an idea about the computational complexity[1] relation between the two standards. From figure 1.1 it is possible to see how much the HEVC computational complexity is bigger than the H.264/AVC.

---

[1] In this text computational complexity is always referring to encoding time.

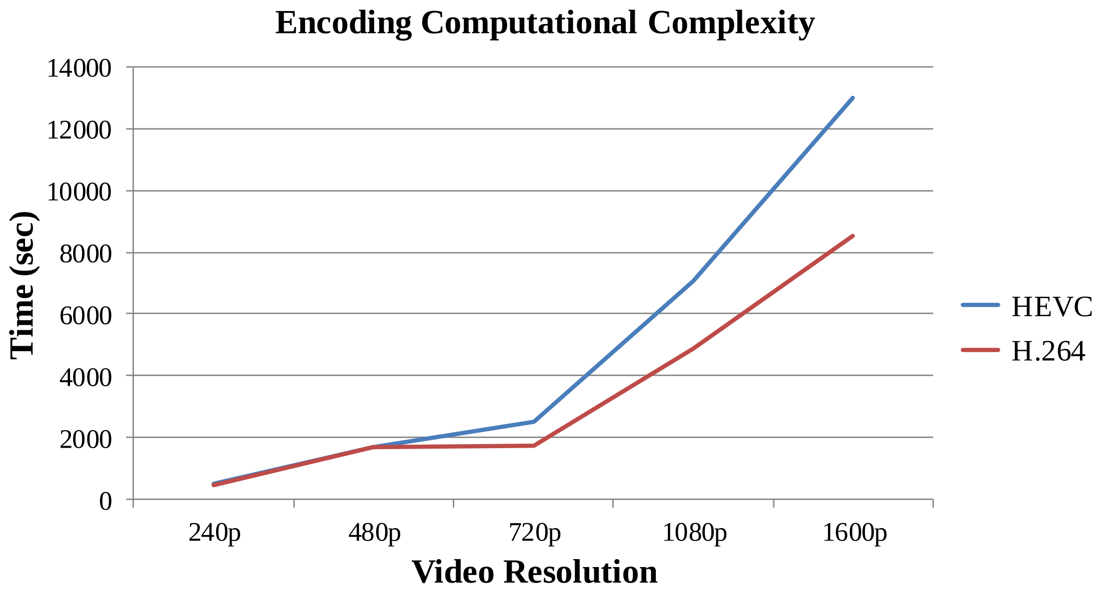


Figure 1.1: Computational complexity comparison between HEVC and H.264/AVC encoders.

Looking only into the intra-frame prediction of the current version of HEVC, which is the focus of this work, the Coding Unit (CU) can be coded according to four different sizes (8x8, 16x16, 32x32 and 64x64 pixels). Moreover, each PU supports up to 34 directions in the intra prediction process to select the best intra mode. The high complexity of this process poses a huge challenge if all directions and all CUs sizes are employed in the rate-distortions optimization (RDO) process (SULLIVAN e WIEGAND, 1998). Thus, it is necessary the development of fast intra prediction engines to allow the use of HEVC encoders in applications that demand for real time processing and low energy consumption.

There are two classes of solutions in the literature that deal with this complexity issue in the intra prediction process of current video coding standards. The first one is to design dedicated hardware architectures (FPGA or ASIC) for the intra prediction process using hardware techniques (parallelism, pipeline and etc.) to improve its performance in terms of execution time, such as (DINIZ, ZATT, *et al.*, 2009). The work (DINIZ, ZATT, *et al.*, 2009) proposes an intra prediction hardware architecture for the H.264/AVC standard targeting high throughput to process high resolution videos, such as HD1080p. The targeting processing rate is achieved by exploring the parallelism of intra prediction and by reducing the latency.

The second class of works try to decrease the number of intra prediction options following on-line or off-line pre-evaluations, based on statics or using neighbor information. Previous works such as (BHARANITHARAN, LIU, *et al.*, 2008) (HUANG, HSIEH, *et al.*, 2005) (HUANG, OU e CHEN, 2010) (PARLAK, ADIBELLI e HAMZAOGLU, 2008) (WANG, WANG, *et al.*, 2007) propose fast algorithms using different strategies for the intra prediction process in the H.264/AVC standard. However, all these works can not be directly used in the new HEVC standard, since there is new coding structures, more blocks sizes and more intra mode directions to be considered in the intra prediction decision process.

In this context, the main goal of this master thesis is to design an intra prediction mode decision engine to be integrated with an intra prediction hardware architecture for the emerging HEVC standard, reducing the computational complexity of this process and reaching a high processing rate. In this work we propose gathering both methods – dedicated hardware architecture and fast intra decision algorithms – in only one solution for the intra prediction of the emerging HEVC standard.

We first developed three fast intra mode decision algorithms based on a new evaluation order in the HEVC coding structure to improve the intra prediction decision performance. Then, we designed a complete intra prediction hardware architecture (PALOMINO, SAMPAIO, *et al.*, 2012) to enable the use of one developed fast mode decision algorithm to increase the architecture performance.

The rest of this master thesis is structured as follows: Chapter 2 introduces some basic video coding concepts that will be useful to the best understanding of our contribution. Chapter 3 explains more detailed the HEVC intra prediction module and how the mode decision is performed. Chapter 4 presents the fast intra mode decision algorithms developed in this work, besides the results achieved by these algorithms. Chapter 5 shows the hardware architecture designed for the HEVC intra prediction module and how the fast intra mode decision algorithm contributed to improve the architecture performance.

# 2 VIDEO CODING

In this chapter the basic concepts of digital videos and video coding are presented. It addresses the digital video characteristics, i.e., how digital videos are represented by frames and color spaces. This chapter also presents the types of redundancy inherent to digital videos and how the hybrid video coding framework takes advantage of these redundancies in order to compress the video content. Then, some coding efficiency metrics are presented to show how the compression efficiency is objectively measured and compared.

## 2.1 Digital Video Characteristics

Usually, a digital video is composed by a sequence of equal size pictures (frames) and the picture size, also called resolution, is defined by two dimensions (column and row) of pixels. The video resolution is as high as is the number of pixels to represent the frame, as higher is the resolution dimension better is the perception of the video details. The motion perception occurs when these frames are sampled considering a defined frequency (frames/second). Usually, the frame frequency necessary for motion perception for the visual human system is, at least, of 24 frames per second (RICHARDSON, 2003).

Digital videos are represented by color spaces that separate pixels in color components. This way, the video encoders can better lead with each color component separately achieving higher compression rates. Several color spaces such as RGB (Red Green Blue) and YCbCr (Y-luminance, Cb-chrominance blue, Cr-chrominance red) (RICHARDSON, 2003) had been used to depict digital pictures. The well-known RGB color space divides each pixel in three color components: R (red), G (green) and B (blue). The YCbCr is also composed by three components: luminance (Y), chrominance blue (Cb) and chrominance red (Cr), which uses a luminance component with a blue and red deviation to define the color when displaying the pixel (RICHARDSON, 2003).

The YCbCr was developed to independently lead with the luminous information, since the perception of the visual human eye is more sensitive for luminous intensity changes than color intensity changes (RICHARDSON, 2003). Thus, both types of information (luminance and chrominance) can be separately manipulated for the video encoders. For example, in the H.264/AVC standard, the YCbCr color space is mandatory and the majority of H.264/AVC profiles use less chrominance samples than luminance samples to represent the frame (RICHARDSON, 2003). This technique is named as color sub-sampling and it is widely used in video coding standards, since it performs a compression directly in the video representation by discarding a great amount of data (related to the chrominance samples) with none or minimal impact to the human eye perception.

## 2.2 Digital Video Redundancies

The video compression techniques take advantage of the inherent data redundancy that is present in digital videos. The main goal of the compression algorithms is to reduce as much as possible the data redundancy in the video representation, which implies in less demand of bandwidth (to transmit) and storage resources (to store) when applications are leading with videos. There are basically three different types of redundancy exploited by the compression algorithms in regular video coding standards: spatial redundancy, temporal redundancy and entropic redundancy (RICHARDSON, 2003).

The spatial redundancy, also called intra-frame redundancy (WIEGAND, 2003), comes from the inherent correlation among pixels in a frame. This correlation is easily seen in neighbor pixels of a frame, which tend to be similar since they may represent the same object in a given scene. In this case, the redundancy can be reduced through the intra-frame coding process that is present in current video coding standards, by intelligently exploiting this neighbor pixels correlation. Another tool used in current video encoders to reduce this type of redundancy is the transform and quantization process. But, in this case, the spatial redundancy is explored considering the prediction residues which result from a first encoding step applied over input frames to explore the spatial (intra prediction) or temporal (inter prediction) redundancies.

The temporal redundancy, also called inter-frames redundancy (WIEGAND, 2003), is related to the high correlation among neighbor temporal frames. As a video sequence has at least 24 frames per second, lots of regions do not significantly change from one frame to another in the video sequence. Besides, it is possible that a frame region may be displaced from one frame to another, for example, an object moving in a scene. This way, the encoder can just send an indication of where the region of one frame is located in another one, instead of sending all information again. The difference between the original and the predicted region, if not equal to zero, must be sent to the other encoder steps. The efficient reduction in the temporal redundancy brings huge compression rates.

The entropic redundancy is related with the amount of bits used to represent each encoded sample. The techniques and algorithms used to reduce this type of redundancy try to transmit as much information as possible per generated symbol.

## 2.3 Hybrid Video Coding Framework

The main goal of the video coding process is to represent the great amount of information in digital videos with as less data as possible. However, there are issues related to the visual quality of the encoded video that must be considered in the video coding process. This way, the encoders must consider a trade-off between the encoded data size and the video visual quality after the video decoding to achieve a good coding efficiency. It means that a video encoder can not only target in reducing the amount of data in the final encoded video, it also needs to care about the visual quality.

As already mentioned, a digital video is composed by frames. However, the hybrid video coding process is not performed over an entire frame, since it would not be possible to exploit the spatial redundancy in a frame. To avoid that, all video coding process is based on a smaller representation, usually blocks of sizes varying from 4x4 up to 64x64 pixels.

The current (or original) block definition means the block that is currently being coded whereas the reference block (or reference area) is the one whose will be used as redundancy reference and it was already encoded and decoded. The hybrid video coding framework is divided in well-established steps and the coding process is performed block by block in a raster way. Figure 2.1 shows a high level block depicting the hybrid video coding framework.

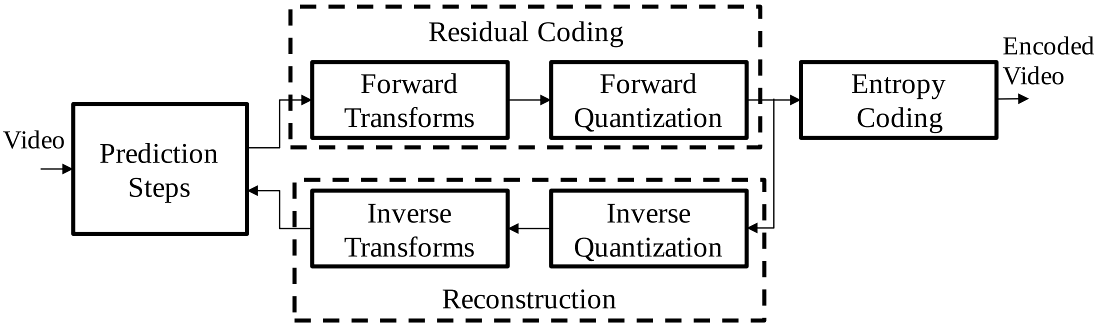


Figure 2.1: Hybrid video coding framework.

The first step covers the prediction steps, where the temporal and spatial redundancies can be exploited by algorithms that try to generate the best representation of the current block using some redundancy references. The redundancy references are generated from the temporal and spatial neighbor blocks already encoded. After the block generation, called predicted block, the difference between this block and the original one is calculated generating the residual block. This way, only the mode how this prediction was performed and the residual information are transmitted to the next coding steps.

The second hybrid video coding frame work step is the residual coding, which also exploits the spatial redundancy in digital videos. One or more two dimensional transforms, typically the 2-D DCT (RICHARDSON, 2003), are applied to the residual data to generate the coefficients that will be finally processed by the quantization step. Although the quantization is used to improve the compression rates, this is a lossy process, i.e., in the decoding process the reconstructed information will not be the same as the original one. The quantization strength used in the coding process is crucial to define the final video visual quality and also the reached compression rate.

The third and last step is the entropy coding, where the data generated by the quantization process is exploited targeting the redundancy in the symbols that will represent the encoded video. This step is lossless and it uses well know variable length and arithmetic encoding techniques over the quantized residual information.

There is a fourth step, which is actually a decoding step inside the encoder (called reconstruction in figure 2.1). This step is very important and it must be performed to avoid mismatch between the coding and decoding processes. As previously mentioned, the quantization step leads the information to some degradation which can not be recovered in the decoding process. In the encoding process, the prediction steps use reference information to generate the predicted block. This reference information must be exactly the same in the decoding process, since they will be used to reassembly the video based on residual and mode information. This way, it is necessary to perform the decoding process over the blocks within the encoder to use the reconstructed information as reference to the next coding blocks, keeping the references equal in both encoder and decoder processes.

## 2.4 Coding Efficiency Metrics

It is important to define some metrics to evaluate the coding efficiency over a digital video. There are two main metrics to be considered. The first one is the bit-rate generated after the encoding process. This is measured by counting the amount of bits after the encoding process. Usually, as lesser is the amount of bits as better is the encoding efficiency. However, the visual quality is also an important issue, and sometimes it is better to sacrifice a little the bit-rate in order to have a good image quality.

The second important metric is the video visual quality after the encoding. The visual quality is hard to be measured and evaluated, since it is very subjective. The subjective metrics are not easy to obtain, since it needs a group of people evaluating and scoring the image quality of a video benchmark. This way, this kind of evaluation will not be considered in this work. The objective metrics are the most used in current video coding standards. The criteria of these metrics are based on comparing the decoded frame (the decoding process inside the encoder) with the original frame, pixel to pixel.

The most known and used in the literature objective distortion metric is the PSNR (Peak Signal-to-Noise Ratio) (GHANBARI, 2003), defined in equation (2.1).

$$PSNR_{dB} = 20.log_{10}(\frac{MAX}{\sqrt{MSE}}) \qquad (2.1)$$

In (2.1), MAX is the maximum value that a luminance sample can reach ($2^n$-1, where *n* is the number of bits to represent a sample) and MSE is mean-squared error, defined in (2.2), where *m* and *n* are the image dimensions in pixels and *O* and *R* are the luminance components in the original and reconstructed frame, respectively.

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left(R_{i,j} - O_{i,j}\right)^2 \qquad (2.2)$$

The MSE is a distortion metric to measure the difference among original and reconstructed frames. As MSE, there are other metrics with the same goal, but using functions less or more accurate and less or more computational complex than MSE. One of this metrics is the SAD (Sum of Absolute Differences) which is shown in equation (2.3).

$$SAD = \sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left|R_{i,j} - O_{i,j}\right| \qquad (2.3)$$

In (2.3), *O, R, m* and *n* means the same as in (2.2). The SAD function is a low complexity objective visual quality metric widely used in the steps of current video coding standards.

# 3 INTRA PREDICTION AND MODE DECISION IN HEVC

In the previous chapter digital video characteristics and video coding concepts were introduced. This chapter presents the coding structure of the HEVC emerging standard, focusing on the intra prediction module (which is the main target of this work). The intra prediction module is detailed explained and the computational complexity involved in the process of how the modes are chosen is explained as well.

## 3.1 Coding Structure in the Emerging HEVC Standard

The emerging HEVC standard keeps the block based hybrid video coding framework. However it provides a highly flexible hierarchy of unit representation. One frame is divided in a sequence of square dimensioned units called tree blocks. Each tree block is formed by *NxN* luminance samples with the two chrominance blocks. The chrominance block dimensions vary according to the used color subsampling. In the current version of HEVC (JCT-VC, 2012), the maximum size allowed for a tree block is 64x64 pixels.

Each tree block is composed by one or more basic coding units (CU), which are macroblock like regions whether compared to the H.264/AVC standard. The main difference is that CUs can have different sizes according to the region that is being coded whereas a macroblock has a fixed size, always 16x16 pixels. The CU can be recursively divided into four equally sized blocks from the tree block. This recursively division process build a coding structure in a quad tree fashion composed by CUs, which can be of 8x8 size up to the tree block size (64x64). Figure 3.1 shows one possible quad tree division, where the tree block size is of 64x64 pixels.

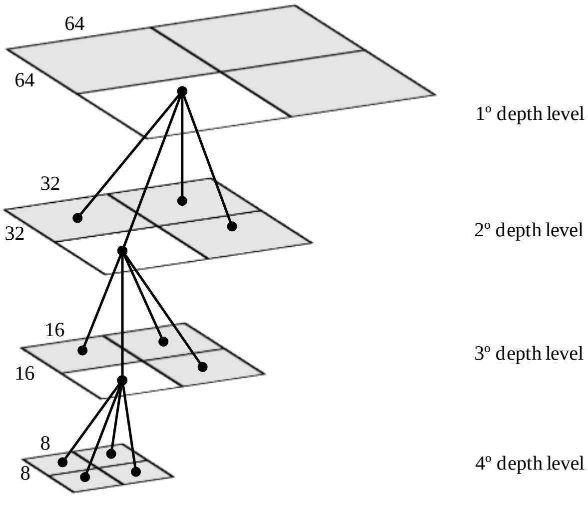


Figure 3.1: Example of one 64x64 tree block division.

In figure 3.1 the tree block is divided until the last possible depth level. In the first division four 32x32 CUs are generated. Three blocks are coded as 32x32 CUs (2º depth level) and one of them is again divided into other four 16x16 CUs. Once again, three blocks are coded as 16x16 CUs (3º depth level), and the other one is divided in four 8x8 CUs. As this is the smallest allowed CU dimension by the HEVC, these blocks can not be divided in smaller CUs. Therefore they are coded as 8x8 CUs in the 4º depth level. With this high flexibility in the coding unit size it is possible to use smaller CUs to encode highly detailed areas and bigger CUs to those more homogenous areas, increasing the compression performance.

After the quad tree generation, each CU can be partitioned in basic prediction units (PU). The PUs can have either square or rectangular shapes. The CU partition is performed according figure 3.2. The PU is the structure where the prediction steps are applied to. The **N** value in figure 3.2 is equal to half (½) of the block size.

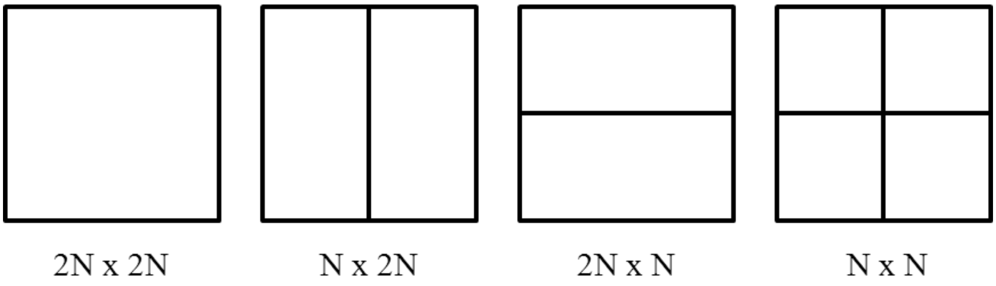
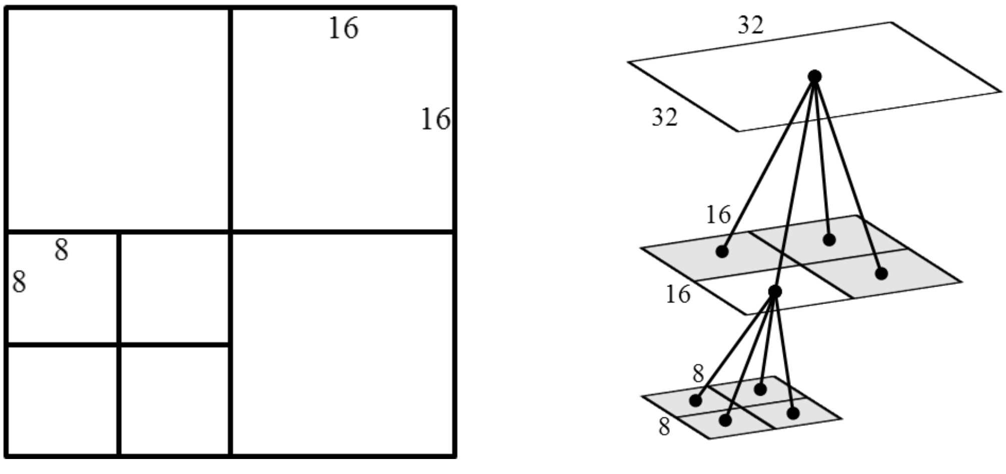


Figure 3.2: Possible CU partitions in Prediction Units (PU).

The basic units for transform and quantization process are called Transform Units (TU). The block shape is always squared and the block size can vary from 4x4 up to 32x32 luminance samples. Each CU can have one or more TUs. The TUs are organized in a quad tree structure as well as the CUs. Figure 3.3 shows one 32x32 CU divided into TUs of 16x16, 8x8 and 4x4 dimensions.



(a) (b)

Figure 3.3: (a) Example of a 32x32 CU divided into TUs; (b) TU quad tree related to the CU division.

Based on the recursive structure used in CU, PU and TU division process, the encoder must find the best combination of CU, PU and TU sizes in a tree block to achieve a good coding efficiency.

## 3.2 Encoder Configuration

The JCT-VC defined eight different encoding scenarios to be used in the tests for the new algorithms proposed to be inserted in the HEVC. These encoding scenarios are called Common Test Conditions (ISO/IEC-JCT1/SC29/WG11, 2011)). All scenarios are combinations of two types of more specific configurations. The first one is related to the number of tools used in the encoding process, whereas the other one is related to the encoding video structure considering the types of used slices (ISO/IEC-JCT1/SC29/WG11, 2011).

### 3.2.1 Encoder Structure Configuration

The set of tools used by the encoder structure configuration can be of two types: High Efficiency (HE) configuration or Low Complexity (LC) configuration. Table 3.1 shows which tools are used in each type of configuration. In the High Efficiency configuration, the encoder uses tools that increase the compression performance at the most. On the other hand, the Low Complexity configuration aims an acceptable compression performance decreasing the involved computational complexity.

Table 3.1: Encoder tool configurations (HM 5.1).

<table>
<tr><th>High Efficiency</th><th>Low Complexity</th></tr>
<tr><td colspan="2">Quad-tree structure of Coding Units (CUs):<br>8x8 up to 64x64 block sizes of luminance samples</td></tr>
<tr><td colspan="2">Prediction Units</td></tr>
<tr><td colspan="2">Quad-tree structure of Transform Units (TUs): maximum of 3 levels</td></tr>
<tr><td colspan="2">Transform block sizes of 4x4 up to 32x32 samples,<br>Mode dependent transform for 4x4 blocks</td></tr>
<tr><td colspan="2">Spatial intra-frame prediction: 34 directional angles and planar,<br>Intra-frame adaptive smoothness<br>Intra-frame prediction for chrominance using luminance samples</td></tr>
<tr><td colspan="2">DCT based filter for luminance (1/4 of pixel, 8 taps)</td></tr>
<tr><td colspan="2">DCT based interpolation filter for chrominance (1/8 of pixel, 4 taps)</td></tr>
<tr><td colspan="2">SKIP Coding Unit and Prediction Unit Merging</td></tr>
<tr><td colspan="2">Advanced motion vector prediction</td></tr>
<tr><td>Context-adaptive binary arithmetic coding</td><td>Context-adaptive VLC</td></tr>
<tr><td>Internal bit-depth raised in two bits</td><td>-</td></tr>
<tr><td colspan="2">Blocking effect reduction Filter</td></tr>
</table>

### 3.2.2 Encoder Temporal Configuration

The second type of configuration defines the temporal prediction structure used in the encoding process. There are three possible configurations: Intra Only, Low Delay and Random Access.

Intra Only: this configuration allows only I frames in the temporal structure, i.e., there is no temporal prediction and none previous frames are used as reference. Besides, the quantization parameter (QP) can not be changed during the encoding process using this configuration. Figure 3.4 shows one example of frames being encoded using the Intra Only configuration, where *Fn* (frame *n*) means one frame coded using only the intra-frame prediction.

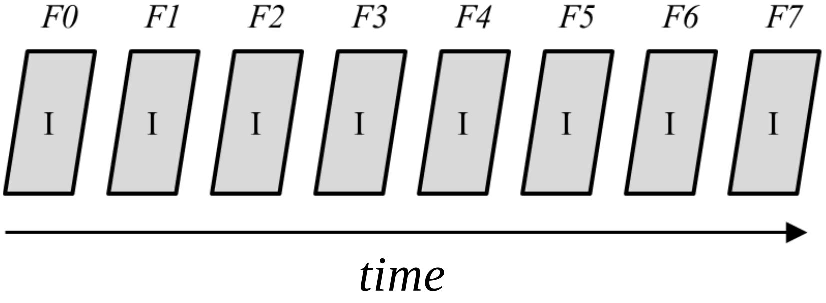


Figure 3.4: Example of coding using Intra Only configuration.

Low Delay: this configuration uses one structure where only the first frame is mandatorily an I frame, whereas the following frames are GPB (Generalized P and B Picture), one new type of frame where up to two frames can be used as reference for the inter-prediction, if both frames have already been displayed. Figure 3.5 shows a graphic representation of the Low Delay configuration, where each *Fn* is one coded frame and *n* is the number of frames in order. The *F1* is the unique I-frame, since it can not use any previous displayed frame as reference, whereas the *F2* to *F9* are GPB frames, since they used until two previous frames as reference. In this configuration the coding order must follow the display order.

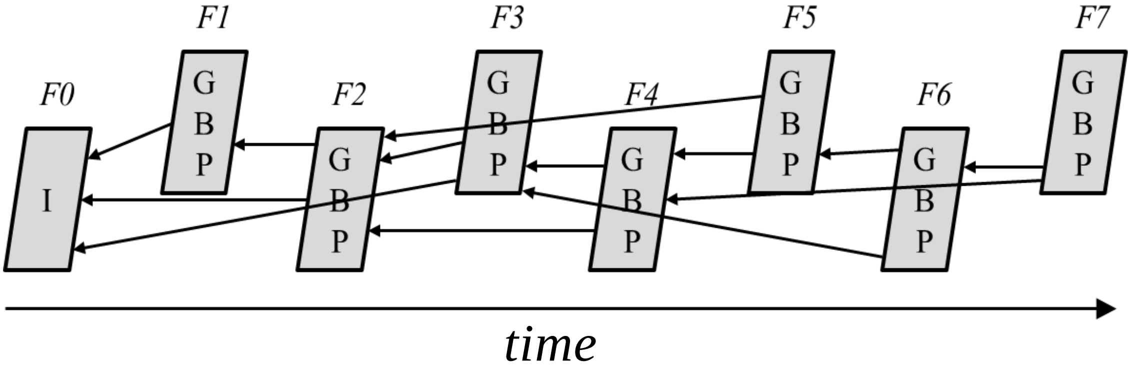


Figure 3.5: Example of coding using the Low Delay configuration.

Random Access: in this configuration, one hierarchical structure of B frames is used in the encoding process. This hierarchical structure, already presented in the H.264/AVC standard, defines limitations for the frames used as reference, as shows in figure 3.6.

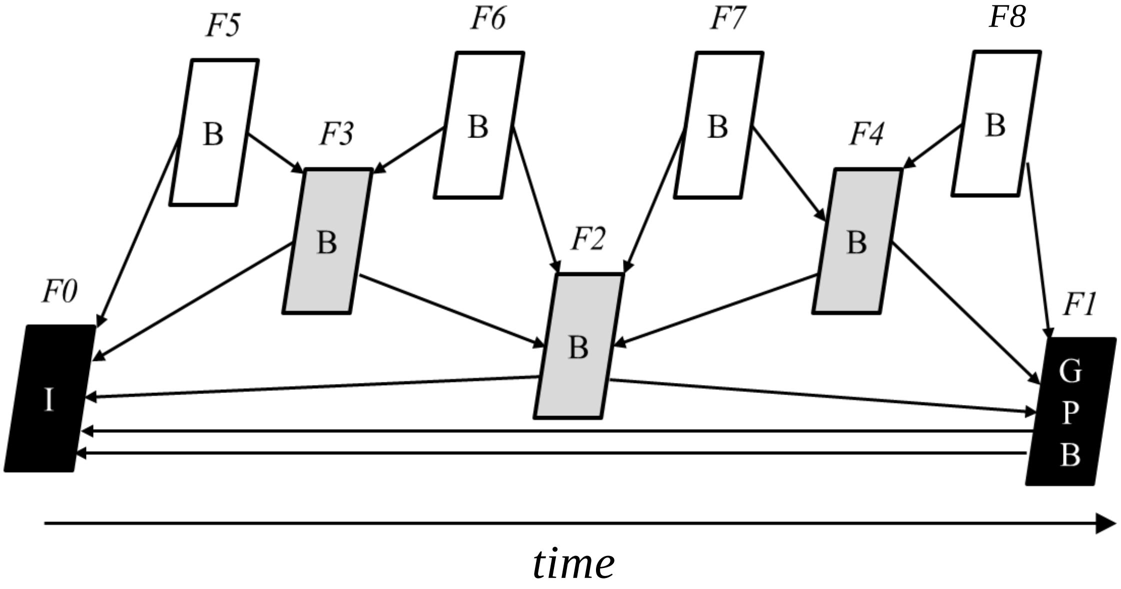


Figure 3.6: Example of coding using the Random Access configuration.

In the Random Access configuration, the first frame of the video sequence is coded using an I frame and the following frames are B. At each determined time, one new I frame must be used (*F0* in figure 3.6), establishing a new Group of Pictures (GoP). The B frame in the lowest hierarchical coding level (*F1* in figure 3.6) can not use the other B frames as reference, unless those previous displayed, thus, this frame is called GPB. The following B frames in the immediately above hierarchical coding levels are regular coded using inter bi-prediction and they can use any frames in the below levels. The B frames in the highest hierarchical level (*F5, F6, F7* and *F8* in figure 3.6) can use any other frame in the levels below. They are not used as reference in the group of pictures structure, since there are no other frames above in the hierarchical coding structure.

## 3.3 Intra Prediction in HEVC

The HEVC intra-frame prediction is employed to reduce the spatial redundancy in one frame. This type of prediction was an innovation of the H.264/AVC standard, used for the first time in widespread video coding schemes (RICHARDSON, 2003). The intra prediction is performed in the spatial domain by using neighbor samples (redundancy reference space) already coded and reconstructed to represent the current block. In the H.264/AVC standard, the intra prediction of the current block is performed using the samples from left-up and top-right regions. Besides, the intra prediction could be performed either over a 16x16 block (fixed macroblock size) or, alternatively, over a 4x4 block. Considering the available neighbor samples, there were up to nine different modes that could be applied on a 4x4 block and four modes on a 16x16 block.

Although intra prediction is still performed in spatial domain in current HEVC, neighbor samples from the left-down region may be used as context samples for the prediction. The most important is that it can provide up to 34 intra prediction modes depending on the PU size considering the emerging HEVC standard. This is a large increase when compared with only nine prediction modes available in the H.264/AVC.

The directions in HEVC intra prediction have angles of +/- [0, 2, 5, 9, 13, 17, 21, 26, 32]/32. The angle is given by a displacement of the bottom row of the block and the reference row above the block in case of vertical prediction, or displacement of the rightmost column of the block and the reference column left from the block in case of horizontal prediction (MIN, 2010). The samples of the target block can be predicted by linearly extrapolating the reference samples at 1/32th pixel accuracy for all block sizes instead of different precision for different block sizes. Besides DC prediction mode, the 33 possible intra prediction directions are illustrated in figure 3.7 where an 8x8 block size is used as example. In the HM 5.1 software, the calculation of the predicted samples is performed following defined tables, where each angle of the directional modes has a defined value (JCT-VC, 2012).

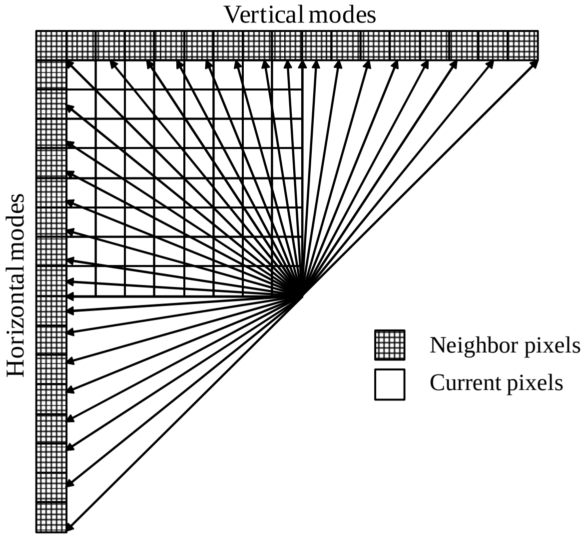


Figure 3.7: 33 intra prediction directions in emerging HEVC standard.

Besides the coding unit (CU) sizes, for intra prediction in HEVC each CU can be also partitioned into PUs, where the prediction steps are actually applied. Two partition sizes are supported for the HEVC intra prediction process, *2Nx2N* and *NxN*, where *N* is ½ of the target CU size (example: for a 64x64 CU, *N* is 32). Figure 3.8 shows both possible types of partition.

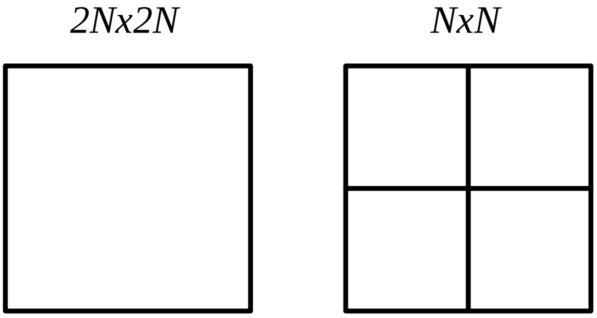


Figure 3.8: Possible types of CU partition in HEVC intra prediction.

The *NxN* is used only in the CU quad tree leafs, i.e., for 8x8 CUs (JCT-VC, 2012). This way, for the HEVC intra prediction perspective the PUs sizes varying from 4x4 up to 64x64.

The number of intra prediction modes supported for each PU size varies according to the availability of the reference neighbors. Table 3.2 (JCT-VC, 2012) shows the number of intra modes for each PU size in HEVC and the number of intra modes for H.264/AVC. The DC mode is also considered.

Table 3.2: Number of intra prediction modes for each PU size.

| PU Size | Number of Intra Modes HEVC | Number of Intra Modes H.264* |
|---|---|---|
| 64x64 | 3 | - |
| 32x32 | 34 | - |
| 16x16 | 34 | 4 |
| 8x8 | 34 | - |
| 4x4 | 17 | 9 |

*H.264/AVC main profile

## 3.4 Rate Distortion Optimization (RDO)

Considering all CU, PU and TU sizes, besides the number of intra prediction modes (Table 3.2), the HEVC encoder must choose the best combination of CU, PU and TU sizes and intra modes to achieve a good relation in terms of visual quality and compression efficiency. As digital videos have different content characteristics, the decision of which CU, PU and TU size and the intra prediction mode that will be used is very important to achieve a good coding result considering visual quality and compression efficiency.

There is no definition about which prediction mode or block size will be used for a particular block. However, the Rate Distortion Optimization (RDO) (SULLIVAN and WIEGAND, 1998) is the most accepted technique in the literature to perform this choice of block sizes and prediction modes. The RDO was proposed to generate the best coding with respect to the video visual quality (distortion) and compression efficiency (bit-rate). Equation (3.1) shows how the rate-distortion (RD) calculation is performed.

$$J = D + \lambda . R \tag{3.1}$$

In (3.1), **D** means the distortion level of the coded and reconstructed image, **R** means the bit-rate and **J** is the RD cost. The λ parameter is called Lagrange multiplier and it relates distortion and bit-rate according to the quantization strength used in the encoding process.

The main goal of the RDO technique is find the best combination of block sizes and prediction modes targeting the **J** cost minimization, which represents the best relation considering bit-rate (amount of bits after the encoding process) and video quality (distortion between encoded and original videos). In the full RDO decision process the **R** and **D** values are exhaustively calculated for all possible prediction modes and all possible block sizes. However, the calculation of these values is not an easy task. The **R** (bit-rate) value is available only after the prediction, forward transforms, forward quantization and entropy coding. The **D** (distortion) value can be calculated only after the prediction, forward transforms and quantization and inverse transforms and quantization (reconstruction process). It means that the **R** and **D** values will only be available when the whole coding and decoding processes are finished for each possible combination of available block sizes and coding modes. Figure 3.9 shows a diagram of this process.

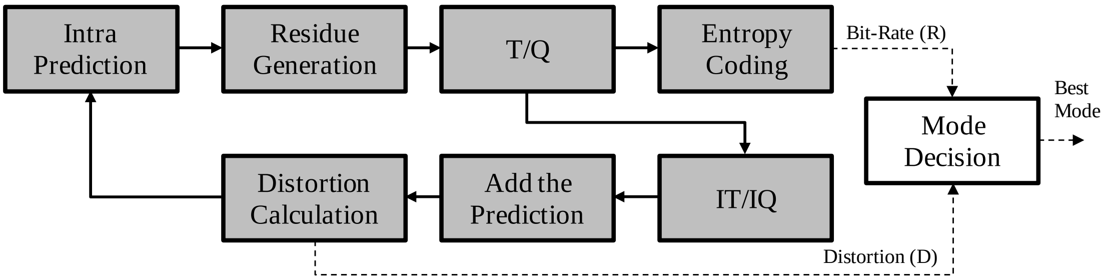


Figure 3.9: Diagram of RDO-based decision.

As previously mentioned, in the HEVC intra prediction process there are lots of encoding possibilities considering all modes and all block sizes. This way, the computational complexity involved in the intra prediction mode decision process using the RDO technique for all possible combinations is very high.

## 3.5 Intra Prediction Mode Decision in HM

The mode decision evaluation implemented in HM 5.1 for the intra prediction follows the depth-first search order in the quad tree coding structure. It means that PUs of nodes above in the quad tree structure are evaluated first (top-down approach) by the RDO technique. Figure 3.10 shows an example of the order used in HM to evaluate the PUs in each CU node, where the numbers inside the nodes mean the mode decision evaluation order in the quad-tree structure.

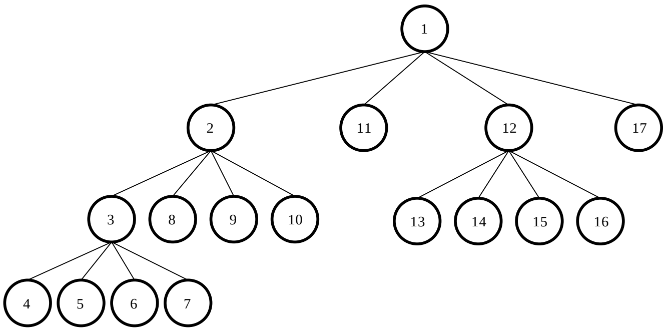


Figure 3.10: Mode decision order in the quad tree coding structure in HM 5.1.

This recursive structure is formed as follows: first, each frame is initially divided into CUs of 64x64 pixels. This initial CU is defined as depth 0 and the RDO evaluation is performed over the PU size (*2Nx2N* = 64x64) considering all intra prediction modes available. After the RDO evaluation, the 64x64 CU is split into four identical smaller CUs with 32x32 pixels (depth 1). Again, the RDO process is applied to the 32x32 PU (*2Nx2N*) of the first CU. After that, this first 32x32 CU is once again divided in four 16x16 CUs, where the PU (*2Nx2N* = 16x16) of the first one is evaluated by the RDO considering the corresponding intra prediction modes. Finally, the first 16x16 CU is once again divided into four 8x8 CUs and the RDO is performed considering the available intra prediction modes over the two possible partition sizes (*2Nx2N* and *NxN*, 8x8 and 4x4 PUs respectively).

It is important to notice that, in the intra prediction process, the PUs and CUs sizes are equal, since the *2Nx2N* partition is always used. The 8x8 CU is the exception since both *2Nx2N* and *NxN* partitions are used generating the 8x8 and 4x4 PUs sizes. The other CUs are evaluated on the way back in the recursive structure. Figure 3.11 shows a pseudo-code for this process.

```
encodeCU_intra(CU node)
    if (node.size == 8) //leafs
        checkRDO_PUs(node)
        return
    else
        checkRDO_PUs(node)
        encodeCU_intra(node.CU0)
        encodeCU_intra(node.CU1)
        encodeCU_intra(node.CU2)
        encodeCU_intra(node.CU3)
```

Figure 3.11: Pseudo-code of the RDO evaluation order in HM 5.1.

In the default configuration of HM 5.1 reference software the full RDO evaluation is not performed considering all intra prediction modes, since it would cause a great burden to the encoder. According to the PU size, there are a limited number of intra prediction candidate modes that are fully evaluated by the RDO process, which was initially proposed in (ZHAO, ZHANG, *et al.*, 2011) and firstly adopted in the HM 2.0. Table 3.3 shows the number of full RDO evaluations for each PU size in comparison with all possible modes.

Table 3.3: Number of candidate modes for full RDO evaluation for each PU size.

| **PU Size** | **Number of modes for full RDO** | **Number of possible modes** |
|---|---|---|
| 64x64 | 3 | 3 |
| 32x32 | 3 | 34 |
| 16x16 | 3 | 34 |
| 8x8 | 8 | 34 |
| 4x4 | 8 | 17 |

For the 64x64 PUs all three possible intra prediction modes (according Table 3.3) are fully evaluated by the RDO decision process. The candidates for other PU sizes are chosen based on a pre-evaluation of all available modes. Each intra prediction mode is performed and the predicted block is generated. Instead of doing all coding and reconstruction process to generate the distortion and bit-rate costs for the target mode, a simpler and less computational complexity cost is calculated. The distortion cost is performed by only applying the absolute sum of Hadamard transform coefficients over the residual data (i.e., before the transforms step) and the bit-rate cost is performed by an estimation based on the neighbor chosen modes. This way, only the candidates with more chance of being chosen are fully evaluated by the RDO process.

Even with this simplification adopted as default in the HM reference software, the intra prediction is still a highly demanding computational step in HEVC encoders, since the full RDO process still needs to be performed several times in the quad-tree structure to choose which CU and PU sizes, besides the intra prediction modes, will generate the best rate-distortion cost. This way, alternative and more efficient intra mode decision engines should be developed in order to reduce the time spent in the intra prediction process in HEVC encoders. This work aims that goal, since it exploits not only fast decision algorithms for the intra prediction process but also dedicated hardware architecture solutions based on these fast intra prediction algorithms. With both approaches gathered it is possible to achieve more efficient solutions to solve the intra prediction mode decision process in HEVC encoders.

# 4 FAST INTRA PREDICTION MODE DECISION ALGORITHMS

Based on the problem presented in the previous chapter about the complexity of choose the best intra prediction mode, this chapter describes the proposed fast mode decision algorithms developed in this work. Firstly, the motivation and observations are pointed out to provide the guidelines to better understand the correlation used as the main idea behind the proposed fast algorithms. A new evaluation order in the quad-tree coding structure is also proposed as base to use the new fast intra decision algorithms.

## 4.1 Preliminary evaluation

The main idea behind the proposed algorithms exploits the coding correlation presented in the quad-tree structure used in the HEVC emerging standard. The LCU (Largest Coding Unit, 64x64) is recursively divided into smaller CUs until the maximum depth allowed (8x8 in HM 5.1 software) and the intra prediction modes are applied in the PUs generated by this division. Considering this quad-tree coding structure, we have observed that there is a correlation between the prediction modes used in different depth levels, which can be exploited for faster prediction. For example, if the intra prediction modes over four neighbors 8x8 PUs are similar, there is a probability that the 16x16 PU above in the quad-tree, which represents the same image region, uses a similar intra prediction mode.

Based on that idea, several profile executions in the HM 5.1 software were performed to measure this correlation among the intra prediction modes in one depth level to another level above. Table 4.1 and Table 4.2 show the percentage of modes that are similar between two depths in high efficiency and low complexity conditions (ISO/IEC-JCT1/SC29/WG11, 2011), respectively. The percentage was calculated considering the modes used for four PUs in a depth $n$ in comparison with the mode used by the PU above (depth $n+1$), representing the same image region. This way, if the intra mode used in depths $n+1$ is one of the four modes used in the depth $n$, we assume that there is a correlation between the intra prediction modes in these two depth levels. The results were obtained using sequences in different resolutions according to the common test conditions specified in (ISO/IEC-JCT1/SC29/WG11, 2011). The classes A to E are related to different digital video sequences recommended to be used in HEVC tests. The 22, 27, 32 and 37 QPs (quantization parameters) used are also the ones recommended to be used in HEVC tests (ISO/IEC-JCT1/SC29/WG11, 2011).

Table 4.1: Correlation among modes in different depths in High Efficiency condition.

| QP | A | B | C | D | E |
|---|---|---|---|---|---|
| 22 | 50.42% | 50.70% | 47.12% | 41.87% | 57.02% |
| 27 | 54.72% | 58.29% | 49.96% | 45.19% | 63.33% |
| 32 | 58.99% | 64.60% | 53.52% | 51.21% | 68.21% |
| 37 | 63.04% | 69.98% | 58.69% | 57.93% | 72.97% |

Table 4.2: Correlation among modes in different depths in Low Complexity condition.

| QP | A | B | C | D | E |
|---|---|---|---|---|---|
| 22 | 47.24% | 47.56% | 45.15% | 40.30% | 55.62% |
| 27 | 52.33% | 55.70% | 48.16% | 43.13% | 62.37% |
| 32 | 57.28% | 62.46% | 51.80% | 49.36% | 67.26% |
| 37 | 62.10% | 68.64% | 58.03% | 55.26% | 71.66% |

Considering the results presented in Table 4.1 and Table 4.2, it is possible to see that there is a high correlation between the intra modes used in one depth to the intra mode used in the depth above when the same image region is considered, varying from 40% (class D, QP 22) up to almost 73% (class E, QP 37). However, this behavior is not explored in the current HM mode decision process.

In this context, this work proposes the use of this correlation information as base to the intra prediction mode decision in HEVC encoders. This way, it is possible to accelerate the intra prediction process using these intra modes already decided for the immediately smaller PU depth as reference to evaluate the intra mode of the current depth.

## 4.2 New mode decision evaluation order in the quad-tree structure

In HM software, the quad-tree structure is evaluated by the RDO following the depth first search order (top-down figure 3.10). However, this evaluation order does not allow the use of intra modes of the levels below as reference to perform the prediction process in the current level, since these modes were not chosen yet. This way, in this work we propose a new evaluation order to allow the use of this correlation information mentioned in the previous section as reference for the intra prediction mode decision.

Instead of first evaluate the current CU and then evaluate the CUs in the next depth, the CUs in lowest depths are first evaluated (bottom-up approach). This new evaluation order does not modify the quad-tree structure construction, since it only changes the evaluation order of the mode decision process in the quad-tree. This way, the coding steps are not affected at all by this modification and they can be performed considering the regular quad-tree structure. Figure 4.1 illustrates the alternative evaluation order (represented by the numbers inside the nodes) used in this work.

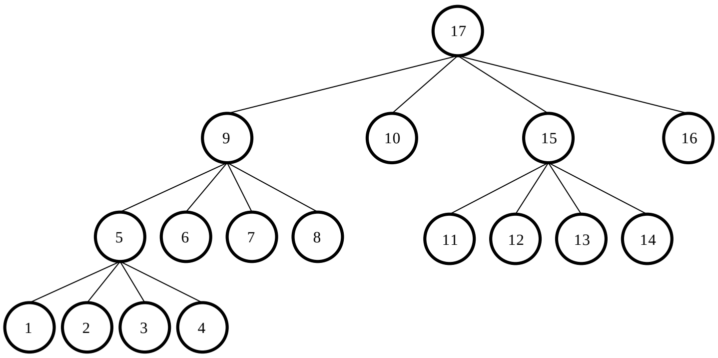


Figure 4.1: New mode decision evaluation order in the quad-tree structure.

Considering this new evaluation order (bottom-up), it is possible to use the intra decisions used in the depths below as reference to the decisions in the depths above. For example, the modes chosen by the CUs 1, 2, 3 and 4 in Figure 4.1 can be used as reference to the mode decision process for the CU 5. Figure 4.2 shows a pseudo-code which demonstrates the modification performed in this work to allow the use of the intra modes of one depth as reference to evaluate the depth above. The only modification was to evaluate the PUs after the encoding of the CUs below. In Figure 4.2 *m0, m1, m2* and *m3* are the best modes in the depth below.

```
encodeCU_intra(CU node)
    if (node.size == 8) //leafs
          checkRDO_PUs(node)
          return node.bestMode
    else
          m0 = encodeCU_intra(node.CU0)
          m1 = encodeCU_intra(node.CU1)
          m2 = encodeCU_intra(node.CU2)
          m3 = encodeCU_intra(node.CU3)
    checkRDO_PUs(node, m0, m1, m2, m3)
    return node.bestMode
```

Figure 4.2: Pseudo-code of the new mode decision evaluation order.

Based on this possibility of use the intra modes of one depth as reference to the decision of depths above, three different heuristic algorithms were developed with the main goal of accelerate the intra mode decision of the HEVC emerging standard. The algorithms were developed aiming different complexity levels with minimum loss in the rate-distortion performance comparing with the regular mode decision in HM 5.1.

## 4.3 First Decision Reuse (FDR) Heuristic

The first heuristic developed in this work was based on the correlation between neighbor samples used for the intra prediction process in one depth and the neighbors that will be used in the depths above. It means that the chosen mode for the first PU in depth *n-1* was based on a neighbor reference space that will be part of the neighbor reference space in the prediction of depth *n*.

Figure 4.3 illustrates the PU of depth *n-1* which uses part of the neighbor samples (in gray) that will be used to perform the intra prediction for the depth *n*.

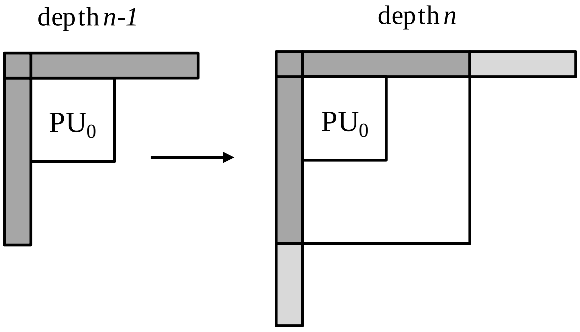


Figure 4.3: Correlation between neighbor samples used in depths *n-1* and *n*.

In the intra prediction process, the first PU in the depth *n-1* uses a great part of the neighbor samples that will be used as reference to perform the intra prediction of the PU in depth *n*. This way, the First Decision Reuse (FDR) heuristic considers that among all four PUs in the depth *n-1*, the intra mode of the first PU is the one which has more correlation with the intra mode of the PU in depths *n* and then it will be reused in the current decision.

Considering this neighbor reference space correlation, the first heuristic is the simplest one developed in this work. The evaluation used in the HM software, i.e., full RDO process over candidates selected by the default fast search (Hadamard plus bits estimation), is applied only to the deepest level in the quad-tree structure, since the PUs in levels above will use only the mode generated by the first PU from the partition of the above CU. This decision highly simplifies the mode decision process in one quad-tree structure, since lots of fast searches (Hadamard and bits estimation) plus several full RDO evaluations are skipped. Figure 4.4 shows how the mode is chosen based on neighbor correlation between depths. In this example, mode 13 is used in the first PU of depth *n-1* and it is also used in the PU of depth *n*.

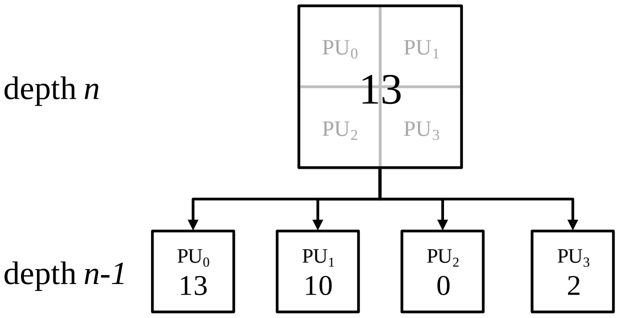


Figure 4.4: Candidate selection for mode decision algorithm based on correlation of neighbor reference samples.

## 4.4 Majority Decision Reuse (MDR) Heuristic

The Majority Decision Reuse (MDR) heuristic algorithm developed in this work was based on the most used intra mode in the PUs of the depth below, i.e., the candidate modes for the depth *n* are those modes that occurs most often in the PUs of depth *n-1*. It means that if one mode is mostly used in one quad-tree level, there is a great chance of this mode to be chosen as the best for the level above. This way, the mode statistics value is used to select the candidates for the full RDO process.

The fast search used in the HM (Hadamard and bits estimation) is still avoided and the number of full RDO evaluation is reduced in comparison with the default mode decision in HM. The number of candidates for full RDO evaluation can be: one when there are two, three or four equal modes in the level below; two when two modes are equal twice; and, finally, four when the modes are all different from each other. Figure 4.5 shows an example of how the candidate modes set can be formed. The candidate modes set is composed by two modes 13 and 0, since these modes both occur twice in the level below.

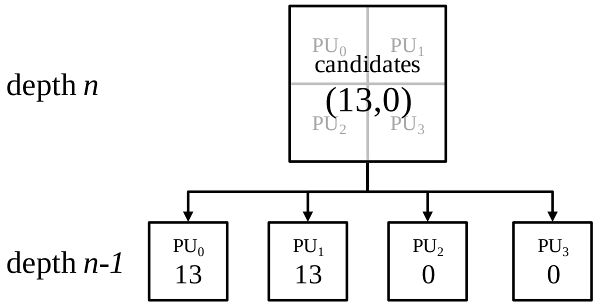


Figure 4.5: Candidate selection for mode decision algorithm based on mode statistic value.

## 4.5 Complete Decision Reuse (CDR) Heuristic

The Complete Decision Reuse (CDR) heuristic algorithm proposed in this work still uses the main idea presented in section 4.2 (the use of intra mode decision information from PUs in below quad-tree structure levels), however, it does not limit the number of candidates for the full RDO process. That is, all different intra decision modes used in levels below are used as candidates for the intra prediction in the level above. This heuristic fully uses the main idea proposed in section 4.2, since it uses all information generated by the intra decision in levels immediately below.

The number of candidates varies from one up to four and the fast search performed in the HM software is avoided as in the other presented heuristics. The number of full RDO evaluations is reduced as well. Figure 4.6 illustrates an example of how the candidates set can be formed. The set is composed by all intra modes used in level below (0, 3, 7 and 13).

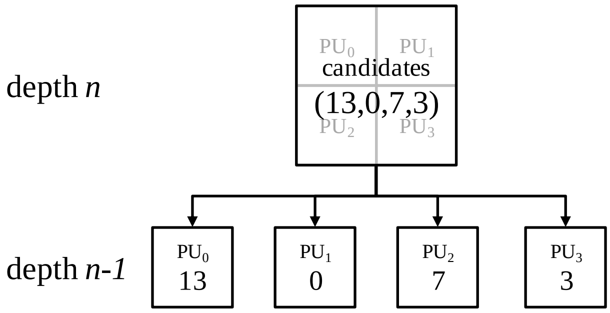


Figure 4.6: Candidate selection for mode decision algorithm based on the CDR heuristic.

## 4.6 Results

The performance of the proposed heuristic algorithms was evaluated in the HM 5.1 software. The experiments were carried out according to the recommended sequences and quantization parameters specified in (ISO/IEC-JCT1/SC29/WG11, 2011), for both high efficiency and low complexity test conditions. The Intra only encoder configuration was used, since this work proposes fast algorithms for the intra mode decision. The test platform used was a 2.67GHz Core i5 Intel Processor with 4 cores and 4GB RAM. The results were measured considering bit-rate increase, PSNR loss and encoding time savings. The default mode decision in HM 5.1 was used as reference.

Besides this heuristic implementation, all algorithms were evaluated varying the maximum depth level from 8 to 64 (8, 16, 32, and 64). It means that a proposed heuristic can be limited to a defined target level. For example, if the target level is 16 the algorithm is applied from the 8x8 PUs until the 16x16 PUs. For the 32x32 and 64x64 PUs the heuristics are not considered and the regular HM intra mode decision is used.

Table 4.3, Table 4.4, and Table 4.5 show the results reached by the three heuristics: FDR, MDR and CDR, respectively and the target level is from 8 up to 64. The results for each video class (A to E) are presented considering the average results among all evaluated QPs (22, 27, 32, 37). The complete results are presented in the APENDIX.

Table 4.3: Results for the First Decision Reuse (FDR) heuristic.

| Class | **Target level 8** | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| | High Efficiency | | | Low Complexity | | |
| | Δ Bitrate (%) | Δ PSNR (dB) | Δ Time (%) | Δ Bitrate (%) | Δ PSNR (dB) | Δ Time (%) |
| A | +1.98 | -0.06 | -8.91 | +1.80 | -0.11 | -14.40 |
| B | +0.91 | -0.05 | -9.58 | +0.88 | -0.07 | -14.64 |
| C | +1.77 | -0.05 | -9.69 | +1.90 | -0.06 | -14.26 |
| D | +1.68 | -0.04 | -10.62 | +1.76 | -0.06 | -14.76 |
| E | +2.11 | -0.07 | -8.80 | +2.02 | -0.10 | -14.35 |
| Average | **+1.69** | **-0.05** | **-9.52** | **+1.67** | **-0.08** | **-14.48** |
| Class | **Target level 16** | | | | | |
| | High Efficiency | | | Low Complexity | | |
| | Δ Bitrate (%) | Δ PSNR (dB) | Δ Time (%) | Δ Bitrate (%) | Δ PSNR (dB) | Δ Time (%) |
| A | +5.31 | -0.10 | -13.05 | +4.98 | -0.16 | -21.59 |
| B | +2.90 | -0.07 | -13.92 | +2.87 | -0.10 | -21.96 |
| C | +4.12 | -0.06 | -14.48 | +4.23 | -0.07 | -21.49 |
| D | +3.60 | -0.04 | -15.48 | +3.53 | -0.06 | -22.03 |
| E | +6.24 | -0.11 | -12.66 | +5.99 | -0.16 | -21.55 |
| Average | **+4.44** | **-0.08** | **-13.92** | **+4.32** | **-0.11** | **-21.72** |
| Class | **Target level 32** | | | | | |
| | High Efficiency | | | Low Complexity | | |
| | Δ Bitrate (%) | Δ PSNR (dB) | Δ Time (%) | Δ Bitrate (%) | Δ PSNR (dB) | Δ Time (%) |
| A | +7.20 | -0.09 | -17.46 | +6.66 | -0.16 | -28.85 |
| B | +4.49 | -0.08 | -18.34 | +4.27 | -0.11 | -28.92 |
| C | +5.19 | -0.06 | -19.46 | +5.22 | -0.07 | -28.69 |
| D | +4.25 | -0.04 | -20.48 | +4.13 | -0.06 | -29.06 |
| E | +8.96 | -0.13 | -16.74 | +8.58 | -0.19 | -28.94 |
| Average | **+6.02** | **-0.08** | **-18.50** | **+5.77** | **-0.12** | **-28.89** |

| Class | Target level 64 | | | | | |
|---|---|---|---|---|---|---|
| | High Efficiency | | | Low Complexity | | |
| | Δ Bitrate (%) | Δ PSNR (dB) | Δ Time (%) | Δ Bitrate (%) | Δ PSNR (dB) | Δ Time (%) |
| A | +7.97 | -0.08 | -22.32 | +7.31 | -0.15 | -36.05 |
| B | +5.94 | -0.07 | -22.96 | +5.53 | -0.11 | -35.89 |
| C | +5.42 | -0.05 | -24.34 | +5.44 | -0.06 | -35.76 |
| D | +4.40 | -0.04 | -24.17 | +4.24 | -0.06 | -34.92 |
| E | +11.22 | -0.13 | -21.73 | +10.59 | -0.19 | -36.39 |
| Average | **+6.99** | **-0.08** | **-23.10** | **+6.62** | **-0.11** | **-35.80** |

Table 4.4: Results for the Majority Decision Reuse (MDR) heuristic.

| Class | Target level 8 | | | | | |
|---|---|---|---|---|---|---|
| | High Efficiency | | | Low Complexity | | |
| | Δ Bitrate (%) | Δ PSNR (dB) | Δ Time (%) | Δ Bitrate (%) | Δ PSNR (dB) | Δ Time (%) |
| A | +1.59 | -0.06 | -8.66 | +1.42 | -0.10 | -13.83 |
| B | +0.67 | -0.05 | -9.05 | +0.63 | -0.06 | -14.37 |
| C | +1.33 | -0.04 | -8.98 | +1.41 | -0.05 | -13.38 |
| D | +1.28 | -0.04 | -9.52 | +1.32 | -0.05 | -13.89 |
| E | +1.69 | -0.06 | -8.47 | +1.60 | -0.08 | -14.04 |
| Average | **+1.31** | **-0.05** | **-8.93** | **+1.28** | **-0.07** | **-13.90** |
| Class | **Target level 16** | | | | | |
| | High Efficiency | | | Low Complexity | | |
| | Δ Bitrate (%) | Δ PSNR (dB) | Δ Time (%) | Δ Bitrate (%) | Δ PSNR (dB) | Δ Time (%) |
| A | +3.56 | -0.08 | -12.58 | +3.24 | -0.14 | -20.57 |
| B | +1.77 | -0.06 | -13.36 | +1.66 | -0.08 | -21.77 |
| C | +2.64 | -0.05 | -13.35 | +2.65 | -0.07 | -19.92 |
| D | +2.28 | -0.04 | -13.90 | +2.19 | -0.06 | -20.17 |
| E | +4.29 | -0.09 | -12.08 | +4.03 | -0.14 | -21.11 |
| Average | **+2.91** | **-0.06** | **-13.05** | **+2.75** | **-0.10** | **-20.71** |
| Class | **Target level 32** | | | | | |
| | High Efficiency | | | Low Complexity | | |
| | Δ Bitrate (%) | Δ PSNR (dB) | Δ Time (%) | Δ Bitrate (%) | Δ PSNR (dB) | Δ Time (%) |
| A | +4.58 | -0.08 | -16.66 | +4.12 | -0.14 | -27.47 |
| B | +2.62 | -0.06 | -17.54 | +2.39 | -0.09 | -28.46 |
| C | +3.13 | -0.05 | -17.79 | +3.10 | -0.07 | -26.35 |
| D | +2.57 | -0.04 | -18.00 | +2.45 | -0.06 | -26.39 |
| E | +5.89 | -0.10 | -16.36 | +5.51 | -0.15 | -27.89 |
| Average | **+3.76** | **-0.07** | **-17.27** | **+3.51** | **-0.10** | **-27.31** |
| Class | **Target level 64** | | | | | |
| | High Efficiency | | | Low Complexity | | |
| | Δ Bitrate (%) | Δ PSNR (dB) | Δ Time (%) | Δ Bitrate (%) | Δ PSNR (dB) | Δ Time (%) |
| A | +4.93 | -0.07 | -21.21 | +4.39 | -0.14 | -34.40 |
| B | +3.31 | -0.06 | -21.85 | +2.95 | -0.09 | -35.04 |

| | | | | | | |
|---|---|---|---|---|---|---|
| C | +3.19 | -0.05 | -22.14 | +3.17 | -0.07 | -32.80 |
| D | +2.62 | -0.04 | -21.50 | +2.48 | -0.06 | -31.83 |
| E | +6.77 | -0.10 | -20.98 | +6.26 | -0.16 | -34.99 |
| Average | **+4.16** | **-0.06** | **-21.54** | **+3.85** | **-0.10** | **-33.81** |

Table 4.5: Results for the complete decision reuse (CDR) heuristic.

| Class | **Target level 8** | | | | | |
|---|---|---|---|---|---|---|
| | High Efficiency | | | Low Complexity | | |
| | Δ Bitrate (%) | Δ PSNR (dB) | Δ Time (%) | Δ Bitrate (%) | Δ PSNR (dB) | Δ Time (%) |
| A | +1.28 | -0.05 | -8.10 | +1.14 | -0.08 | -12.90 |
| B | +0.51 | -0.04 | -8.63 | +0.48 | -0.05 | -13.18 |
| C | +1.05 | -0.03 | -8.36 | +1.11 | -0.05 | -12.44 |
| D | +1.00 | -0.03 | -8.71 | +1.03 | -0.04 | -12.64 |
| E | +1.33 | -0.05 | -7.85 | +1.25 | -0.07 | -13.40 |
| Average | **+1.03** | **-0.04** | **-8.33** | **+1.00** | **-0.06** | **-12.91** |
| Class | **Target level 16** | | | | | |
| | High Efficiency | | | Low Complexity | | |
| | Δ Bitrate (%) | Δ PSNR (dB) | Δ Time (%) | Δ Bitrate (%) | Δ PSNR (dB) | Δ Time (%) |
| A | +2.66 | -0.06 | -11.69 | +2.40 | -0.10 | -18.79 |
| B | +1.27 | -0.04 | -12.35 | +1.20 | -0.06 | -19.24 |
| C | +1.90 | -0.04 | -11.84 | +1.94 | -0.06 | -18.02 |
| D | +1.59 | -0.03 | -12.12 | +1.53 | -0.05 | -18.33 |
| E | +3.10 | -0.07 | -11.22 | +2.90 | -0.10 | -19.51 |
| Average | **+2.10** | **-0.05** | **-11.85** | **+2.00** | **-0.07** | **-18.78** |
| Class | **Target level 32** | | | | | |
| | High Efficiency | | | Low Complexity | | |
| | Δ Bitrate (%) | Δ PSNR (dB) | Δ Time (%) | Δ Bitrate (%) | Δ PSNR (dB) | Δ Time (%) |
| A | +3.34 | -0.06 | -15.03 | +3.00 | -0.11 | -24.29 |
| B | +1.78 | -0.04 | -15.99 | +1.66 | -0.07 | -25.10 |
| C | +2.19 | -0.04 | -15.35 | +2.21 | -0.06 | -23.18 |
| D | +1.79 | -0.03 | -15.12 | +1.70 | -0.05 | -22.89 |
| E | +4.19 | -0.07 | -14.96 | +3.94 | -0.11 | -25.61 |
| Average | **+2.66** | **-0.05** | **-15.29** | **+2.50** | **-0.08** | **-24.21** |
| Class | **Target level 64** | | | | | |
| | High Efficiency | | | Low Complexity | | |
| | Δ Bitrate (%) | Δ PSNR (dB) | Δ Time (%) | Δ Bitrate (%) | Δ PSNR (dB) | Δ Time (%) |
| A | +3.54 | -0.05 | -18.56 | +3.17 | -0.10 | -29.99 |
| B | +2.26 | -0.04 | -19.37 | +2.07 | -0.07 | -30.60 |
| C | +2.22 | -0.04 | -18.56 | +2.22 | -0.06 | -28.10 |
| D | +1.82 | -0.03 | -17.52 | +1.74 | -0.05 | -26.72 |
| E | +4.70 | -0.07 | -18.66 | +4.37 | -0.11 | -31.46 |
| Average | **+2.91** | **-0.05** | **-18.53** | **+2.71** | **-0.08** | **-29.37** |

The results show that the FDR heuristic is the most time saving, since it uses only the mode of the first PU in the depth below. It can save until 35.8% of encoding time in average considering the low complexity test condition when it is applied to the whole quad-tree structure, i.e., when the target level is 64. The bit-rate increasing varies from 1.69% up to 6.99% according to the target level. The image quality loss is negligible being 0.11dB in the worst case.

The MDR heuristic is a middle term considering complexity among the three proposed mode decision algorithms. This heuristic selects the candidates for full RDO process based on the mostly used mode in the quad-tree level below. It can provide up to 33.81% of encoding time savings for the low complexity test condition when the target level is 64. The bit-rate increase is 4.16% in the worst case (high efficiency condition and target level 64) and the image quality loss is negligible, being 0.1dB in the worst case.

The CDR heuristic can be classified as the best one among all the proposed heuristics. It selects the candidate modes for full RDO process using all intra modes information from the quad-tree levels below. Even using all modes, it can save up to almost 30% in encoding time considering the low complexity test condition at level 64, which is very close to the results presented by the two other heuristics. On top of that, the rate-distortion results are the best, since the bit rate increase is only 2.91% (high efficiency at target level 64) and the PSNR loss is limited to only 0.08dB, both considering the worst result.

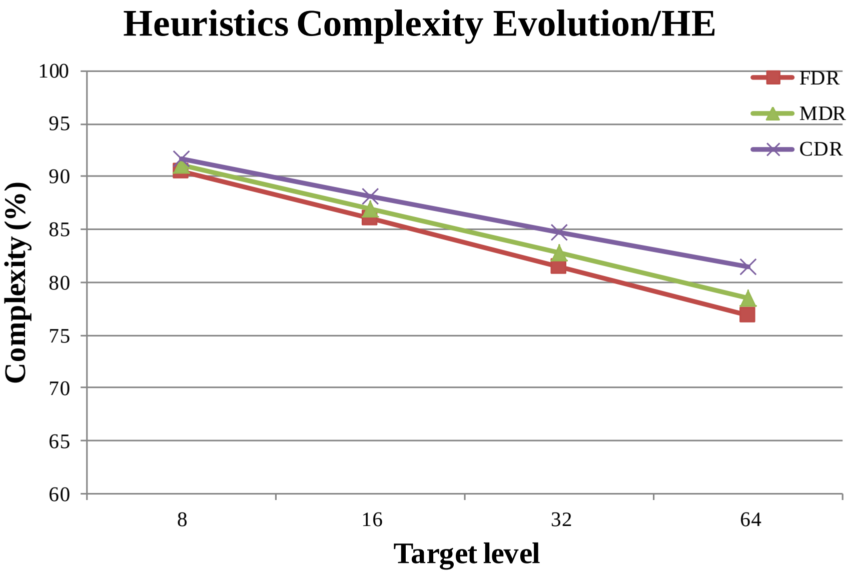
All three heuristics can be used in different scenarios of complexity, energy consumption and rate distortion performance. The FDR heuristic should be used when processing time and battery are considered critical to the target application, since it can achieve almost 36% of encoding time saving with a good rate-distortion performance, hence decreasing the energy consumption. When portable devices are being considered, battery life can be the most important issue. In this case, the FDR heuristic in the highest level (64) may be used to save energy. The rate-distortion performance will obviously be the worst among the three options.

The CDR heuristic should be used when rate-distortion performance is an issue to be considered. It achieved the best results in these terms (only 2.91% of bit-rate increase and 0.08dB in PSNR loss, in the worst cases) and it still can achieve almost 19% of time encoding savings for high efficiency and almost 30% for low complexity test conditions, also reducing energy consumption.

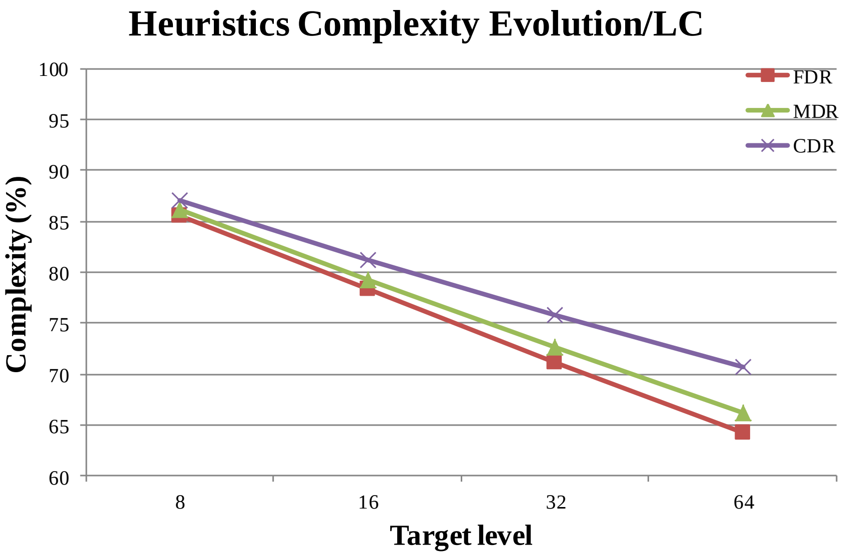
The MDR heuristic can be used when the relation between rate-distortion performance and complexity is balanced.

Besides the three different heuristics, it is possible to change the target level for each heuristic form 8 up to 64, increasing the possibilities of saving more or less encoding time according to the target requirements.

Figure 4.7 (a) and (b) show the relation between the three proposed heuristics in different target levels (8, 16, 32 and 64) considering the time savings for both (a) high efficiency (HE) and (b) low complexity (LC) test conditions. The complexity considers the difference between the average results in Table 4.3, Table 4.4, and Table 4.5 in comparison with the reference.



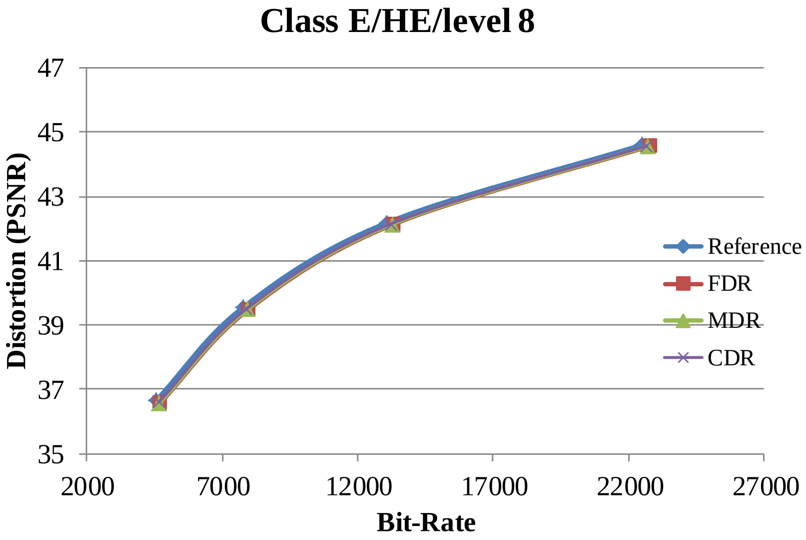
(a)



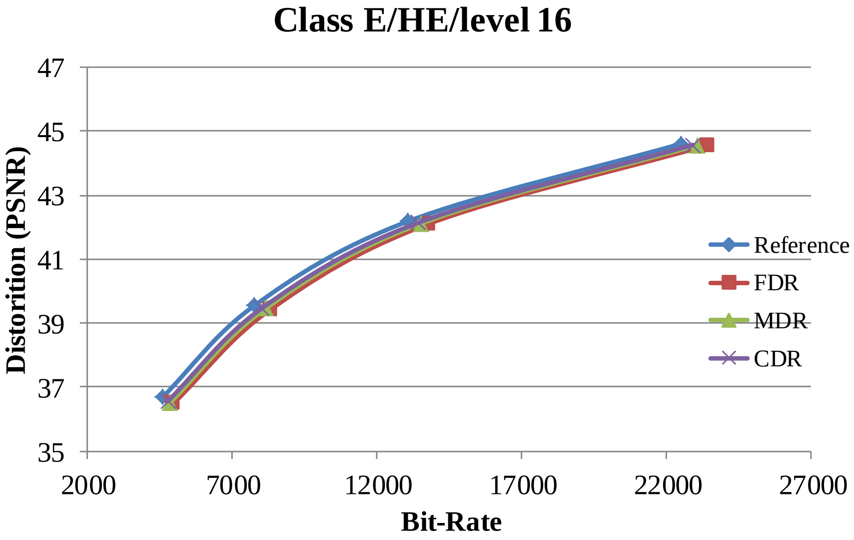
(b)

Figure 4.7: Complexity relation between the three proposed heuristics varying target levels (a) High Efficiency (b) Low Complexity.

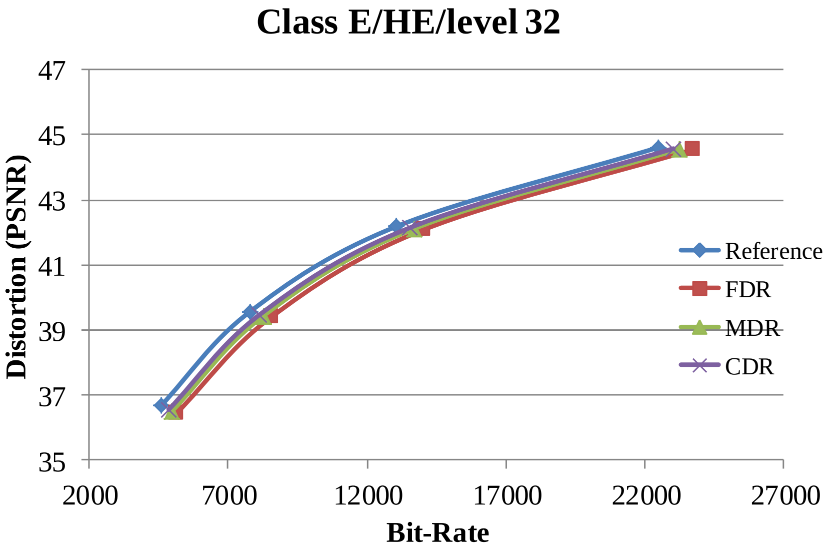
It is important to note that, as long as the heuristics go higher in the quad-tree structure, the encoding time is smaller. However, the rate-distortion results get worst, since the regular HM mode decision process is less used. The worst rate-distortion results presented by all heuristics are related to the class E sequences in the high efficiency test condition. However, from the rate-distortion curves in figure 4.8 (a), (b), (c), and (d), it is possible to observe that the proposed heuristics (mainly CDR) perform a close coding efficiency from low to high bit-rate compared to the reference. The curves also show how the rate-distortion is disturbed as long as the heuristics go higher in the quad-tree structure.


Class E/HE/level 8
Distortion (PSNR)
47
45
43
41
39
37
35
2000
7000
12000
17000
22000
27000
Bit-Rate
Reference
FDR
MDR
CDR

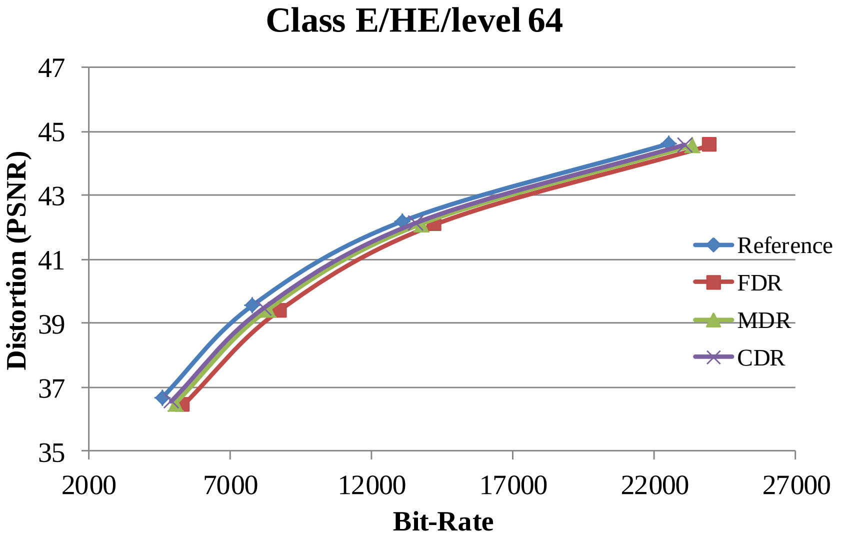
(a)


Class E/HE/level 16
Distoriton (PSNR)
47
45
43
41
39
37
35
2000
7000
12000
17000
22000
27000
Bit-Rate
Reference
FDR
MDR
CDR

(b)



(c)



(d)

Figure 4.8: Rate-distortion curve of class E sequences in High Efficiency condition (a) level 8, (b) level 16, (c) level 32 and (d) level 64.

Taking all these complexity reduction and compression efficiency results into consideration, it is possible to conclude that the investigation on fast intra mode decision algorithms can be a good solution to save time and energy when real HEVC intra coding systems are considered. In next chapter, a hardware architecture for the intra prediction process using one fast algorithm will be presented, and the importance of using fast algorithms will become more clear.

# 5 INTRA PREDICTION ARCHITECTURE

This chapter presents the architecture design for the intra-frame prediction process for HEVC encoders using the CDR (complete decision reuse) heuristic presented in section 4.5 to accelerate the intra prediction process. The architectural design was planned to allow real time processing for high resolution videos, such as HD1080p, i.e., high throughput is mandatory. Besides, another target of the architecture is to reduce the number of memory accesses which are required for the reference software implementation in the HM 5.1. The architecture was designed to support all five different PU sizes (64x64, 32x32, 16x16, 8x8 and 4x4) and all 34 different prediction modes (33 directional modes plus the DC mode). In the end of this chapter, the synthesis results are presented. A comparison is also presented considering the intra prediction architecture with no fast decision algorithm to show the importance of using this type of algorithms to improve the performance of intra prediction engines.

## 5.1 Architecture Data Path

In the intra prediction process, the predicted samples are generated either by a simple copy of the corresponding neighbor reference samples (integer predicted sample) or by an interpolation of two neighbor samples based on the intra prediction directional angle (fractional predicted sample). In the HM 5.1 software, the calculation of the predicted samples is performed following defined tables, where each angle of the directional modes has a defined value.

In order to simplify the data path of the designed architecture, the pixel prediction calculation was derived to simpler equations. The integer predicted pixels **PInt** are generated only by copying the corresponding neighbor as shown in equation (5.1). The fractional predicted pixels **PFract** are generated following equation (5.2). The constants $\mathbf{k_0}$ and $\mathbf{k_1}$ are calculated based on the prediction angle.

$$PInt_{(i,j)} = neighbor_l \tag{5.1}$$

$$\begin{gathered} PFrac_{\ i,j} \ = \ neighbor_l . k_0 + neighbor_{l+1} . k_1 \ + 16) \gg 5 \\ where\ k_0 + k_1 = 32 \end{gathered} \tag{5.2}$$

Based on these equations, the first design decision to increase throughput was remove the required multiplier in equation (5.1). This multiplication was decomposed in shifts and adds since $\boldsymbol{k_0}$ and $\boldsymbol{k_1}$ are constants with values between 1 and 31. Figure 5.1 shows a diagram of the fractional prediction architecture.

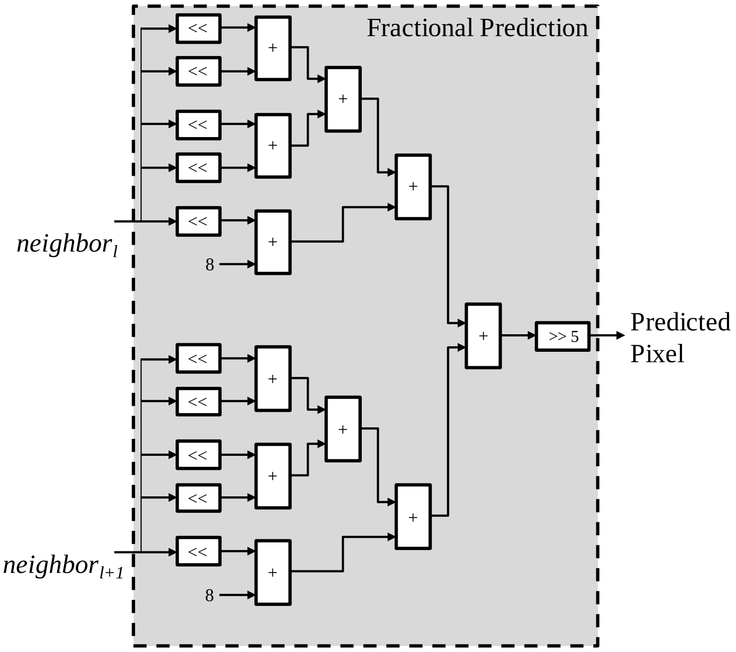


Figure 5.1 – Diagram of the fractional prediction architecture.

Each adder used in the intra prediction architecture was included in one independent pipeline stage to provide a critical path as smaller as possible. This way, the architecture data path is composed by 9 pipeline stages.

The second design decision to increase the designed architecture throughput was the data path replication in two parallel components (vertical data path and horizontal data path). The vertical data path is responsible to perform all vertical modes (17 modes) whereas the horizontal data path performs all horizontal modes (17 modes). With this data path replication, the parallelism level of the designed architecture was doubled, increasing the throughput. Mode 3 is performed in both data paths to maintain the architecture regularity. Figure 5.2 shows how the angular modes were divided to each data path. The DC and planar modes are not considered in the designed architecture.

Besides the throughput increase caused by the processing of two modes in parallel, this replication in two data paths provides some other benefits to the intra prediction architecture. The modes processed in parallel are those with equal direction angles (for example mode 18 and mode 26 from figure 5.2), since the constants (***k0*** and ***k1***) used in the pixel prediction and the addresses to access the neighbor pixels are equal, which highly simplifies the architecture control.

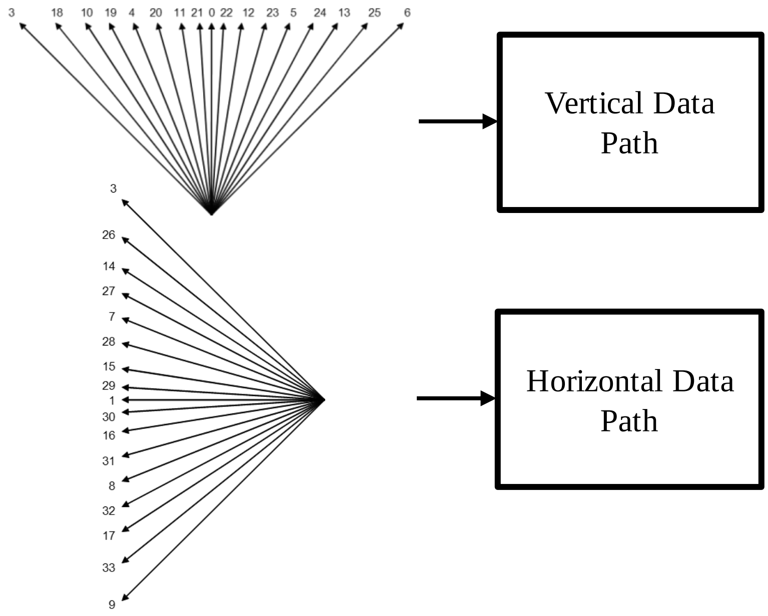


Figure 5.2: Division of modes in each data path.

The third design decision to increase throughput was the definition of the used parallelism in each data path (vertical and horizontal), i.e., how many samples will be processed in parallel by each data path. This decision was done through a high level evaluation considering: (a) use of hardware resources (area), (b) number of clock cycles necessary to process a whole block (64x64) prediction considering all sizes and all modes, and (c) the required memory bandwidth. Figure 5.3 shows a graph comparing five different parallelism levels (16, 8, 4, 2 and 1 samples) considering area and cycles. The parallelism level itself indicates the necessary memory bandwidth, since it defines how many samples must be read from the external memory at each clock cycle.

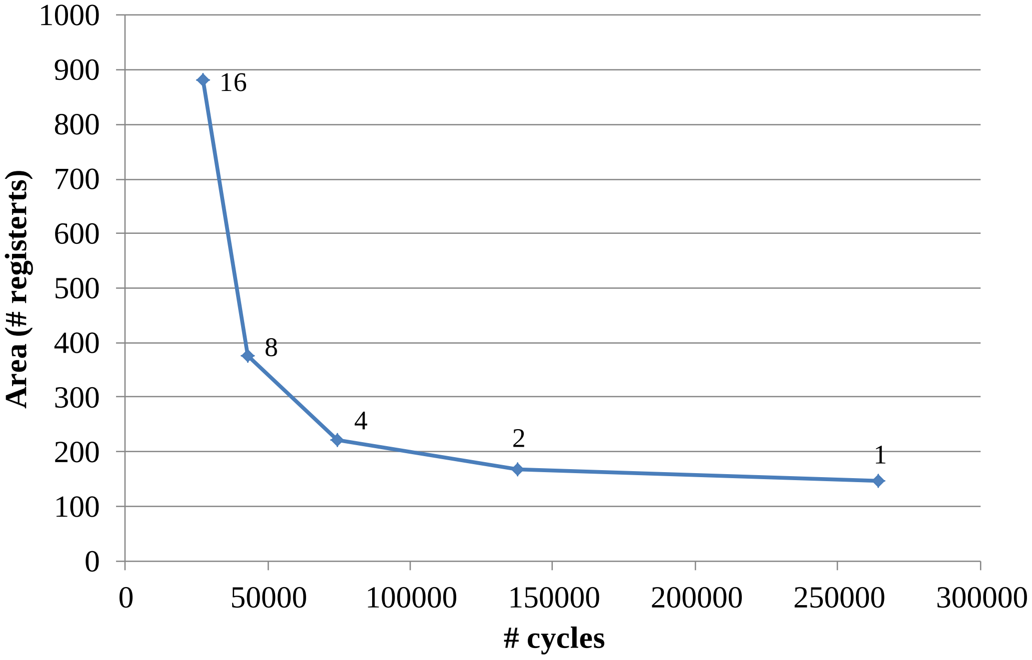


Figure 5.3: Evaluation of parallelism level for the intra prediction architecture.

Based on the results presented in figure 5.3, it is possible to observe that the use of a parallelism level of 16 samples would be the best choice considering the number of clock cycles, however, the area is too large and the memory bandwidth required to feed the architecture with 16 samples per cycle would be large as well. On the other hand, the parallelism level of one sample is the least hardware consuming, but the time expend to process a whole block is very long and based in the number of clock cycles available to allow real time processing it does not allow to achieve the necessary throughput to process high resolution videos. This way, the parallelism level of four samples was used, since it demands a fair memory bandwidth, the hardware resources are not so big compared to the least one and the time to process a whole block is not so far to the best case. Besides, this parallelism level allows real time processing considering the number of clock cycles available.

Figure 5.4 shows a high level block diagram of the designed architecture.

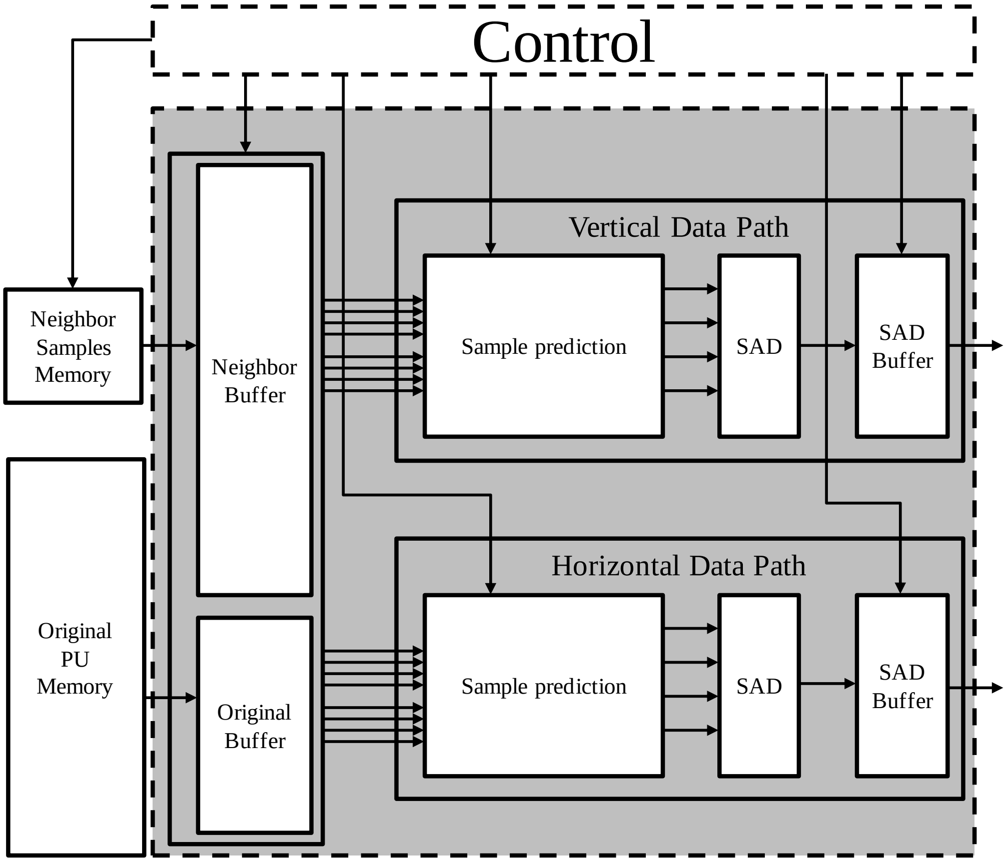


Figure 5.4: High level block diagram of the designed architecture.

## 5.2 Buffers and Internal Memories

In order to locally store all necessary samples for the intra prediction execution, two different on-chip memories were designed: Neighbor Memory and Original Memory. The neighbor memory uses 129 words of eight bits and it can delivers one sample per clock. The original memory has 1024 words of 32 bits each, which is the amount necessary to store a whole 64x64 block. The samples were organized to be aligned in the memory. It means that each read loads 4 samples per clock cycle (properly feeding the architecture parallelism).

Besides the predicted block, the architecture also performs the residual generation, i.e., the difference between the original and the predicted block. In HM 5.1 software the intra prediction process over a PU is performed in two steps. First, one predicted block is generated according to one of the available modes. Then, the residual between the original block and the predicted block is generated. Figure 5.5 shows how the prediction process is performed in the HM software, where: **O** means the original block, **Pn** are the predicted blocks according to the mode **n,** and **Rn** are the residual blocks generated by each mode.

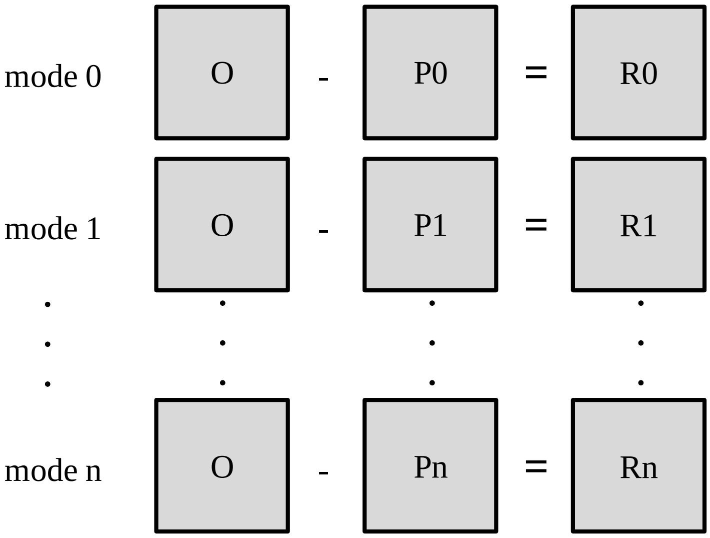


Figure 5.5: Intra prediction process in HM software.

Looking to figure 5.5 it is possible to see that the original block (**O**) is accessed as many times as the number of available modes, increasing the number of redundant memory accesses. In this work, the intra prediction is performed in a different way in order to save memory accesses, since it is a bottleneck to encoder systems.

Initially, all necessary neighbor samples are loaded from the neighbor memory to an internal register bank (Neighbor Buffer in figure 5.4). This register bank is composed by 129 registers of 8 bits, since in the worst case (when the prediction is over 64x64 PUs) there are 129 neighbors to be used by the intra prediction process. The neighbor buffer was used since the data paths need four or five samples (integer or fractional prediction) per clock cycle whereas the memory delivers only one sample per cycle. Finally, four original samples are loaded from the memory of original samples to another buffer (Original Buffer in figure 5.4). When the load step is complete, the prediction of each mode starts.

Based on the prediction process presented in figure 5.5, a different way to use the original samples was designed. Instead of performing one mode prediction using all four original samples at a time and then reload the same original samples again to perform the next mode, all available modes are performed over this four samples. This way, the original samples are loaded only once from the external memory, decreasing the number of memory accesses. Since only four samples (in each above and left data path) are calculated in parallel, another buffer (SAD Buffer) with 17 registers was used for each data path to store the partial SAD values.

The data reuse designed in the intra prediction architecture has provided a huge decrease in the number of memory accesses considering the original samples, since they are fetched from the memory only once for each PU size. Figure 5.6 shows a comparison in terms of memory accesses necessary for all PU sizes and all intra prediction modes for a given 64x64 block using the corner cases of the design in $log_{10}$ scale.

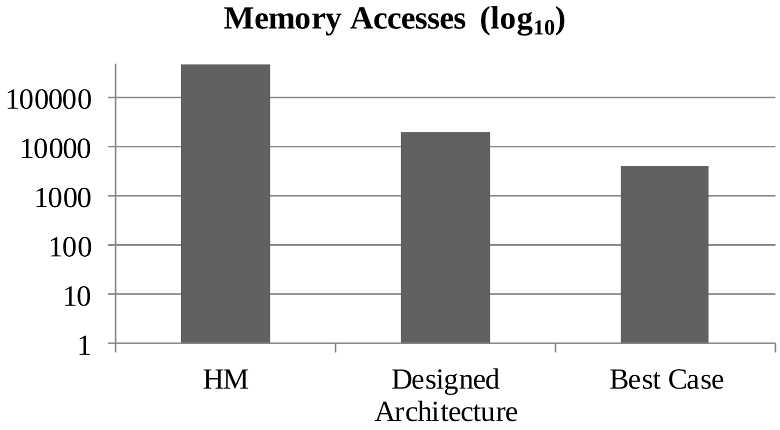


Figure 5.6: Memory accesses comparison.

In the HM software each original sample is accessed for each mode prediction of one PU size whereas the proposed architecture access this original sample only once for each PU size. This way, the designed architecture saves 95% in memory access when compared to the HM software. Besides, comparing with the best case (when the original samples are fetched from the memory only once for a whole 64x64 block intra prediction) the designed architecture uses five times more access. The best case was not considered as target, since it would imply in store all 4096 (64x64) samples in registers, highly increasing the used hardware resources.

## 5.3 Architecture Schedule

Since all samples (neighbor and original) are available after the load step, the pipeline technique was also used to increase the architecture throughput. Besides, it also improves the architecture resources usage, since after the latency of loading the neighbor samples, the architecture will always have valid input data to be processed. Figure 5.7 shows the pipeline schedule diagram of the designed architecture.

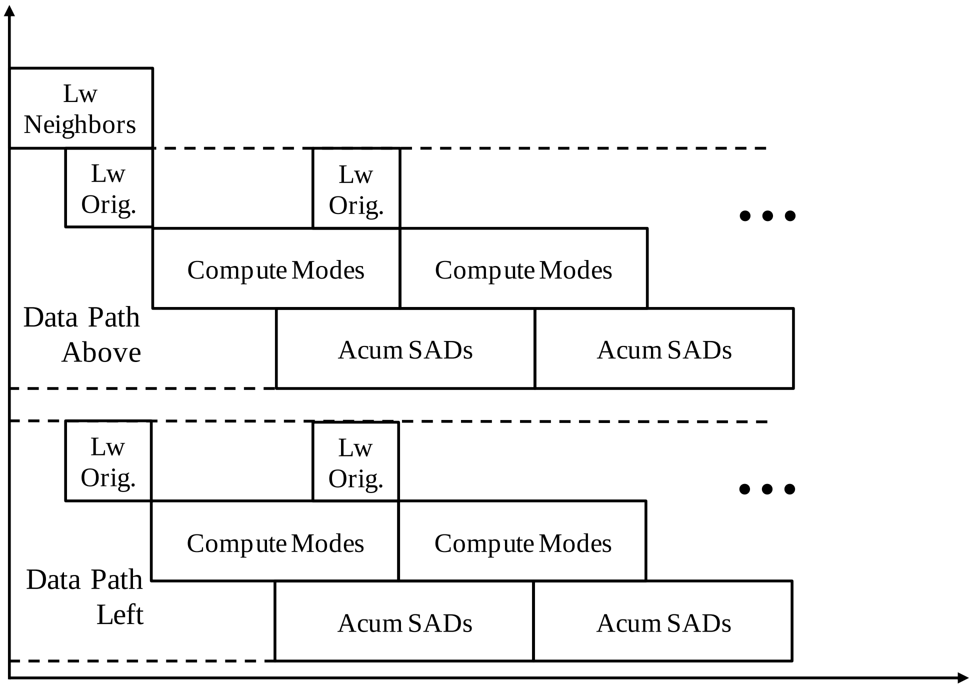


Figure 5.7: Schedule diagram of the designed architecture.

The first step is to load one by one all available neighbor samples from external memory to the neighbor buffer. This step is performed only once for each PU size and the time needed to load these neighbor samples varies (from 9 to 129) according to the target PU size and the available neighbors. These neighbors will be used for both data paths. Then, the loading step of the four original samples for each data path takes only one clock cycle. After that, the computing of all possible modes for a given PU is performed. Finally the partial SAD values are stored in the SAD buffer.

The control unit must deliver to all memories and buffers the addresses and the two constants to perform the sample prediction when the prediction is fractional. These signals are always the same for both data paths, since the intra directional modes are processed according to the regularity among the angles directions previously mentioned.

## 5.4 Complete Decision Reuse Heuristic (CDR) in the Architecture Design

The presented architecture is able to process all PU sizes and all intra prediction modes. Then, it can be used in any design which demands the calculation of all intra prediction modes proposed in the HEVC emerging standard. However, we demonstrated in chapter 4 that it is possible to decrease the number of evaluated intra modes by the full rate-distortion optimization (RDO) technique with little damage to the compression performance. This way, we included the possibility of using the Complete Decision Reuse (CDR) heuristic to improve the hardware architecture performance. As already explained in section 4.1, the CDR heuristic uses all modes from the level below in the quad-tree structure as references, to accelerate the intra prediction process, decreasing the number of clock cycles necessary to perform the prediction over one PU. The hardware architecture modifications necessary to support the CDR fast algorithm were very simple.

Firstly, one buffer with four positions was designed to store the four modes from the level below. These modes will be used for the control unit to decide which modes will be performed. Then, one signal (***fast_prediction***) was added to indicate whether or not the CDR heuristic will be used. Finally, the control unit was modified to support more transitions between the states that perform the intra prediction modes according to the candidates generated by the CDR heuristic.

## 5.5 Synthesis Results

All architecture modules (buffers and data paths) were described in VHDL and synthesized to IBM 65nm standard cell library using the Cadence Encounter RTL Compiler. We found only one work so far in the literature (LI e SHI, 2011) that presents a hardware design for the HEVC intra prediction module. That work proposes a simplified architecture to process the intra prediction in HEVC only for 4x4 PUs whereas our proposed architecture is able to process all PU sizes. It is important to notice that if only 4x4 PUs are used, the compression gains provided by the intra prediction process in the emerging HEVC standard cannot be achieved. Besides, their technology target was TSMC 130nm. This way, it is hard to do a more detailed comparison between both works. Table 5.1 shows the synthesis results for the designed intra prediction architecture considering target technology, number of gates and the maximum operational frequency achieved.

Table 5.1: Synthesis results

| **Work** | **This work** |
|---|---|
| Technology | IBM 65nm |
| Gate Count | 36,734 |
| Max Frequency (MHz) | 500 |
| PU size supported | All |

The designed architecture can work at a high operation frequency (500 MHz). This high operation frequency was achieved since the critical path of the architecture is only one adder of 29 bits. Our work used four times more gates than the work (LI e SHI, 2011). However, our architecture supports all 64x64, 32x32, 16x16, 8x8 and 4x4 PU sizes.

The external memories results were generated using the HP Cacti 5.3 memory model tool targeting 65nm technology. The results are presented in table 5.2 considering memory sizes, number of ports and word width, read power and time access. We could not compare these results with the related work (LI e SHI, 2011), since they did not provide such results in the paper.

Table 5.2: Memory results.

| **Memory** | **Neighbor** | **Original** |
|---|---|---|
| Size | 129B | 4KB |
| Ports | 1 | 2 |
| Word Width | 8 | 32 |
| Read Power | 1.2mW | 9.3mW |
| Time Access | 0.92 ns | 1.2 ns |

The neighbor memory size is 129 bytes since, in the worst, 129 neighbor samples with 8 bits are necessary to the intra prediction of a 64x64 block (64 up, 64 left and 1 diagonal). The original PU memory size is 4 KB since it needs to store a whole 64x64 PU. The original memory is organized in words of 32 bits. Each 32 bits word stores four original samples (eight bits per sample) that are read in one clock cycle to feed the intra prediction architecture. The original memory has 2 read ports, one for each vertical data path and horizontal data path respectively.

## 5.6 Performance evaluation of CDR based architecture

The intra prediction hardware architecture was designed to support real time processing for high resolution videos, such as HD1080p. In order to achieve such goal it is necessary to process 15,300 64x64 quad-trees per second. This performance constraint depends on the number of clock cycles used to process each quad-tree and the operational frequency achieved by the architecture synthesis. This way, we performed an evaluation of the proposed hardware architecture to check if the performance requirements were achieved.

Table 5.3 shows the number of clock cycles that the intra prediction architecture takes to process each PU size comparing the use of the CDR based architecture with no fast mode decision at all, i.e., the regular HM 5.1 based decision processing all modes for all PU sizes. The CDR based results consider the worst case in number of modes to be processed, since there will be cases when all reference modes will be the same, and then, the CDR heuristic will consider only one mode to be processed.

Table 5.3: Number of clock cycles necessary to process each PU size.

| PU Size | Regular based | CDR based |
|---|---|---|
| 64x64 | 1,162 | 1,162 |
| 32x32 | 4,490 | 650 |
| 16x16 | 1,162 | 202 |
| 8x8 | 314 | 74 |
| 4x4 | 62 | 62 |
| one quad-tree | 73,682 | 27,602 |

Considering the results in table 5.3 it is possible to notice that the number of clock cycles necessary to process one entire tree block is much lower when the CDR based architecture is considered. The result considering the 4x4 PU size is the same since it will generate the reference modes that will be used by the CDR heuristic in the above levels in the quad tree structure. The 64x64 result is also the same since the number of directional modes is only two and this number is never reduced by the CDR heuristic in the quad tree level.

Based on the results presented in table 5.3, it is possible to calculate the exactly operational frequency necessary to the architecture process each video resolution in real time. Table 5.4 shows the architectural performance for some video resolutions comparing again the designed CDR based architecture with a regular based one. The comparison is performed considering the throughput of each design in terms of the number of tree blocks per second and frames per second running at 500 MHz (maximum operational frequency achieved by the synthesis), besides the minimum frequency necessary to run each video resolution at 30 frames per second is also presented.

Table 5.4: Architectural performance for some video resolutions.

| Video Resolutions | Regular based architecture | | | CDR based architecture | | |
|---|---|---|---|---|---|---|
| | treeblocks/s (500 MHz) | frames/s (500 MHz) | Min. Freq. 30 frames/s (MHz) | treeblocks/s (500 MHz) | frames/s (500 MHz) | Min. Freq. 30 frames/s (MHz) |
| 2Kx1K (2048x1024) | 6,785 | 13 | 1,131.75 | 18,115 | 35 | 423.9 |
| HD1080p (1920x1088) | | 13 | 1,127.33 | | 35 | 422.3 |
| HD720p (1280x720) | | 30 | 497.35 | | 80 | 186.3 |
| VGA (640x480) | | 90 | 165.78 | | 241 | 62.1 |

Table 5.4 shows that the designed architecture achieved the proposed requirement to process high resolution videos in real time. Another important result is that the use of CDR fast mode decision algorithm as base for the architecture design highly increases the architecture throughput. The CDR based architecture can process almost three times more tree blocks when compared to the regular based one. Besides, the CDR based architecture allows the processing of both HD1080p and 2Kx1K video resolution at 30 frames per second, which is not possible when the regular based architecture is considered. The CDR based architecture also allows the use lower operational frequency than the regular based one, which is very useful when power constraints are considered.

# 6 CONCLUSIONS

This master thesis presented a dedicated hardware architecture for the intra-frame prediction process of the emerging HEVC standard. The main point of this work was to design a dedicated architecture for the intra prediction based on a fast mode decision algorithm to improve the intra prediction performance in HEVC encoders.

Initially, an investigation was performed targeting the intra prediction module in HEVC encoders, focusing on the mode decision process. Based on this investigation three fast mode decision heuristics were developed according to a new proposed evaluation order in the coding structure used in the HEVC. The algorithms were developed and evaluated in the HM 5.1 model software considering compression efficiency and complexity reduction. After that, an intra-frame prediction hardware architecture was designed using the previously developed Complete Decision Reuse (CDR) heuristic.

The results presented in this work have shown that is important to investigate the use fast intra mode decision algorithms to increase the performance in any intra prediction engine. In this work, the use of the CDR fast algorithm has provided an increase of three times in the processing rates. This throughput increase was obtained with a small degradation of 2.71% in the bit rate and of 0.08 in quality (considering the worst case), when compared with the reference solution. Besides, the designed architecture using the CDR fast algorithm has been able to process high resolution videos, such as HD1080p and 2Kx1K in real time.

# REFERENCES

BHARANITHARAN, K. et al. A Low Complexity Detection of Discrete Cross Differences for Fast H.264/AVC Intra Prediction. **IEEE Transactions on Multimedia**, v. 10, p. 1250-1260, Novembro 2008.

DINIZ, C. et al. **A Real Time H.264/AVC Intra frame Prediction Hardware Architecture for HDTV 1080p**. IEEE INTERNATIONAL CONFERENCE ON MULTIMEDIA AND EXPO. New York, USA [S.l.]: [s.n.]. 2009. p. 1138-1141.

GHANBARI, M. Standard Codecs: Image Compression to Advanced Video Coding. **IEEE Press**, 2003.

HUANG, Y. et al. Analysis, Fast Algorithm, and VLSI Architecture Design for H.264/AVC Intra Frame Coder. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 15, p. 378-401, Março 2005.

HUANG, Y.; OU, T.; CHEN, H. Fast Decision of Block Size, Prediction Mode, and Intra Block for H.264 Intra Prediction. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 20, p. 1122-1132, Agosto 2010.

ISO/IEC-JCT1/SC29/WG11. **Common test conditions and software reference**. Daegu, Korea. 2011.

ISO/IEC-JTC1/SC29/WG11. **High Efficiency Video Coding (HEVC) Test Model 3 Encoder Description**. [S.l.]. 2011.

JCT-VC. High Efficiency Video Coding, 2012. Disponivel em: <http://hevc.hhi.fraunhofer.de/>. Acesso em: Outubro 2012.

JCT-VC. **High Efficiency Video Coding (HEVC) text specification draft 8**. Stockholm. 2012.

JM. H.264/AVC Software Coordination, 2012. Disponivel em: <http://iphome.hhi.de/suehring/tml/>. Acesso em: Outubro 2012.

JVT. **Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264/AVC or ISO/IEC 14496-10 AVC)**. [S.l.]. 2003.

LI, F.; SHI, G. **An Efficient VLSI Architecture for 4x4 Intra Prediction in the HighEfficiency Video Coding (HEVC) Standard**. IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING. Brussels, Belguim [S.l.]: [s.n.]. 2011. p. 381-384.

MIN, J. H. **Unification of the directional intra prediction**. [S.l.]. 2010.

PALOMINO, D. et al. **A Memory Aware and Multiplierless VLSI Architecture for the Complete Intra Prediction of the HEVC Emerging Standard**. IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING. Orlando, USA, [s.n.]. 2012.

PARLAK, M.; ADIBELLI, Y.; HAMZAOGLU, I. A Novel Computational Complexity and Power Reduction Technique for H.264 Intra Prediction. **IEEE Transactions on Consumer Electronics**, v. 54, p. 2006-2014, Novembro 2008.

RICHARDSON, Y. **H.264/AVC and MPEG-4 Video Compression - Video Coding for Next-Generation Multimedia**. [S.l.]: [s.n.], 2003.

SULLIVAN, G.; WIEGAND, T. Rate-distortion optimization for Video Compression. **IEEE Signal Processing Magazine**, v. 15, p. 74-90, 1998.

SUNNA, P. **AVC/H.264 - an advanced video coding system for SD and HD broadcasting**. Ebu Technical Review. [S.l.]: [s.n.]. 2005.

WANG, J.-C. et al. A Fast Mode Decision Algorithm and Its VLSI Design for H.264/AVC Intra-Prediction. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 17, p. 1414-1422, Outubro 2007.

WIEGAND, T. Overview of the H.264/AVC Video Coding Standard. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 13, p. 560-576, July 2003.

ZHAO, L. et al. **Fast Mode Decision Algorithm for Intra Prediction in HEVC**. IEEE VISUAL COMMUNICATIONS AND IMAGE PROCESSING. Tainan City, Taiwan. 2011.

# ANEX – PUBLICATIONS

One paper related to this master thesis work is presented in this annex. The paper entitled *A Memory Aware and Multiplierless VLSI Architecture for the Complete Intra Prediction of the HEVC Emerging Standard* was published in the annual IEEE International Conference on Image Processing (ICIP) 2012.

Other papers published during the master course period:

- *A High Throughput H.264/AVC Intra-Frame Encoding Loop Architecture for HD1080p* in the annual IEEE International Symposium on Circuits and Systems, 2011.

- *SHBS: A Heuristic for Fast Inter Mode Decision of H.264/AVC Standard Targeting VLSI Design* in the annual IEEE International Conference on Multimedia and Expo, 2011.

- *Algorithm and Hardware Design of a Fast Intra-Frame Mode Decision Module for H.264/AVC Encoders* in the annual Symposium on Integrated Circuits and Systems Design, 2011.

- *Low Complexity Hierarchical Mode Decision Algorithms Targeting VLSI Architecture Design for the H.264/AVC Video Encoder* in the Hindwai VLSI Design Journal, 2012.

- *Algorithm and Hardware Design of a Fast Intra Frame Mode Decision Module for H.264/AVC Encoders* in the Hindwai International Journal of Reconfigurable Computing (IJRC), 2012.

- *FPGA Based Design for Motion Vector Predicton in H.264/AVC Encoders Targeting HD1080p Resolution* in the annual Southern Programmable Logic Conference, 2012.

# APENDIX A – DETAILED RESULTS TABLES

Table 1: Results HM regular mode decision on High Efficiency condition.

| Class | Video | QPISlice | kbps | PSNR | Time (sec) |
|---|---|---|---|---|---|
| Class A<br>4K | Traffic | 22 | 104188,30 | 43,57 | 7159,21 |
| | | 27 | 58045,88 | 40,41 | 6696,14 |
| | | 32 | 32927,82 | 37,39 | 6285,01 |
| | | 37 | 18564,26 | 34,34 | 5960,01 |
| | People on Street | 22 | 106686,39 | 43,48 | 7242,97 |
| | | 27 | 61760,44 | 40,07 | 6750,07 |
| | | 32 | 35115,85 | 37,00 | 6357,82 |
| | | 37 | 20391,03 | 34,16 | 6084,62 |
| Class B<br>1080p | Kimono | 22 | 18170,75 | 43,09 | 3389,55 |
| | | 27 | 10288,17 | 41,79 | 3128,66 |
| | | 32 | 6158,42 | 40,00 | 3014,57 |
| | | 37 | 3655,96 | 37,61 | 2941,15 |
| | ParkScene | 22 | 53563,11 | 41,80 | 3732,77 |
| | | 27 | 28914,19 | 38,78 | 3356,21 |
| | | 32 | 15054,17 | 35,78 | 3111,48 |
| | | 37 | 7433,26 | 32,91 | 2909,23 |
| | Cactus | 22 | 112629,97 | 40,71 | 3851,78 |
| | | 27 | 51813,75 | 37,97 | 3365,80 |
| | | 32 | 28117,13 | 35,73 | 3123,28 |
| | | 37 | 15224,43 | 33,26 | 2925,67 |
| | BasketballDrive | 22 | 64380,88 | 41,27 | 3303,12 |
| | | 27 | 26068,71 | 38,96 | 3035,28 |
| | | 32 | 13554,08 | 37,35 | 2931,18 |
| | | 37 | 7592,11 | 35,49 | 2785,71 |
| | BQTerrace | 22 | 177950,52 | 42,77 | 3587,82 |
| | | 27 | 90472,03 | 37,84 | 3415,03 |
| | | 32 | 48789,21 | 34,82 | 3174,07 |
| | | 37 | 26863,78 | 32,00 | 2954,42 |
| Class C<br>WVGA | BasketballDrill | 22 | 20688,24 | 41,90 | 732,25 |
| | | 27 | 11030,84 | 38,54 | 665,68 |
| | | 32 | 5863,11 | 35,60 | 611,76 |
| | | 37 | 3259,02 | 33,03 | 576,26 |
| | BQMall | 22 | 27750,12 | 41,66 | 677,39 |
| | | 27 | 16793,07 | 38,65 | 643,73 |
| | | 32 | 9968,36 | 35,46 | 613,22 |
| | | 37 | 5607,58 | 32,22 | 583,68 |
| | PartyScene | 22 | 48383,56 | 41,20 | 768,98 |
| | | 27 | 30497,79 | 36,86 | 758,78 |
| | | 32 | 18369,96 | 32,93 | 716,97 |
| | | 37 | 10131,36 | 29,17 | 660,10 |
| | RaceHorses | 22 | 17008,54 | 42,52 | 666,66 |

| | | | | | |
|---|---|---|---|---|---|
| | | 27 | 10517,18 | 38,89 | 616,65 |
| | | 32 | 6146,89 | 35,15 | 579,19 |
| | | 37 | 3084,36 | 31,32 | 539,78 |
| Class D WQVGA | BasketballPass | 22 | 4921,28 | 43,24 | 177,31 |
| | | 27 | 2897,17 | 39,69 | 153,58 |
| | | 32 | 1619,04 | 36,27 | 146,68 |
| | | 37 | 882,94 | 33,12 | 137,90 |
| | BQSquare | 22 | 13272,07 | 41,54 | 192,63 |
| | | 27 | 8712,89 | 37,14 | 181,33 |
| | | 32 | 5583,81 | 33,38 | 175,72 |
| | | 37 | 3449,85 | 29,75 | 161,95 |
| | BlowingBubbles | 22 | 9615,24 | 41,19 | 186,81 |
| | | 27 | 5654,04 | 37,14 | 179,50 |
| | | 32 | 3086,30 | 33,48 | 165,75 |
| | | 37 | 1575,99 | 30,24 | 150,37 |
| | RaceHorses | 22 | 4853,17 | 42,64 | 172,99 |
| | | 27 | 2967,39 | 38,53 | 159,98 |
| | | 32 | 1658,78 | 34,54 | 147,89 |
| | | 37 | 835,87 | 31,05 | 135,98 |
| Class E 720p | Vidyo1 | 22 | 21089,45 | 44,79 | 1421,39 |
| | | 27 | 12279,65 | 42,48 | 1356,45 |
| | | 32 | 7344,83 | 39,88 | 1313,70 |
| | | 37 | 4409,68 | 36,98 | 1272,44 |
| | Vidyo3 | 22 | 22426,52 | 44,67 | 1409,40 |
| | | 27 | 13635,36 | 42,34 | 1318,14 |
| | | 32 | 8458,15 | 39,65 | 1280,06 |
| | | 37 | 5117,64 | 36,57 | 1248,83 |
| | Vidyo4 | 22 | 23943,38 | 44,35 | 1408,93 |
| | | 27 | 13282,04 | 41,73 | 1321,78 |
| | | 32 | 7497,69 | 39,12 | 1276,15 |
| | | 37 | 4211,18 | 36,46 | 1245,93 |
| Class F | BasketballDrillText | 22 | 21098,56 | 42,17 | 703,09 |
| | | 27 | 11690,72 | 38,77 | 652,25 |
| | | 32 | 6466,42 | 35,76 | 612,58 |
| | | 37 | 3722,38 | 33,02 | 578,01 |
| | ChinaSpeed | 22 | 24320,47 | 44,95 | 1259,63 |
| | | 27 | 16506,08 | 41,01 | 1188,30 |
| | | 32 | 10886,99 | 37,21 | 1125,66 |
| | | 37 | 7064,64 | 33,55 | 1062,85 |
| | SlideEditing | 22 | 39049,84 | 46,21 | 1606,00 |
| | | 27 | 29243,11 | 41,87 | 1557,90 |
| | | 32 | 21894,33 | 37,25 | 1518,98 |
| | | 37 | 15682,95 | 32,44 | 1447,17 |
| | SlideShow | 22 | 4589,65 | 51,21 | 1116,85 |
| | | 27 | 2983,33 | 47,67 | 1101,50 |
| | | 32 | 1953,45 | 44,02 | 1081,24 |
| | | 37 | 1270,40 | 40,34 | 1058,50 |

Table 2: Results for the FDR heuristic on High Efficiency condition (target level 8).

| Class | Video | QPISlice | kbps | PSNR | Time (sec) |
|---|---|---|---|---|---|
| Class A<br>4K | Traffic | 22 | 105139,36 | 43,52 | 6532,66 |
| | | 27 | 58795,00 | 40,35 | 6082,20 |
| | | 32 | 33464,72 | 37,31 | 5740,53 |
| | | 37 | 18865,68 | 34,24 | 5451,77 |
| | People on Street | 22 | 108090,23 | 43,46 | 6543,71 |
| | | 27 | 62939,77 | 40,03 | 6135,67 |
| | | 32 | 36215,21 | 36,93 | 5812,73 |
| | | 37 | 21217,21 | 34,06 | 5547,49 |
| Class B<br>1080p | Kimono | 22 | 18191,77 | 43,08 | 3066,07 |
| | | 27 | 10316,30 | 41,78 | 2853,20 |
| | | 32 | 6180,70 | 39,98 | 2718,43 |
| | | 37 | 3671,46 | 37,60 | 2667,03 |
| | ParkScene | 22 | 53828,39 | 41,76 | 3343,86 |
| | | 27 | 29060,80 | 38,73 | 3038,51 |
| | | 32 | 15103,78 | 35,72 | 2825,42 |
| | | 37 | 7450,10 | 32,86 | 2659,04 |
| | Cactus | 22 | 113592,18 | 40,69 | 3419,85 |
| | | 27 | 52446,90 | 37,94 | 3044,76 |
| | | 32 | 28608,24 | 35,68 | 2827,99 |
| | | 37 | 15583,83 | 33,20 | 2701,15 |
| | BasketballDrive | 22 | 63688,04 | 41,16 | 2915,00 |
| | | 27 | 26390,05 | 38,94 | 2735,38 |
| | | 32 | 13784,57 | 37,33 | 2642,07 |
| | | 37 | 7736,05 | 35,45 | 2548,93 |
| | BQTerrace | 22 | 179306,88 | 42,53 | 3214,56 |
| | | 27 | 91110,43 | 37,76 | 3068,10 |
| | | 32 | 49674,43 | 34,77 | 2863,32 |
| | | 37 | 27482,19 | 31,93 | 2713,58 |
| Class C<br>WVGA | BasketballDrill | 22 | 21113,25 | 41,88 | 647,41 |
| | | 27 | 11318,96 | 38,50 | 600,71 |
| | | 32 | 6031,66 | 35,55 | 554,90 |
| | | 37 | 3353,48 | 32,96 | 528,33 |
| | BQMall | 22 | 28074,36 | 41,64 | 600,67 |
| | | 27 | 17054,45 | 38,61 | 581,30 |
| | | 32 | 10174,24 | 35,39 | 556,90 |
| | | 37 | 5747,54 | 32,13 | 529,63 |
| | PartyScene | 22 | 48736,65 | 41,20 | 686,46 |
| | | 27 | 30808,98 | 36,85 | 681,37 |
| | | 32 | 18648,84 | 32,90 | 647,42 |
| | | 37 | 10360,64 | 29,10 | 596,35 |
| | RaceHorses | 22 | 17165,80 | 42,50 | 599,93 |
| | | 27 | 10636,10 | 38,86 | 561,84 |
| | | 32 | 6229,45 | 35,08 | 525,41 |
| | | 37 | 3137,99 | 31,25 | 496,01 |

| | | | | | |
|---|---|---|---|---|---|
| Class D WQVGA | BasketballPass | 22 | 4972,99 | 43,20 | 144,71 |
| | | 27 | 2934,53 | 39,65 | 137,94 |
| | | 32 | 1653,72 | 36,21 | 133,81 |
| | | 37 | 910,70 | 33,05 | 125,00 |
| | BQSquare | 22 | 13365,04 | 41,52 | 167,67 |
| | | 27 | 8790,11 | 37,12 | 161,67 |
| | | 32 | 5662,25 | 33,36 | 155,71 |
| | | 37 | 3526,36 | 29,73 | 147,48 |
| | BlowingBubbles | 22 | 9704,79 | 41,18 | 163,99 |
| | | 27 | 5727,55 | 37,12 | 161,62 |
| | | 32 | 3141,40 | 33,42 | 148,62 |
| | | 37 | 1608,85 | 30,15 | 136,84 |
| | RaceHorses | 22 | 4917,48 | 42,63 | 155,19 |
| | | 27 | 3013,85 | 38,48 | 144,77 |
| | | 32 | 1694,15 | 34,47 | 134,63 |
| | | 37 | 860,33 | 30,97 | 124,09 |
| Class E 720p | Vidyo1 | 22 | 21437,49 | 44,75 | 1290,84 |
| | | 27 | 12523,99 | 42,42 | 1237,62 |
| | | 32 | 7522,47 | 39,79 | 1194,66 |
| | | 37 | 4523,56 | 36,87 | 1169,48 |
| | Vidyo3 | 22 | 22777,44 | 44,64 | 1270,09 |
| | | 27 | 13935,57 | 42,29 | 1206,90 |
| | | 32 | 8700,56 | 39,57 | 1165,25 |
| | | 37 | 5276,13 | 36,44 | 1143,76 |
| | Vidyo4 | 22 | 24167,06 | 44,32 | 1276,18 |
| | | 27 | 13489,74 | 41,69 | 1205,16 |
| | | 32 | 7652,49 | 39,06 | 1165,27 |
| | | 37 | 4311,22 | 36,38 | 1148,40 |
| Class F | BasketballDrillText | 22 | 21488,54 | 42,15 | 631,69 |
| | | 27 | 11968,25 | 38,74 | 591,01 |
| | | 32 | 6636,76 | 35,70 | 556,50 |
| | | 37 | 3827,93 | 32,95 | 530,09 |
| | ChinaSpeed | 22 | 24541,77 | 44,96 | 1123,65 |
| | | 27 | 16683,49 | 40,97 | 1063,57 |
| | | 32 | 11034,73 | 37,16 | 1003,07 |
| | | 37 | 7180,56 | 33,48 | 965,39 |
| | SlideEditing | 22 | 39237,70 | 46,23 | 1425,25 |
| | | 27 | 29424,06 | 41,83 | 1396,59 |
| | | 32 | 22058,52 | 37,22 | 1355,68 |
| | | 37 | 15856,77 | 32,42 | 1303,86 |
| | SlideShow | 22 | 4659,12 | 51,22 | 994,69 |
| | | 27 | 3039,33 | 47,65 | 995,74 |
| | | 32 | 1997,96 | 43,98 | 980,20 |
| | | 37 | 1307,29 | 40,25 | 954,95 |

Table 3: Results for the FDR heuristic on High Efficiency condition (target level 16).

| Class | Video | QPISlice | kbps | PSNR | Time (sec) |
|---|---|---|---|---|---|
| Class A<br>4K | Traffic | 22 | 106883,39 | 43,51 | 6234,87 |
| | | 27 | 60172,23 | 40,32 | 5795,01 |
| | | 32 | 34531,57 | 37,26 | 5492,50 |
| | | 37 | 19578,54 | 34,17 | 5266,42 |
| | People on Street | 22 | 110102,26 | 43,44 | 6176,17 |
| | | 27 | 64731,53 | 40,02 | 5844,06 |
| | | 32 | 37717,78 | 36,91 | 5519,03 |
| | | 37 | 22529,12 | 34,01 | 5327,69 |
| Class B<br>1080p | Kimono | 22 | 18310,74 | 43,07 | 2916,70 |
| | | 27 | 10414,11 | 41,77 | 2713,47 |
| | | 32 | 6245,14 | 39,96 | 2611,31 |
| | | 37 | 3707,33 | 37,56 | 2565,33 |
| | ParkScene | 22 | 54359,57 | 41,76 | 3175,88 |
| | | 27 | 29401,34 | 38,72 | 2882,87 |
| | | 32 | 15299,42 | 35,70 | 2702,68 |
| | | 37 | 7538,38 | 32,82 | 2544,20 |
| | Cactus | 22 | 115213,52 | 40,70 | 3242,95 |
| | | 27 | 53520,61 | 37,93 | 2883,90 |
| | | 32 | 29383,81 | 35,65 | 2700,68 |
| | | 37 | 16138,75 | 33,15 | 2587,06 |
| | BasketballDrive | 22 | 64669,30 | 41,15 | 2757,12 |
| | | 27 | 27140,98 | 38,93 | 2615,83 |
| | | 32 | 14316,66 | 37,31 | 2542,42 |
| | | 37 | 8102,29 | 35,41 | 2441,65 |
| | BQTerrace | 22 | 180950,24 | 42,52 | 3010,47 |
| | | 27 | 92479,96 | 37,76 | 2896,45 |
| | | 32 | 50855,52 | 34,76 | 2714,11 |
| | | 37 | 28454,22 | 31,90 | 2560,60 |
| Class C<br>WVGA | BasketballDrill | 22 | 21495,14 | 41,88 | 608,73 |
| | | 27 | 11672,85 | 38,50 | 563,96 |
| | | 32 | 6296,33 | 35,54 | 534,81 |
| | | 37 | 3541,39 | 32,92 | 507,01 |
| | BQMall | 22 | 28401,43 | 41,63 | 564,71 |
| | | 27 | 17338,29 | 38,60 | 547,69 |
| | | 32 | 10439,56 | 35,37 | 528,13 |
| | | 37 | 5966,37 | 32,07 | 505,05 |
| | PartyScene | 22 | 48885,74 | 41,21 | 638,60 |
| | | 27 | 30966,48 | 36,86 | 642,92 |
| | | 32 | 18810,09 | 32,90 | 608,30 |
| | | 37 | 10552,44 | 29,09 | 566,62 |
| | RaceHorses | 22 | 17373,14 | 42,51 | 568,75 |
| | | 27 | 10806,93 | 38,85 | 530,33 |
| | | 32 | 6379,24 | 35,07 | 501,60 |

| | | | | | |
|---|---|---|---|---|---|
| | | 37 | 3258,72 | 31,23 | 472,66 |
| Class D WQVGA | BasketballPass | 22 | 5042,59 | 43,19 | 136,47 |
| | | 27 | 3000,79 | 39,65 | 130,58 |
| | | 32 | 1702,00 | 36,20 | 127,23 |
| | | 37 | 950,60 | 33,02 | 118,77 |
| | BQSquare | 22 | 13405,02 | 41,52 | 158,01 |
| | | 27 | 8836,90 | 37,12 | 151,75 |
| | | 32 | 5703,41 | 33,37 | 146,98 |
| | | 37 | 3568,11 | 29,74 | 138,57 |
| | BlowingBubbles | 22 | 9763,46 | 41,20 | 154,11 |
| | | 27 | 5782,23 | 37,13 | 152,85 |
| | | 32 | 3204,29 | 33,42 | 140,32 |
| | | 37 | 1663,35 | 30,13 | 130,71 |
| | RaceHorses | 22 | 4969,82 | 42,64 | 146,58 |
| | | 27 | 3060,54 | 38,50 | 137,22 |
| | | 32 | 1735,81 | 34,47 | 127,38 |
| | | 37 | 898,42 | 30,96 | 117,88 |
| Class E 720p | Vidyo1 | 22 | 22207,52 | 44,75 | 1232,09 |
| | | 27 | 13096,02 | 42,39 | 1180,41 |
| | | 32 | 7963,04 | 39,74 | 1145,51 |
| | | 37 | 4822,29 | 36,79 | 1137,21 |
| | Vidyo3 | 22 | 23237,18 | 44,62 | 1205,02 |
| | | 27 | 14292,47 | 42,25 | 1146,89 |
| | | 32 | 8998,36 | 39,51 | 1117,02 |
| | | 37 | 5509,96 | 36,37 | 1096,39 |
| | Vidyo4 | 22 | 24736,26 | 44,30 | 1214,67 |
| | | 27 | 13914,03 | 41,65 | 1157,56 |
| | | 32 | 7965,81 | 38,99 | 1116,64 |
| | | 37 | 4561,45 | 36,30 | 1108,00 |
| Class F | BasketballDrillText | 22 | 21818,62 | 42,13 | 597,12 |
| | | 27 | 12279,90 | 38,73 | 558,29 |
| | | 32 | 6886,93 | 35,69 | 531,93 |
| | | 37 | 4011,85 | 32,90 | 505,04 |
| | ChinaSpeed | 22 | 24718,20 | 44,94 | 1051,10 |
| | | 27 | 16863,43 | 40,97 | 995,89 |
| | | 32 | 11200,55 | 37,15 | 942,12 |
| | | 37 | 7321,66 | 33,47 | 903,11 |
| | SlideEditing | 22 | 39345,90 | 46,17 | 1340,42 |
| | | 27 | 29503,22 | 41,78 | 1321,30 |
| | | 32 | 22163,38 | 37,19 | 1278,83 |
| | | 37 | 15940,57 | 32,38 | 1227,79 |
| | SlideShow | 22 | 4747,40 | 51,20 | 944,76 |
| | | 27 | 3114,50 | 47,63 | 936,44 |
| | | 32 | 2057,09 | 43,95 | 922,03 |
| | | 37 | 1360,92 | 40,23 | 909,56 |

Table 4: Results for the FDR heuristic on High Efficiency condition (target level 32).

| Class | Video | QPISlice | kbps | PSNR | Time (sec) |
|---|---|---|---|---|---|
| Class A<br>4K | Traffic | 22 | 107977,74 | 43,52 | 5890,99 |
| | | 27 | 61042,20 | 40,32 | 5503,23 |
| | | 32 | 35248,83 | 37,26 | 5218,65 |
| | | 37 | 20194,79 | 34,15 | 5033,28 |
| | People on Street | 22 | 111169,08 | 43,45 | 5846,70 |
| | | 27 | 65387,62 | 40,02 | 5552,76 |
| | | 32 | 38315,07 | 36,91 | 5226,64 |
| | | 37 | 23207,65 | 34,01 | 5058,96 |
| Class B<br>1080p | Kimono | 22 | 18471,585 | 43,058 | 2773,886 |
| | | 27 | 10524,467 | 41,751 | 2590,051 |
| | | 32 | 6315,2064 | 39,929 | 2480,99 |
| | | 37 | 3760,7942 | 37,526 | 2446,963 |
| | ParkScene | 22 | 54730,428 | 41,776 | 2995,906 |
| | | 27 | 29637,395 | 38,726 | 2725,89 |
| | | 32 | 15450,787 | 35,701 | 2558,546 |
| | | 37 | 7634,016 | 32,816 | 2441,87 |
| | Cactus | 22 | 116416,08 | 40,72 | 3062,581 |
| | | 27 | 54220,096 | 37,936 | 2734,599 |
| | | 32 | 29905,872 | 35,652 | 2572,494 |
| | | 37 | 16587,648 | 33,15 | 2474,443 |
| | BasketballDrive | 22 | 65544,252 | 41,149 | 2595,279 |
| | | 27 | 27740,312 | 38,923 | 2467,693 |
| | | 32 | 14762,28 | 37,293 | 2412,216 |
| | | 37 | 8462,82 | 35,392 | 2325,664 |
| | BQTerrace | 22 | 181583,68 | 42,488 | 2841,072 |
| | | 27 | 93136,205 | 37,766 | 2731,018 |
| | | 32 | 51438,902 | 34,767 | 2563,896 |
| | | 37 | 29013,134 | 31,908 | 2426,626 |
| Class C<br>WVGA | BasketballDrill | 22 | 21569,412 | 41,877 | 573,535 |
| | | 27 | 11746,98 | 38,471 | 534,233 |
| | | 32 | 6433,18 | 35,51 | 500,571 |
| | | 37 | 3677,464 | 32,901 | 483,279 |
| | BQMall | 22 | 28539,269 | 41,634 | 527,363 |
| | | 27 | 17445,355 | 38,591 | 513,207 |
| | | 32 | 10552,517 | 35,364 | 497,501 |
| | | 37 | 6089,5584 | 32,07 | 479,466 |
| | PartyScene | 22 | 48905,108 | 41,215 | 592,917 |
| | | 27 | 30980,628 | 36,855 | 598,668 |
| | | 32 | 18833,392 | 32,898 | 574,105 |
| | | 37 | 10594,88 | 29,098 | 535,515 |
| | RaceHorses | 22 | 17479,406 | 42,531 | 537,904 |

| | | | | | |
|---|---|---|---|---|---|
| | | 27 | 10890,672 | 38,87 | 501,815 |
| | | 32 | 6448,8888 | 35,089 | 472,723 |
| | | 37 | 3318,8472 | 31,253 | 445,895 |
| Class D WQVGA | BasketballPass | 22 | 5081,236 | 43,169 | 127,432 |
| | | 27 | 3032,648 | 39,628 | 123,927 |
| | | 32 | 1733,724 | 36,191 | 120,023 |
| | | 37 | 971,02 | 33,009 | 113,545 |
| | BQSquare | 22 | 13414,45 | 41,526 | 146,663 |
| | | 27 | 8846,8752 | 37,124 | 140,388 |
| | | 32 | 5705,4048 | 33,369 | 137,791 |
| | | 37 | 3573,5184 | 29,742 | 130,161 |
| | BlowingBubbles | 22 | 9769,58 | 41,203 | 144,224 |
| | | 27 | 5792,284 | 37,13 | 143,296 |
| | | 32 | 3212,788 | 33,422 | 132,817 |
| | | 37 | 1680,892 | 30,141 | 122,533 |
| | RaceHorses | 22 | 4981,8984 | 42,652 | 138,396 |
| | | 27 | 3072,5064 | 38,508 | 129,415 |
| | | 32 | 1742,376 | 34,475 | 120,389 |
| | | 37 | 907,3992 | 30,97 | 111,858 |
| Class E 720p | Vidyo1 | 22 | 22577,496 | 44,74 | 1169,262 |
| | | 27 | 13394,573 | 42,374 | 1132,551 |
| | | 32 | 8210,88 | 39,718 | 1091,044 |
| | | 37 | 5020,752 | 36,752 | 1085,976 |
| | Vidyo3 | 22 | 23550,778 | 44,602 | 1136,641 |
| | | 27 | 14519,712 | 42,23 | 1087,362 |
| | | 32 | 9210,7392 | 39,485 | 1060,743 |
| | | 37 | 5670,2256 | 36,334 | 1045,558 |
| | Vidyo4 | 22 | 25112,246 | 44,295 | 1155,753 |
| | | 27 | 14209,013 | 41,639 | 1109,534 |
| | | 32 | 8218,0128 | 38,972 | 1073,086 |
| | | 37 | 4760,184 | 36,258 | 1060,681 |
| Class F | BasketballDrillText | 22 | 21893,6 | 42,132 | 560,278 |
| | | 27 | 12351,416 | 38,709 | 525,169 |
| | | 32 | 7015,98 | 35,671 | 501,279 |
| | | 37 | 4139,76 | 32,883 | 477,716 |
| | ChinaSpeed | 22 | 24794,1 | 44,935 | 977,683 |
| | | 27 | 16932,446 | 40,97 | 932,529 |
| | | 32 | 11268,444 | 37,148 | 881,989 |
| | | 37 | 7382,9136 | 33,462 | 846,382 |
| | SlideEditing | 22 | 39375,931 | 46,143 | 1252,213 |
| | | 27 | 29538,938 | 41,765 | 1231,706 |
| | | 32 | 22191,278 | 37,157 | 1194,59 |
| | | 37 | 15998,914 | 32,365 | 1154,724 |
| | SlideShow | 22 | 4777,5472 | 51,192 | 885,577 |
| | | 27 | 3139,3888 | 47,607 | 889,814 |
| | | 32 | 2082,6064 | 43,936 | 872,412 |
| | | 37 | 1380,8256 | 40,18 | 861,196 |

Table 5: Results for the FDR heuristic on High Efficiency condition (target level 64).

| Class | Video | QPISlice | kbps | PSNR | Time (sec) |
|---|---|---|---|---|---|
| Class A 4K | Traffic | 22 | 108372,06 | 43,53 | 5529,37 |
| | | 27 | 61361,56 | 40,33 | 5174,04 |
| | | 32 | 35607,17 | 37,27 | 4919,38 |
| | | 37 | 20601,57 | 34,19 | 4759,53 |
| | People on Street | 22 | 111483,65 | 43,46 | 5478,67 |
| | | 27 | 65506,66 | 40,02 | 5223,30 |
| | | 32 | 38427,44 | 36,92 | 4933,18 |
| | | 37 | 23430,45 | 34,04 | 4758,88 |
| Class B 1080p | Kimono | 22 | 18765,24 | 43,06 | 2607,31 |
| | | 27 | 10637,72 | 41,75 | 2460,45 |
| | | 32 | 6397,49 | 39,92 | 2354,32 |
| | | 37 | 3828,94 | 37,52 | 2314,33 |
| | ParkScene | 22 | 54900,16 | 41,79 | 2815,69 |
| | | 27 | 29737,07 | 38,73 | 2584,19 |
| | | 32 | 15517,35 | 35,71 | 2427,96 |
| | | 37 | 7703,57 | 32,83 | 2303,17 |
| | Cactus | 22 | 117078,45 | 40,74 | 2881,37 |
| | | 27 | 54690,38 | 37,95 | 2575,68 |
| | | 32 | 30324,86 | 35,66 | 2422,99 |
| | | 37 | 16938,64 | 33,17 | 2343,14 |
| | BasketballDrive | 22 | 66361,91 | 41,17 | 2430,27 |
| | | 27 | 28043,12 | 38,92 | 2326,13 |
| | | 32 | 15048,06 | 37,28 | 2269,45 |
| | | 37 | 8794,10 | 35,39 | 2203,57 |
| | BQTerrace | 22 | 183087,74 | 42,49 | 2644,45 |
| | | 27 | 94463,45 | 37,78 | 2567,88 |
| | | 32 | 52344,94 | 34,77 | 2439,60 |
| | | 37 | 29731,00 | 31,92 | 2286,49 |
| Class C WVGA | BasketballDrill | 22 | 21574,54 | 41,88 | 538,33 |
| | | 27 | 11748,34 | 38,47 | 503,29 |
| | | 32 | 6450,04 | 35,50 | 470,61 |
| | | 37 | 3716,89 | 32,90 | 450,50 |
| | BQMall | 22 | 28570,51 | 41,64 | 492,17 |
| | | 27 | 17466,67 | 38,59 | 479,93 |
| | | 32 | 10575,03 | 35,36 | 469,46 |
| | | 37 | 6125,85 | 32,08 | 451,64 |
| | PartyScene | 22 | 48910,38 | 41,22 | 553,40 |
| | | 27 | 30986,02 | 36,85 | 560,03 |
| | | 32 | 18834,22 | 32,90 | 540,54 |
| | | 37 | 10601,04 | 29,10 | 504,79 |
| | RaceHorses | 22 | 17524,96 | 42,54 | 505,49 |

| | | | | | |
|---|---|---|---|---|---|
| | | 27 | 10922,40 | 38,88 | 471,44 |
| | | 32 | 6466,57 | 35,10 | 445,35 |
| | | 37 | 3320,56 | 31,26 | 422,81 |
| Class D WQVGA | BasketballPass | 22 | 5097,60 | 43,17 | 121,67 |
| | | 27 | 3046,41 | 39,62 | 118,79 |
| | | 32 | 1744,02 | 36,19 | 115,42 |
| | | 37 | 979,27 | 33,01 | 108,78 |
| | BQSquare | 22 | 13416,16 | 41,53 | 137,07 |
| | | 27 | 8845,07 | 37,12 | 131,54 |
| | | 32 | 5706,75 | 33,37 | 130,26 |
| | | 37 | 3575,44 | 29,75 | 124,36 |
| | BlowingBubbles | 22 | 9770,11 | 41,21 | 136,88 |
| | | 27 | 5788,65 | 37,13 | 137,02 |
| | | 32 | 3213,14 | 33,43 | 127,39 |
| | | 37 | 1678,34 | 30,14 | 117,65 |
| | RaceHorses | 22 | 4982,97 | 42,65 | 131,76 |
| | | 27 | 3074,76 | 38,51 | 123,63 |
| | | 32 | 1743,20 | 34,48 | 115,43 |
| | | 37 | 907,63 | 30,98 | 107,40 |
| Class E 720p | Vidyo1 | 22 | 22732,48 | 44,74 | 1094,10 |
| | | 27 | 13532,52 | 42,38 | 1058,80 |
| | | 32 | 8348,93 | 39,73 | 1030,64 |
| | | 37 | 5165,64 | 36,77 | 1016,65 |
| | Vidyo3 | 22 | 23799,53 | 44,61 | 1067,63 |
| | | 27 | 14771,80 | 42,24 | 1024,68 |
| | | 32 | 9478,86 | 39,50 | 999,32 |
| | | 37 | 5911,04 | 36,34 | 980,05 |
| | Vidyo4 | 22 | 25310,04 | 44,30 | 1091,25 |
| | | 27 | 14362,48 | 41,64 | 1046,13 |
| | | 32 | 8390,11 | 38,98 | 1008,97 |
| | | 37 | 4970,45 | 36,27 | 999,34 |
| Class F | BasketballDrillText | 22 | 21896,44 | 42,13 | 523,57 |
| | | 27 | 12355,64 | 38,70 | 495,08 |
| | | 32 | 7029,69 | 35,66 | 471,18 |
| | | 37 | 4182,30 | 32,89 | 449,47 |
| | ChinaSpeed | 22 | 24811,45 | 44,92 | 904,61 |
| | | 27 | 16957,04 | 40,96 | 858,32 |
| | | 32 | 11298,57 | 37,15 | 814,85 |
| | | 37 | 7419,81 | 33,47 | 788,50 |
| | SlideEditing | 22 | 39376,86 | 46,17 | 1153,75 |
| | | 27 | 29556,23 | 41,77 | 1139,69 |
| | | 32 | 22198,67 | 37,16 | 1110,87 |
| | | 37 | 16005,41 | 32,36 | 1077,83 |
| | SlideShow | 22 | 4808,61 | 51,16 | 824,79 |
| | | 27 | 3173,47 | 47,61 | 829,14 |
| | | 32 | 2120,14 | 43,94 | 821,13 |
| | | 37 | 1421,06 | 40,21 | 802,71 |

Table 6: Results for the MDR heuristic on High Efficiency condition (target level 8).

| Class | Video | QPISlice | kbps | PSNR | Time (sec) |
|---|---|---|---|---|---|
| Class A 4K | Traffic | 22 | 104714,39 | 43,54 | 6655,65 |
| | | 27 | 58475,57 | 40,37 | 6166,28 |
| | | 32 | 33267,29 | 37,33 | 5861,53 |
| | | 37 | 18778,34 | 34,26 | 5491,29 |
| | People on Street | 22 | 107400,41 | 43,45 | 5795,72 |
| | | 27 | 62509,27 | 40,04 | 5389,07 |
| | | 32 | 35850,84 | 36,95 | 5083,67 |
| | | 37 | 20971,36 | 34,09 | 4852,31 |
| Class B 1080p | Kimono | 22 | 18174,89 | 43,08 | 3113,83 |
| | | 27 | 10302,21 | 41,78 | 2880,44 |
| | | 32 | 6169,35 | 39,99 | 2747,12 |
| | | 37 | 3662,70 | 37,60 | 2689,16 |
| | ParkScene | 22 | 53686,15 | 41,77 | 3415,97 |
| | | 27 | 28972,60 | 38,74 | 3096,73 |
| | | 32 | 15069,41 | 35,74 | 2872,39 |
| | | 37 | 7436,81 | 32,88 | 2692,42 |
| | Cactus | 22 | 113202,68 | 40,70 | 3511,51 |
| | | 27 | 52195,18 | 37,95 | 3088,07 |
| | | 32 | 28422,98 | 35,69 | 2858,54 |
| | | 37 | 15470,01 | 33,22 | 2721,22 |
| | BasketballDrive | 22 | 63839,93 | 41,20 | 2988,36 |
| | | 27 | 26230,60 | 38,94 | 2757,23 |
| | | 32 | 13681,88 | 37,33 | 2667,07 |
| | | 37 | 7684,41 | 35,46 | 2550,93 |
| | BQTerrace | 22 | 178649,68 | 42,63 | 3246,26 |
| | | 27 | 90808,44 | 37,79 | 3120,19 |
| | | 32 | 49325,68 | 34,78 | 2911,76 |
| | | 37 | 27248,87 | 31,95 | 2739,04 |
| Class C WVGA | BasketballDrill | 22 | 20970,86 | 41,88 | 665,34 |
| | | 27 | 11218,49 | 38,50 | 611,10 |
| | | 32 | 5970,76 | 35,55 | 564,75 |
| | | 37 | 3322,81 | 32,97 | 533,59 |
| | BQMall | 22 | 27907,42 | 41,64 | 613,60 |
| | | 27 | 16927,60 | 38,62 | 589,78 |
| | | 32 | 10086,35 | 35,42 | 563,00 |
| | | 37 | 5690,30 | 32,16 | 534,36 |
| | PartyScene | 22 | 48551,48 | 41,20 | 704,04 |
| | | 27 | 30639,81 | 36,85 | 695,94 |
| | | 32 | 18503,39 | 32,91 | 654,53 |
| | | 37 | 10259,96 | 29,13 | 602,78 |
| | RaceHorses | 22 | 17096,93 | 42,51 | 600,22 |

| | | | | | |
|---|---|---|---|---|---|
| | | 27 | 10581,17 | 38,86 | 557,55 |
| | | 32 | 6195,28 | 35,10 | 520,96 |
| | | 37 | 3118,89 | 31,27 | 491,76 |
| Class D WQVGA | BasketballPass | 22 | 4944,55 | 43,21 | 146,83 |
| | | 27 | 2921,14 | 39,67 | 140,55 |
| | | 32 | 1640,19 | 36,23 | 133,99 |
| | | 37 | 899,98 | 33,07 | 127,26 |
| | BQSquare | 22 | 13326,97 | 41,53 | 172,97 |
| | | 27 | 8763,19 | 37,13 | 164,60 |
| | | 32 | 5628,40 | 33,37 | 159,51 |
| | | 37 | 3496,41 | 29,74 | 149,98 |
| | BlowingBubbles | 22 | 9662,51 | 41,18 | 169,05 |
| | | 27 | 5690,95 | 37,12 | 164,23 |
| | | 32 | 3120,03 | 33,44 | 151,00 |
| | | 37 | 1596,47 | 30,17 | 137,72 |
| | RaceHorses | 22 | 4891,62 | 42,63 | 158,08 |
| | | 27 | 2992,74 | 38,50 | 147,11 |
| | | 32 | 1680,27 | 34,50 | 136,43 |
| | | 37 | 851,28 | 30,99 | 125,70 |
| Class E 720p | Vidyo1 | 22 | 21317,01 | 44,76 | 1309,49 |
| | | 27 | 12430,48 | 42,44 | 1253,12 |
| | | 32 | 7458,58 | 39,81 | 1214,58 |
| | | 37 | 4485,07 | 36,90 | 1175,27 |
| | Vidyo3 | 22 | 22627,47 | 44,65 | 1287,75 |
| | | 27 | 13810,79 | 42,30 | 1218,51 |
| | | 32 | 8606,99 | 39,60 | 1196,24 |
| | | 37 | 5221,33 | 36,50 | 1167,24 |
| | Vidyo4 | 22 | 24056,28 | 44,32 | 1292,48 |
| | | 27 | 13402,21 | 41,70 | 1219,34 |
| | | 32 | 7597,68 | 39,08 | 1178,83 |
| | | 37 | 4284,66 | 36,40 | 1153,45 |
| Class F | BasketballDrillText | 22 | 21343,88 | 42,15 | 644,53 |
| | | 27 | 11865,62 | 38,74 | 601,43 |
| | | 32 | 6578,39 | 35,72 | 565,71 |
| | | 37 | 3786,27 | 32,97 | 534,44 |
| | ChinaSpeed | 22 | 24432,44 | 44,96 | 1141,78 |
| | | 27 | 16598,37 | 40,99 | 1083,83 |
| | | 32 | 10960,94 | 37,18 | 1019,36 |
| | | 37 | 7123,91 | 33,51 | 972,54 |
| | SlideEditing | 22 | 39122,79 | 46,21 | 1455,15 |
| | | 27 | 29285,75 | 41,84 | 1420,24 |
| | | 32 | 21951,61 | 37,24 | 1377,85 |
| | | 37 | 15749,39 | 32,44 | 1320,49 |
| | SlideShow | 22 | 4624,76 | 51,18 | 1001,80 |
| | | 27 | 3014,20 | 47,63 | 995,49 |
| | | 32 | 1974,11 | 43,98 | 978,60 |
| | | 37 | 1289,24 | 40,30 | 959,57 |

Table 7: Results for the MDR heuristic on High Efficiency condition (target level 16).

| Class | Video | QPISlice | kbps | PSNR | Time (sec) |
|---|---|---|---|---|---|
| Class A 4K | Traffic | 22 | 105459,21 | 43,53 | 6398,11 |
| | | 27 | 59075,18 | 40,36 | 5957,74 |
| | | 32 | 33731,75 | 37,31 | 5617,65 |
| | | 37 | 19120,21 | 34,23 | 5302,04 |
| | People on Street | 22 | 108120,40 | 43,45 | 5536,03 |
| | | 27 | 63123,51 | 40,04 | 5174,09 |
| | | 32 | 36411,60 | 36,94 | 4868,28 |
| | | 37 | 21534,18 | 34,07 | 4674,00 |
| Class B 1080p | Kimono | 22 | 18219,52 | 43,08 | 2984,07 |
| | | 27 | 10345,21 | 41,78 | 2741,89 |
| | | 32 | 6200,43 | 39,97 | 2628,85 |
| | | 37 | 3684,03 | 37,58 | 2585,64 |
| | ParkScene | 22 | 53764,49 | 41,77 | 3292,17 |
| | | 27 | 29005,56 | 38,74 | 2977,13 |
| | | 32 | 15095,02 | 35,74 | 2765,14 |
| | | 37 | 7460,38 | 32,87 | 2566,96 |
| | Cactus | 22 | 113711,94 | 40,70 | 3388,81 |
| | | 27 | 52577,20 | 37,94 | 2957,42 |
| | | 32 | 28751,04 | 35,68 | 2749,05 |
| | | 37 | 15742,54 | 33,20 | 2627,69 |
| | BasketballDrive | 22 | 64263,73 | 41,21 | 2850,71 |
| | | 27 | 26497,70 | 38,94 | 2632,30 |
| | | 32 | 13885,12 | 37,32 | 2545,57 |
| | | 37 | 7851,93 | 35,44 | 2456,84 |
| | BQTerrace | 22 | 179113,27 | 42,64 | 3108,23 |
| | | 27 | 91321,82 | 37,79 | 3013,37 |
| | | 32 | 49747,67 | 34,78 | 2798,08 |
| | | 37 | 27611,52 | 31,94 | 2615,40 |
| Class C WVGA | BasketballDrill | 22 | 21172,98 | 41,88 | 634,90 |
| | | 27 | 11382,57 | 38,50 | 585,81 |
| | | 32 | 6093,68 | 35,54 | 545,14 |
| | | 37 | 3401,56 | 32,94 | 515,28 |
| | BQMall | 22 | 28005,96 | 41,64 | 586,13 |
| | | 27 | 17017,79 | 38,62 | 565,58 |
| | | 32 | 10165,04 | 35,41 | 543,21 |
| | | 37 | 5765,62 | 32,14 | 514,07 |
| | PartyScene | 22 | 48581,08 | 41,20 | 677,25 |
| | | 27 | 30670,94 | 36,85 | 671,19 |
| | | 32 | 18533,12 | 32,91 | 632,39 |
| | | 37 | 10294,85 | 29,13 | 580,57 |
| | RaceHorses | 22 | 17156,10 | 42,51 | 580,55 |

| | | | | | |
|---|---|---|---|---|---|
| | | 27 | 10632,67 | 38,86 | 536,22 |
| | | 32 | 6237,92 | 35,09 | 503,41 |
| | | 37 | 3162,68 | 31,27 | 470,24 |
| Class D WQVGA | BasketballPass | 22 | 4973,17 | 43,20 | 141,09 |
| | | 27 | 2939,60 | 39,66 | 134,25 |
| | | 32 | 1659,45 | 36,23 | 129,19 |
| | | 37 | 914,98 | 33,06 | 120,93 |
| | BQSquare | 22 | 13345,15 | 41,53 | 167,01 |
| | | 27 | 8771,27 | 37,12 | 159,13 |
| | | 32 | 5639,18 | 33,37 | 152,85 |
| | | 37 | 3506,27 | 29,74 | 142,72 |
| | BlowingBubbles | 22 | 9669,08 | 41,18 | 165,22 |
| | | 27 | 5702,32 | 37,13 | 159,51 |
| | | 32 | 3131,79 | 33,44 | 145,85 |
| | | 37 | 1612,78 | 30,17 | 132,70 |
| | RaceHorses | 22 | 4899,92 | 42,63 | 153,06 |
| | | 27 | 3005,71 | 38,50 | 141,95 |
| | | 32 | 1691,99 | 34,49 | 130,96 |
| | | 37 | 864,74 | 30,99 | 120,51 |
| Class E 720p | Vidyo1 | 22 | 21678,60 | 44,76 | 1263,94 |
| | | 27 | 12709,01 | 42,42 | 1202,75 |
| | | 32 | 7667,79 | 39,79 | 1162,42 |
| | | 37 | 4627,37 | 36,85 | 1138,89 |
| | Vidyo3 | 22 | 22788,88 | 44,64 | 1236,64 |
| | | 27 | 13944,03 | 42,29 | 1170,16 |
| | | 32 | 8719,21 | 39,58 | 1152,07 |
| | | 37 | 5322,35 | 36,47 | 1129,06 |
| | Vidyo4 | 22 | 24272,02 | 44,32 | 1242,30 |
| | | 27 | 13564,01 | 41,69 | 1180,20 |
| | | 32 | 7718,09 | 39,05 | 1134,91 |
| | | 37 | 4385,15 | 36,36 | 1115,18 |
| Class F | BasketballDrillText | 22 | 21536,84 | 42,14 | 618,70 |
| | | 27 | 12021,23 | 38,74 | 576,15 |
| | | 32 | 6687,36 | 35,71 | 544,13 |
| | | 37 | 3859,03 | 32,94 | 513,85 |
| | ChinaSpeed | 22 | 24482,84 | 44,94 | 1092,70 |
| | | 27 | 16654,96 | 40,99 | 1031,69 |
| | | 32 | 11023,08 | 37,18 | 971,80 |
| | | 37 | 7176,84 | 33,51 | 931,32 |
| | SlideEditing | 22 | 39132,91 | 46,20 | 1392,02 |
| | | 27 | 29284,56 | 41,84 | 1358,96 |
| | | 32 | 21953,89 | 37,23 | 1321,50 |
| | | 37 | 15749,22 | 32,43 | 1260,18 |
| | SlideShow | 22 | 4660,75 | 51,18 | 969,49 |
| | | 27 | 3041,95 | 47,62 | 947,32 |
| | | 32 | 1998,25 | 43,96 | 934,33 |
| | | 37 | 1300,47 | 40,27 | 908,28 |

Table 8: Results for the MDR heuristic on High Efficiency condition (target level 32).

| Class | Video | QPISlice | kbps | PSNR | Time (sec) |
|---|---|---|---|---|---|
| Class A 4K | Traffic | 22 | 105846,64 | 43,54 | 1,71 |
| | | 27 | 59426,11 | 40,36 | 1,59 |
| | | 32 | 34047,82 | 37,32 | 1,50 |
| | | 37 | 19395,60 | 34,23 | 1,42 |
| | People on Street | 22 | 108336,84 | 43,45 | 1,49 |
| | | 27 | 63249,24 | 40,04 | 1,38 |
| | | 32 | 36563,21 | 36,95 | 1,31 |
| | | 37 | 21770,24 | 34,08 | 1,25 |
| Class B 1080p | Kimono | 22 | 18230,00 | 43,07 | 0,79 |
| | | 27 | 10371,66 | 41,77 | 0,73 |
| | | 32 | 6225,66 | 39,96 | 0,70 |
| | | 37 | 3710,61 | 37,57 | 0,68 |
| | ParkScene | 22 | 53787,95 | 41,77 | 0,88 |
| | | 27 | 29025,95 | 38,74 | 0,79 |
| | | 32 | 15113,58 | 35,74 | 0,73 |
| | | 37 | 7487,88 | 32,87 | 0,68 |
| | Cactus | 22 | 113970,01 | 40,70 | 0,90 |
| | | 27 | 52804,45 | 37,94 | 0,79 |
| | | 32 | 28949,43 | 35,68 | 0,74 |
| | | 37 | 15920,19 | 33,20 | 0,70 |
| | BasketballDrive | 22 | 64514,42 | 41,21 | 0,76 |
| | | 27 | 26657,64 | 38,94 | 0,70 |
| | | 32 | 14022,95 | 37,32 | 0,68 |
| | | 37 | 7992,10 | 35,44 | 0,65 |
| | BQTerrace | 22 | 179365,40 | 42,64 | 0,83 |
| | | 27 | 91592,38 | 37,79 | 0,80 |
| | | 32 | 50018,49 | 34,78 | 0,75 |
| | | 37 | 27844,15 | 31,95 | 0,70 |
| Class C WVGA | BasketballDrill | 22 | 21218,02 | 41,88 | 0,17 |
| | | 27 | 11414,29 | 38,49 | 0,16 |
| | | 32 | 6138,68 | 35,54 | 0,14 |
| | | 37 | 3446,89 | 32,93 | 0,14 |
| | BQMall | 22 | 28052,06 | 41,64 | 0,16 |
| | | 27 | 17058,79 | 38,62 | 0,15 |
| | | 32 | 10205,92 | 35,41 | 0,14 |
| | | 37 | 5797,52 | 32,15 | 0,14 |
| | PartyScene | 22 | 48581,79 | 41,20 | 0,18 |
| | | 27 | 30670,51 | 36,85 | 0,18 |
| | | 32 | 18534,30 | 32,91 | 0,17 |
| | | 37 | 10297,02 | 29,13 | 0,16 |
| | RaceHorses | 22 | 17162,17 | 42,51 | 0,16 |

| | | | | | |
|---|---|---|---|---|---|
| | | 27 | 10640,59 | 38,86 | 0,14 |
| | | 32 | 6248,94 | 35,09 | 0,13 |
| | | 37 | 3173,20 | 31,27 | 0,12 |
| Class D WQVGA | BasketballPass | 22 | 4990,10 | 43,20 | 0,04 |
| | | 27 | 2955,62 | 39,66 | 0,04 |
| | | 32 | 1670,72 | 36,23 | 0,03 |
| | | 37 | 923,63 | 33,05 | 0,03 |
| | BQSquare | 22 | 13345,58 | 41,53 | 0,05 |
| | | 27 | 8774,08 | 37,12 | 0,04 |
| | | 32 | 5639,04 | 33,37 | 0,04 |
| | | 37 | 3511,51 | 29,74 | 0,04 |
| | BlowingBubbles | 22 | 9668,84 | 41,18 | 0,04 |
| | | 27 | 5702,47 | 37,12 | 0,04 |
| | | 32 | 3131,52 | 33,44 | 0,04 |
| | | 37 | 1616,25 | 30,17 | 0,04 |
| | RaceHorses | 22 | 4901,58 | 42,63 | 0,04 |
| | | 27 | 3006,35 | 38,50 | 0,04 |
| | | 32 | 1690,33 | 34,48 | 0,04 |
| | | 37 | 866,66 | 30,99 | 0,03 |
| Class E 720p | Vidyo1 | 22 | 21848,99 | 44,77 | 0,34 |
| | | 27 | 12863,83 | 42,42 | 0,32 |
| | | 32 | 7793,45 | 39,78 | 0,31 |
| | | 37 | 4730,44 | 36,84 | 0,30 |
| | Vidyo3 | 22 | 22875,78 | 44,64 | 0,33 |
| | | 27 | 14014,58 | 42,29 | 0,31 |
| | | 32 | 8789,75 | 39,57 | 0,31 |
| | | 37 | 5385,48 | 36,47 | 0,30 |
| | Vidyo4 | 22 | 24363,81 | 44,32 | 0,33 |
| | | 27 | 13658,87 | 41,68 | 0,31 |
| | | 32 | 7805,11 | 39,05 | 0,30 |
| | | 37 | 4462,09 | 36,35 | 0,30 |
| Class F | BasketballDrillText | 22 | 21578,14 | 42,14 | 0,17 |
| | | 27 | 12048,96 | 38,73 | 0,15 |
| | | 32 | 6724,94 | 35,70 | 0,15 |
| | | 37 | 3897,59 | 32,93 | 0,14 |
| | ChinaSpeed | 22 | 24497,48 | 44,95 | 0,29 |
| | | 27 | 16674,41 | 40,99 | 0,27 |
| | | 32 | 11044,22 | 37,17 | 0,26 |
| | | 37 | 7196,70 | 33,51 | 0,25 |
| | SlideEditing | 22 | 39138,20 | 46,23 | 0,37 |
| | | 27 | 29292,59 | 41,85 | 0,36 |
| | | 32 | 21953,25 | 37,23 | 0,35 |
| | | 37 | 15752,37 | 32,43 | 0,33 |
| | SlideShow | 22 | 4660,70 | 51,18 | 0,25 |
| | | 27 | 3045,46 | 47,62 | 0,25 |
| | | 32 | 2006,50 | 43,97 | 0,25 |
| | | 37 | 1315,72 | 40,27 | 0,24 |

Table 9: Results for the MDR heuristic on High Efficiency condition (target level 64).

| Class | Video | QPISlice | kbps | PSNR | Time (sec) |
|---|---|---|---|---|---|
| Class A 4K | Traffic | 22 | 105885,78 | 43,54 | 5892,13 |
| | | 27 | 59498,78 | 40,36 | 5479,37 |
| | | 32 | 34153,74 | 37,32 | 5183,68 |
| | | 37 | 19530,70 | 34,25 | 4875,81 |
| | People on Street | 22 | 108348,39 | 43,45 | 5115,50 |
| | | 27 | 63262,70 | 40,04 | 4760,45 |
| | | 32 | 36589,95 | 36,95 | 4522,11 |
| | | 37 | 21832,71 | 34,08 | 4308,02 |
| Class B 1080p | Kimono | 22 | 18236,67 | 43,07 | 2714,43 |
| | | 27 | 10377,31 | 41,77 | 2515,04 |
| | | 32 | 6235,42 | 39,96 | 2441,13 |
| | | 37 | 3720,57 | 37,57 | 2364,15 |
| | ParkScene | 22 | 53784,80 | 41,77 | 3050,04 |
| | | 27 | 29023,43 | 38,74 | 2740,14 |
| | | 32 | 15117,54 | 35,74 | 2532,17 |
| | | 37 | 7499,05 | 32,88 | 2351,43 |
| | Cactus | 22 | 114015,26 | 40,70 | 3119,89 |
| | | 27 | 52967,62 | 37,95 | 2738,22 |
| | | 32 | 29143,54 | 35,69 | 2543,18 |
| | | 37 | 16090,55 | 33,21 | 2416,65 |
| | BasketballDrive | 22 | 64523,96 | 41,21 | 2597,43 |
| | | 27 | 26693,09 | 38,94 | 2416,34 |
| | | 32 | 14079,22 | 37,32 | 2335,37 |
| | | 37 | 8064,72 | 35,44 | 2246,52 |
| | BQTerrace | 22 | 180585,98 | 42,65 | 2876,86 |
| | | 27 | 92665,67 | 37,80 | 2775,10 |
| | | 32 | 50697,45 | 34,78 | 2597,58 |
| | | 37 | 28348,86 | 31,95 | 2416,58 |
| Class C WVGA | BasketballDrill | 22 | 21219,15 | 41,89 | 585,97 |
| | | 27 | 11409,32 | 38,49 | 538,24 |
| | | 32 | 6137,84 | 35,53 | 500,85 |
| | | 37 | 3452,48 | 32,92 | 472,38 |
| | BQMall | 22 | 28054,64 | 41,64 | 539,37 |
| | | 27 | 17061,35 | 38,62 | 520,60 |
| | | 32 | 10208,60 | 35,41 | 501,91 |
| | | 37 | 5800,50 | 32,15 | 475,58 |
| | PartyScene | 22 | 48583,56 | 41,20 | 627,82 |
| | | 27 | 30674,82 | 36,85 | 625,05 |
| | | 32 | 18537,67 | 32,91 | 588,80 |
| | | 37 | 10297,25 | 29,13 | 540,09 |
| | RaceHorses | 22 | 17165,81 | 42,51 | 538,10 |

| | | | | | |
|---|---|---|---|---|---|
| | | 27 | 10643,52 | 38,86 | 495,96 |
| | | 32 | 6250,68 | 35,09 | 463,55 |
| | | 37 | 3177,80 | 31,28 | 433,39 |
| Class D WQVGA | BasketballPass | 22 | 4996,42 | 43,20 | 132,05 |
| | | 27 | 2959,32 | 39,66 | 125,72 |
| | | 32 | 1672,16 | 36,22 | 120,41 |
| | | 37 | 924,78 | 33,04 | 112,67 |
| | BQSquare | 22 | 13345,69 | 41,53 | 158,51 |
| | | 27 | 8772,07 | 37,12 | 150,37 |
| | | 32 | 5641,20 | 33,37 | 143,14 |
| | | 37 | 3509,62 | 29,74 | 133,75 |
| | BlowingBubbles | 22 | 9674,10 | 41,18 | 152,91 |
| | | 27 | 5704,44 | 37,13 | 150,49 |
| | | 32 | 3132,58 | 33,44 | 137,16 |
| | | 37 | 1615,16 | 30,17 | 123,71 |
| | RaceHorses | 22 | 4901,32 | 42,63 | 144,17 |
| | | 27 | 3005,73 | 38,50 | 133,65 |
| | | 32 | 1690,69 | 34,49 | 123,64 |
| | | 37 | 867,26 | 30,99 | 113,87 |
| Class E 720p | Vidyo1 | 22 | 21910,20 | 44,77 | 1152,44 |
| | | 27 | 12926,88 | 42,43 | 1109,67 |
| | | 32 | 7846,09 | 39,79 | 1066,87 |
| | | 37 | 4768,43 | 36,83 | 1041,79 |
| | Vidyo3 | 22 | 22906,29 | 44,64 | 1126,69 |
| | | 27 | 14051,59 | 42,29 | 1073,75 |
| | | 32 | 8821,32 | 39,57 | 1050,24 |
| | | 37 | 5420,61 | 36,47 | 1035,61 |
| | Vidyo4 | 22 | 24375,42 | 44,32 | 1141,52 |
| | | 27 | 13697,18 | 41,69 | 1081,88 |
| | | 32 | 7845,72 | 39,05 | 1038,18 |
| | | 37 | 4517,66 | 36,36 | 1026,08 |
| Class F | BasketballDrillText | 22 | 21583,09 | 42,14 | 571,76 |
| | | 27 | 12051,49 | 38,73 | 533,52 |
| | | 32 | 6721,77 | 35,69 | 502,39 |
| | | 37 | 3900,04 | 32,92 | 472,09 |
| | ChinaSpeed | 22 | 24497,28 | 44,93 | 1000,20 |
| | | 27 | 16675,48 | 40,99 | 947,49 |
| | | 32 | 11049,23 | 37,18 | 889,07 |
| | | 37 | 7205,85 | 33,51 | 845,63 |
| | SlideEditing | 22 | 39133,63 | 46,23 | 1273,76 |
| | | 27 | 29287,88 | 41,85 | 1245,61 |
| | | 32 | 21955,65 | 37,23 | 1207,35 |
| | | 37 | 15751,54 | 32,43 | 1154,63 |
| | SlideShow | 22 | 4663,72 | 51,17 | 859,98 |
| | | 27 | 3044,94 | 47,62 | 855,03 |
| | | 32 | 2008,46 | 43,96 | 841,74 |
| | | 37 | 1315,09 | 40,26 | 827,41 |

Table 10: Results for the CDR heuristic on High Efficiency condition (target level 8).

| Class | Video | QPISlice | kbps | PSNR | Time (sec) |
|---|---|---|---|---|---|
| Class A 4K | Traffic | 22 | 104842,64 | 43,53 | 6543,98 |
| | | 27 | 58589,47 | 40,36 | 6091,36 |
| | | 32 | 33346,07 | 37,32 | 5788,71 |
| | | 37 | 18819,73 | 34,25 | 5528,98 |
| | People on Street | 22 | 107681,47 | 43,45 | 5692,16 |
| | | 27 | 62701,64 | 40,04 | 5308,88 |
| | | 32 | 36026,06 | 36,94 | 5012,51 |
| | | 37 | 21102,58 | 34,07 | 4803,52 |
| Class B 1080p | Kimono | 22 | 18181,25 | 43,08 | 3075,06 |
| | | 27 | 10308,24 | 41,78 | 2858,13 |
| | | 32 | 6174,41 | 39,98 | 2736,73 |
| | | 37 | 3667,66 | 37,60 | 2678,68 |
| | ParkScene | 22 | 53743,31 | 41,77 | 3325,03 |
| | | 27 | 29004,98 | 38,73 | 3034,47 |
| | | 32 | 15087,73 | 35,73 | 2825,09 |
| | | 37 | 7441,92 | 32,87 | 2629,30 |
| | Cactus | 22 | 113346,48 | 40,69 | 3478,72 |
| | | 27 | 52296,69 | 37,94 | 3071,39 |
| | | 32 | 28499,94 | 35,68 | 2865,25 |
| | | 37 | 15525,82 | 33,21 | 2718,37 |
| | BasketballDrive | 22 | 63692,34 | 41,18 | 2890,67 |
| | | 27 | 26297,48 | 38,94 | 2722,75 |
| | | 32 | 13718,19 | 37,33 | 2630,20 |
| | | 37 | 7700,04 | 35,45 | 2528,09 |
| | BQTerrace | 22 | 178806,44 | 42,57 | 2801,19 |
| | | 27 | 90894,06 | 37,78 | 2664,64 |
| | | 32 | 49461,17 | 34,78 | 2499,45 |
| | | 37 | 27349,94 | 31,94 | 2344,47 |
| Class C WVGA | BasketballDrill | 22 | 21004,78 | 41,88 | 650,43 |
| | | 27 | 11260,05 | 38,50 | 599,50 |
| | | 32 | 5989,35 | 35,54 | 561,52 |
| | | 37 | 3336,76 | 32,97 | 531,04 |
| | BQMall | 22 | 27966,77 | 41,64 | 607,43 |
| | | 27 | 16969,80 | 38,62 | 584,75 |
| | | 32 | 10125,83 | 35,41 | 561,70 |
| | | 37 | 5717,71 | 32,14 | 534,54 |
| | PartyScene | 22 | 48609,20 | 41,20 | 698,79 |
| | | 27 | 30695,72 | 36,85 | 687,94 |
| | | 32 | 18560,44 | 32,91 | 648,81 |
| | | 37 | 10308,48 | 29,12 | 602,83 |
| | RaceHorses | 22 | 17121,11 | 42,51 | 596,90 |

| | | | | | |
|---|---|---|---|---|---|
| | | 27 | 10602,62 | 38,86 | 551,99 |
| | | 32 | 6209,96 | 35,09 | 518,26 |
| | | 37 | 3128,11 | 31,26 | 487,23 |
| Class D WQVGA | BasketballPass | 22 | 4952,84 | 43,20 | 145,51 |
| | | 27 | 2926,06 | 39,66 | 139,35 |
| | | 32 | 1643,95 | 36,22 | 134,18 |
| | | 37 | 905,00 | 33,06 | 126,92 |
| | BQSquare | 22 | 13336,76 | 41,52 | 174,67 |
| | | 27 | 8772,26 | 37,12 | 164,81 |
| | | 32 | 5646,58 | 33,36 | 158,00 |
| | | 37 | 3511,38 | 29,74 | 148,59 |
| | BlowingBubbles | 22 | 9673,87 | 41,18 | 168,96 |
| | | 27 | 5702,63 | 37,12 | 164,17 |
| | | 32 | 3128,80 | 33,43 | 150,64 |
| | | 37 | 1602,73 | 30,16 | 138,12 |
| | RaceHorses | 22 | 4897,35 | 42,62 | 156,37 |
| | | 27 | 3000,54 | 38,49 | 145,76 |
| | | 32 | 1687,16 | 34,48 | 134,64 |
| | | 37 | 855,89 | 30,97 | 124,69 |
| Class E 720p | Vidyo1 | 22 | 21367,03 | 44,76 | 1293,57 |
| | | 27 | 12470,19 | 42,43 | 1235,75 |
| | | 32 | 7484,10 | 39,80 | 1192,25 |
| | | 37 | 4501,80 | 36,89 | 1162,46 |
| | Vidyo3 | 22 | 22697,78 | 44,64 | 1253,63 |
| | | 27 | 13868,37 | 42,30 | 1197,03 |
| | | 32 | 8655,26 | 39,59 | 1162,91 |
| | | 37 | 5250,84 | 36,45 | 1137,33 |
| | Vidyo4 | 22 | 24095,90 | 44,32 | 1263,55 |
| | | 27 | 13441,81 | 41,70 | 1193,65 |
| | | 32 | 7626,67 | 39,07 | 1160,06 |
| | | 37 | 4296,82 | 36,39 | 1137,35 |
| Class F | BasketballDrillText | 22 | 21392,06 | 42,15 | 639,00 |
| | | 27 | 11909,22 | 38,74 | 598,70 |
| | | 32 | 6604,94 | 35,71 | 561,57 |
| | | 37 | 3805,35 | 32,96 | 533,80 |
| | ChinaSpeed | 22 | 24476,97 | 44,96 | 1131,85 |
| | | 27 | 16636,26 | 40,98 | 1068,62 |
| | | 32 | 10989,54 | 37,16 | 1011,59 |
| | | 37 | 7150,08 | 33,50 | 972,73 |
| | SlideEditing | 22 | 39153,92 | 46,22 | 1437,99 |
| | | 27 | 29328,16 | 41,85 | 1410,35 |
| | | 32 | 21990,30 | 37,23 | 1366,75 |
| | | 37 | 15799,10 | 32,43 | 1303,38 |
| | SlideShow | 22 | 4643,02 | 51,20 | 997,24 |
| | | 27 | 3024,09 | 47,62 | 987,50 |
| | | 32 | 1984,65 | 43,97 | 980,79 |
| | | 37 | 1293,14 | 40,30 | 951,15 |

Table 11: Results for the CDR heuristic on High Efficiency condition (target level 16).

| Class | Video | QPISlice | kbps | PSNR | Time (sec) |
|---|---|---|---|---|---|
| Class A<br>4K | Traffic | 22 | 105818,41 | 43,52 | 6249,89 |
| | | 27 | 59364,78 | 40,34 | 5820,74 |
| | | 32 | 33965,72 | 37,29 | 5538,64 |
| | | 37 | 19286,38 | 34,21 | 5330,66 |
| | People on Street | 22 | 108718,06 | 43,44 | 5414,52 |
| | | 27 | 63642,13 | 40,02 | 5078,94 |
| | | 32 | 36891,60 | 36,92 | 4788,14 |
| | | 37 | 21942,33 | 34,04 | 4619,45 |
| Class B<br>1080p | Kimono | 22 | 18241,27 | 43,07 | 2908,41 |
| | | 27 | 10364,23 | 41,77 | 2718,69 |
| | | 32 | 6216,94 | 39,96 | 2631,08 |
| | | 37 | 3692,93 | 37,57 | 2567,00 |
| | ParkScene | 22 | 53913,17 | 41,76 | 3174,26 |
| | | 27 | 29102,78 | 38,72 | 2879,95 |
| | | 32 | 15145,27 | 35,71 | 2672,36 |
| | | 37 | 7482,87 | 32,85 | 2513,20 |
| | Cactus | 22 | 114145,72 | 40,69 | 3316,12 |
| | | 27 | 52871,48 | 37,94 | 2921,38 |
| | | 32 | 28971,02 | 35,67 | 2730,22 |
| | | 37 | 15894,83 | 33,18 | 2610,29 |
| | BasketballDrive | 22 | 64356,02 | 41,19 | 2753,80 |
| | | 27 | 26683,06 | 38,93 | 2584,30 |
| | | 32 | 14024,50 | 37,32 | 2502,07 |
| | | 37 | 7939,79 | 35,43 | 2425,91 |
| | BQTerrace | 22 | 179357,12 | 42,59 | 2627,95 |
| | | 27 | 91610,47 | 37,78 | 2536,73 |
| | | 32 | 50094,34 | 34,77 | 2380,97 |
| | | 37 | 27899,23 | 31,93 | 2244,57 |
| Class C<br>WVGA | BasketballDrill | 22 | 21233,68 | 41,87 | 617,12 |
| | | 27 | 11473,06 | 38,50 | 573,06 |
| | | 32 | 6164,91 | 35,53 | 537,26 |
| | | 37 | 3464,21 | 32,93 | 509,09 |
| | BQMall | 22 | 28123,52 | 41,64 | 573,53 |
| | | 27 | 17102,03 | 38,61 | 554,23 |
| | | 32 | 10253,59 | 35,39 | 534,37 |
| | | 37 | 5840,40 | 32,12 | 507,36 |
| | PartyScene | 22 | 48651,06 | 41,20 | 658,65 |
| | | 27 | 30753,18 | 36,85 | 655,66 |
| | | 32 | 18614,58 | 32,90 | 619,37 |
| | | 37 | 10388,41 | 29,12 | 573,21 |
| | RaceHorses | 22 | 17213,41 | 42,51 | 567,58 |

| | | | | | |
|---|---|---|---|---|---|
| | | 27 | 10683,71 | 38,85 | 527,19 |
| | | 32 | 6283,03 | 35,08 | 495,64 |
| | | 37 | 3198,31 | 31,25 | 465,52 |
| Class D WQVGA | BasketballPass | 22 | 4989,76 | 43,19 | 138,42 |
| | | 27 | 2957,12 | 39,65 | 132,42 |
| | | 32 | 1673,32 | 36,22 | 128,38 |
| | | 37 | 928,42 | 33,04 | 121,81 |
| | BQSquare | 22 | 13366,87 | 41,52 | 166,07 |
| | | 27 | 8799,29 | 37,12 | 155,88 |
| | | 32 | 5666,90 | 33,36 | 150,55 |
| | | 37 | 3537,86 | 29,73 | 140,79 |
| | BlowingBubbles | 22 | 9690,48 | 41,18 | 159,80 |
| | | 27 | 5721,60 | 37,12 | 156,09 |
| | | 32 | 3153,26 | 33,43 | 143,13 |
| | | 37 | 1630,60 | 30,15 | 130,27 |
| | RaceHorses | 22 | 4919,17 | 42,63 | 149,36 |
| | | 27 | 3021,56 | 38,49 | 139,15 |
| | | 32 | 1705,92 | 34,48 | 128,72 |
| | | 37 | 878,79 | 30,98 | 119,05 |
| Class E 720p | Vidyo1 | 22 | 21851,19 | 44,76 | 1242,72 |
| | | 27 | 12841,53 | 42,41 | 1186,49 |
| | | 32 | 7775,50 | 39,76 | 1145,85 |
| | | 37 | 4698,04 | 36,82 | 1130,69 |
| | Vidyo3 | 22 | 22954,00 | 44,63 | 1194,16 |
| | | 27 | 14085,85 | 42,27 | 1146,46 |
| | | 32 | 8839,58 | 39,55 | 1114,78 |
| | | 37 | 5417,22 | 36,41 | 1091,32 |
| | Vidyo4 | 22 | 24395,99 | 44,31 | 1217,59 |
| | | 27 | 13679,87 | 41,67 | 1152,27 |
| | | 32 | 7809,39 | 39,03 | 1108,66 |
| | | 37 | 4458,16 | 36,33 | 1090,72 |
| Class F | BasketballDrillText | 22 | 21595,03 | 42,13 | 606,52 |
| | | 27 | 12107,51 | 38,74 | 567,61 |
| | | 32 | 6757,20 | 35,69 | 537,36 |
| | | 37 | 3918,22 | 32,92 | 508,71 |
| | ChinaSpeed | 22 | 24567,58 | 44,95 | 1071,04 |
| | | 27 | 16730,16 | 40,99 | 1012,18 |
| | | 32 | 11096,64 | 37,16 | 950,64 |
| | | 37 | 7237,88 | 33,49 | 914,04 |
| | SlideEditing | 22 | 39175,95 | 46,19 | 1357,03 |
| | | 27 | 29337,84 | 41,83 | 1334,70 |
| | | 32 | 22009,02 | 37,22 | 1289,28 |
| | | 37 | 15799,35 | 32,42 | 1240,66 |
| | SlideShow | 22 | 4692,92 | 51,21 | 943,81 |
| | | 27 | 3070,78 | 47,61 | 939,64 |
| | | 32 | 2022,91 | 43,96 | 927,38 |
| | | 37 | 1328,14 | 40,27 | 906,15 |

Table 12: Results for the CDR heuristic on High Efficiency condition (target level 32).

| Class | Video | QPISlice | kbps | PSNR | Time (sec) |
|---|---|---|---|---|---|
| Class A 4K | Traffic | 22 | 106395,19 | 43,53 | 5984,36 |
| | | 27 | 59849,10 | 40,34 | 5538,84 |
| | | 32 | 34381,54 | 37,29 | 5313,25 |
| | | 37 | 19663,77 | 34,20 | 5052,33 |
| | People on Street | 22 | 109137,57 | 43,44 | 5173,09 |
| | | 27 | 63897,06 | 40,02 | 4832,13 |
| | | 32 | 37164,36 | 36,93 | 4558,38 |
| | | 37 | 22327,86 | 34,04 | 4397,99 |
| Class B 1080p | Kimono | 22 | 18311,35 | 43,06 | 2767,02 |
| | | 27 | 10427,50 | 41,76 | 2601,82 |
| | | 32 | 6261,08 | 39,94 | 2500,55 |
| | | 37 | 3731,77 | 37,54 | 2442,05 |
| | ParkScene | 22 | 54047,03 | 41,76 | 3029,20 |
| | | 27 | 29180,39 | 38,72 | 2736,95 |
| | | 32 | 15208,44 | 35,72 | 2556,82 |
| | | 37 | 7538,46 | 32,85 | 2400,46 |
| | Cactus | 22 | 114697,41 | 40,70 | 3156,62 |
| | | 27 | 53235,86 | 37,94 | 2782,90 |
| | | 32 | 29272,56 | 35,67 | 2603,08 |
| | | 37 | 16169,44 | 33,18 | 2481,35 |
| | BasketballDrive | 22 | 64780,82 | 41,19 | 2592,54 |
| | | 27 | 26961,46 | 38,93 | 2452,49 |
| | | 32 | 14244,80 | 37,31 | 2396,59 |
| | | 37 | 8125,21 | 35,42 | 2311,15 |
| | BQTerrace | 22 | 179641,53 | 42,59 | 2478,58 |
| | | 27 | 92031,51 | 37,78 | 2409,85 |
| | | 32 | 50473,21 | 34,77 | 2268,12 |
| | | 37 | 28243,52 | 31,93 | 2139,59 |
| Class C WVGA | BasketballDrill | 22 | 21283,61 | 41,88 | 589,42 |
| | | 27 | 11504,97 | 38,48 | 540,85 |
| | | 32 | 6238,00 | 35,52 | 511,65 |
| | | 37 | 3539,20 | 32,91 | 483,80 |
| | BQMall | 22 | 28188,42 | 41,63 | 537,70 |
| | | 27 | 17175,91 | 38,61 | 524,32 |
| | | 32 | 10313,74 | 35,39 | 504,42 |
| | | 37 | 5888,06 | 32,12 | 481,50 |
| | PartyScene | 22 | 48665,51 | 41,20 | 628,72 |
| | | 27 | 30755,22 | 36,85 | 622,76 |
| | | 32 | 18624,67 | 32,91 | 588,74 |
| | | 37 | 10399,27 | 29,12 | 543,29 |
| | RaceHorses | 22 | 17235,60 | 42,50 | 539,85 |

| | | | | | |
|---|---|---|---|---|---|
| | | 27 | 10706,56 | 38,85 | 501,24 |
| | | 32 | 6305,39 | 35,08 | 469,92 |
| | | 37 | 3220,69 | 31,26 | 441,63 |
| Class D WQVGA | BasketballPass | 22 | 5013,42 | 43,19 | 131,57 |
| | | 27 | 2978,75 | 39,65 | 125,96 |
| | | 32 | 1687,77 | 36,21 | 122,57 |
| | | 37 | 939,17 | 33,03 | 114,80 |
| | BQSquare | 22 | 13366,35 | 41,52 | 155,39 |
| | | 27 | 8798,88 | 37,11 | 147,73 |
| | | 32 | 5672,67 | 33,36 | 144,39 |
| | | 37 | 3543,79 | 29,74 | 134,73 |
| | BlowingBubbles | 22 | 9695,03 | 41,19 | 152,92 |
| | | 27 | 5724,36 | 37,12 | 149,11 |
| | | 32 | 3154,50 | 33,43 | 136,44 |
| | | 37 | 1634,49 | 30,15 | 124,23 |
| | RaceHorses | 22 | 4922,83 | 42,63 | 142,92 |
| | | 27 | 3023,66 | 38,49 | 132,84 |
| | | 32 | 1707,72 | 34,48 | 123,23 |
| | | 37 | 882,04 | 30,98 | 113,31 |
| Class E 720p | Vidyo1 | 22 | 22082,02 | 44,75 | 1172,89 |
| | | 27 | 13040,47 | 42,39 | 1133,14 |
| | | 32 | 7940,64 | 39,75 | 1090,48 |
| | | 37 | 4839,14 | 36,79 | 1075,82 |
| | Vidyo3 | 22 | 23091,90 | 44,61 | 1137,07 |
| | | 27 | 14189,69 | 42,26 | 1086,91 |
| | | 32 | 8950,69 | 39,54 | 1061,55 |
| | | 37 | 5514,35 | 36,39 | 1041,50 |
| | Vidyo4 | 22 | 24584,93 | 44,31 | 1157,07 |
| | | 27 | 13818,80 | 41,66 | 1092,15 |
| | | 32 | 7933,95 | 39,02 | 1056,84 |
| | | 37 | 4580,00 | 36,32 | 1041,14 |
| Class F | BasketballDrillText | 22 | 21641,26 | 42,13 | 573,32 |
| | | 27 | 12134,26 | 38,72 | 534,46 |
| | | 32 | 6816,69 | 35,68 | 507,13 |
| | | 37 | 3990,29 | 32,91 | 484,98 |
| | ChinaSpeed | 22 | 24592,78 | 44,94 | 1012,58 |
| | | 27 | 16765,00 | 40,99 | 949,96 |
| | | 32 | 11127,36 | 37,16 | 894,74 |
| | | 37 | 7268,98 | 33,49 | 868,93 |
| | SlideEditing | 22 | 39191,39 | 46,19 | 1274,71 |
| | | 27 | 29343,78 | 41,83 | 1258,34 |
| | | 32 | 22008,00 | 37,22 | 1218,61 |
| | | 37 | 15808,12 | 32,42 | 1171,82 |
| | SlideShow | 22 | 4698,74 | 51,18 | 886,93 |
| | | 27 | 3083,76 | 47,59 | 891,33 |
| | | 32 | 2042,94 | 43,95 | 874,06 |
| | | 37 | 1346,23 | 40,27 | 862,98 |

Table 13: Results for the CDR heuristic on High Efficiency condition (target level 64).

| Class | Video | QPISlice | kbps | PSNR | Time (sec) |
|---|---|---|---|---|---|
| Class A<br>4K | Traffic | 22 | 106499,57 | 43,53 | 5656,22 |
| | | 27 | 59987,04 | 40,35 | 5268,60 |
| | | 32 | 34562,57 | 37,30 | 5017,19 |
| | | 37 | 19868,25 | 34,21 | 4761,53 |
| | People on Street | 22 | 109188,83 | 43,44 | 4863,53 |
| | | 27 | 63925,35 | 40,02 | 4581,74 |
| | | 32 | 37202,05 | 36,93 | 4315,86 |
| | | 37 | 22452,05 | 34,05 | 4155,38 |
| Class B<br>1080p | Kimono | 22 | 18328,93 | 43,06 | 2627,51 |
| | | 27 | 10450,19 | 41,76 | 2491,80 |
| | | 32 | 6288,92 | 39,94 | 2381,27 |
| | | 37 | 3762,13 | 37,54 | 2310,38 |
| | ParkScene | 22 | 54105,86 | 41,77 | 2856,02 |
| | | 27 | 29207,83 | 38,73 | 2614,03 |
| | | 32 | 15220,52 | 35,72 | 2431,16 |
| | | 37 | 7562,48 | 32,85 | 2284,66 |
| | Cactus | 22 | 114791,68 | 40,71 | 2978,62 |
| | | 27 | 53477,59 | 37,94 | 2625,98 |
| | | 32 | 29546,10 | 35,67 | 2469,90 |
| | | 37 | 16396,33 | 33,19 | 2372,16 |
| | BasketballDrive | 22 | 64920,62 | 41,20 | 2433,33 |
| | | 27 | 27035,70 | 38,93 | 2310,48 |
| | | 32 | 14313,60 | 37,31 | 2264,45 |
| | | 37 | 8247,70 | 35,42 | 2183,81 |
| | BQTerrace | 22 | 180927,76 | 42,59 | 2330,99 |
| | | 27 | 93143,44 | 37,79 | 2278,95 |
| | | 32 | 51259,80 | 34,77 | 2156,36 |
| | | 37 | 28848,99 | 31,94 | 2026,39 |
| Class C<br>WVGA | BasketballDrill | 22 | 21289,15 | 41,87 | 551,25 |
| | | 27 | 11500,31 | 38,47 | 512,71 |
| | | 32 | 6242,54 | 35,51 | 482,77 |
| | | 37 | 3552,58 | 32,90 | 458,37 |
| | BQMall | 22 | 28206,04 | 41,64 | 509,77 |
| | | 27 | 17177,68 | 38,60 | 492,13 |
| | | 32 | 10323,59 | 35,39 | 478,75 |
| | | 37 | 5904,15 | 32,12 | 458,53 |
| | PartyScene | 22 | 48647,77 | 41,20 | 589,66 |
| | | 27 | 30759,68 | 36,85 | 589,82 |
| | | 32 | 18627,05 | 32,91 | 558,23 |
| | | 37 | 10398,61 | 29,12 | 521,79 |
| | RaceHorses | 22 | 17238,47 | 42,51 | 510,66 |
| | | 27 | 10705,41 | 38,85 | 473,92 |
| | | 32 | 6305,05 | 35,08 | 448,18 |

| | | | | | |
|---|---|---|---|---|---|
| | | 37 | 3222,09 | 31,26 | 419,88 |
| Class D WQVGA | BasketballPass | 22 | 5021,79 | 43,19 | 125,46 |
| | | 27 | 2981,48 | 39,64 | 120,51 |
| | | 32 | 1691,88 | 36,21 | 116,25 |
| | | 37 | 942,52 | 33,03 | 110,14 |
| | BQSquare | 22 | 13372,04 | 41,53 | 149,20 |
| | | 27 | 8800,48 | 37,12 | 141,05 |
| | | 32 | 5674,14 | 33,37 | 136,19 |
| | | 37 | 3543,10 | 29,74 | 127,82 |
| | BlowingBubbles | 22 | 9695,72 | 41,19 | 145,91 |
| | | 27 | 5721,63 | 37,12 | 142,40 |
| | | 32 | 3156,14 | 33,43 | 132,36 |
| | | 37 | 1633,60 | 30,15 | 119,55 |
| | RaceHorses | 22 | 4922,39 | 42,63 | 137,18 |
| | | 27 | 3022,27 | 38,49 | 127,70 |
| | | 32 | 1706,62 | 34,48 | 118,01 |
| | | 37 | 882,76 | 30,99 | 110,01 |
| Class E 720p | Vidyo1 | 22 | 22186,65 | 44,75 | 1107,72 |
| | | 27 | 13122,92 | 42,40 | 1067,49 |
| | | 32 | 8013,36 | 39,75 | 1030,84 |
| | | 37 | 4913,78 | 36,79 | 1012,23 |
| | Vidyo3 | 22 | 23173,58 | 44,62 | 1068,79 |
| | | 27 | 14265,80 | 42,26 | 1027,60 |
| | | 32 | 9019,36 | 39,54 | 1004,94 |
| | | 37 | 5582,09 | 36,39 | 986,68 |
| | Vidyo4 | 22 | 24634,03 | 44,30 | 1092,10 |
| | | 27 | 13878,03 | 41,66 | 1035,21 |
| | | 32 | 8002,34 | 39,02 | 1000,20 |
| | | 37 | 4669,92 | 36,32 | 986,72 |
| Class F | BasketballDrillText | 22 | 21660,09 | 42,14 | 541,36 |
| | | 27 | 12139,23 | 38,71 | 506,09 |
| | | 32 | 6822,38 | 35,67 | 480,61 |
| | | 37 | 3998,58 | 32,90 | 458,12 |
| | ChinaSpeed | 22 | 24599,57 | 44,94 | 933,21 |
| | | 27 | 16767,23 | 40,98 | 884,14 |
| | | 32 | 11138,69 | 37,16 | 847,28 |
| | | 37 | 7283,28 | 33,49 | 815,90 |
| | SlideEditing | 22 | 39182,08 | 46,19 | 1191,42 |
| | | 27 | 29347,83 | 41,83 | 1164,59 |
| | | 32 | 22010,06 | 37,22 | 1138,98 |
| | | 37 | 15812,19 | 32,42 | 1098,83 |
| | SlideShow | 22 | 4713,36 | 51,18 | 836,85 |
| | | 27 | 3096,94 | 47,62 | 835,15 |
| | | 32 | 2049,34 | 43,96 | 825,36 |
| | | 37 | 1356,78 | 40,26 | 810,00 |

# APENDIX B – RESUMO – PORTUGUÊS

**Arquitetura de hardware baseada em algoritmo para a predição intra-quadro do padrão de codificação de vídeo HEVC.**

## B 1. Resumo

Este trabalho apresenta uma arquitetura de hardware para a predição intra-quadro do padrão emergente HEVC de codificação de vídeo. O padrão HEVC está sendo desenvolvido com o objetivo de aumentar em 50% a eficiência de compressão, quando comparado com o H.264/AVC, atual padrão de codificação de vídeo. Para atingir tal objetivo, várias novas ferramentas de codificação foram introduzidas no padrão HEVC, o que causou um aumento considerável na complexidade computacional envolvida no processo de codificação, quando comparado ao H.264/AVC. Analisando somente a os avanços na predição intra (foco deste trabalho), vários novos modos direcionais foram introduzidos no processo de predição intra, além de um maior número de tamanhos de bloco. Nesse contexto, este trabalho propõe uma arquitetura de hardware dedicada para a predição intra baseada em um algoritmo rápido de decisão de modo, também desenvolvido neste trabalho, para acelerar o processo de predição intra. Três algoritmos foram desenvolvidos para acelerar o processo de decisão de modo intra. Os resultados mostraram que é possível reduzir a complexidade computacional do processo de predição intra com pequenas perdas na eficiência de compressão. Com base em um dos algoritmos a arquitetura de hardware dedicada para a predição intra foi desenvolvida. Os resultados mostraram que a arquitetura é capaz de atingir uma alta taxa de processamento quando sintetizada para a tecnologia IBM 0,65um, sendo capaz de processar vídeos de resolução Full HD em tempo real.

## B 2. Introdução

Os últimos avanços da tecnologia baseada em silício tem permitido um grande aumento na capacidade de processamento de dispositivos móveis modernos. Aplicações, como as multimídia, que demandam altas taxas de processamento estão cada vez mais populares, já que os últimos avanços tecnológicos proporcionaram uma grande expansão deste mercado. Telefones celulares, TV digital, computadores portáteis e PDAs estão entre os dispositivos mais populares capazes de receber e reproduzir vídeos digitais de alta resolução. Muitos desses dispositivos podem também capturar e transmitir vídeos digitais em redes de comunicação. Além disso, muitos destes dispositivos que possuem câmeras digitais embarcadas são capazes de codificar e decodificar vídeos de alta definição.

Devido a grande quantidade de dados necessária para representar vídeos digitais, o processo de compressão de vídeos digitais é essencial para o sucesso das aplicações que lidam com este tipo de mídia. Como vídeos digitais estão cada vez mais presentes em todos os tipos de aplicação existe uma demanda crescente por qualidade visual destes vídeos. Portanto, as aplicações multimídia atuais precisam suportar imagens com resolução e qualidade cada vez maiores. Nesse contexto, tanto a academia como a indústria tem investido esforços na pesquisa de novos algoritmos de compressão de vídeo a fim de possibilitar armazenamento e transmissão eficiente de vídeos digitais de alta qualidade e resolução.

Vários padrões de codificação de vídeo foram desenvolvidos nos últimos anos tendo como principal objetivo o aumento da capacidade de compressão. Grupos de especialistas, tais como o VCEG (Video Coding Experts Group) da ITU-T (International Telecommunication Union) e o MPEG (Moving Picture Experts Group) da ISO/IEC (International Organization Standardization) vêm desenvolvendo esse novos padrões de codificação de vídeo. Novas ferramentas e técnicas de compressão vêm sendo desenvolvidas com o objetivo de possibilitar altas taxas de compressão minimizando os impactos na qualidade visual.

O H.264/AVC (JVT, 2003), desenvolvido pelo JVT (Joint Video Team) em 2003, ainda é o padrão estado-da-arte em compressão de vídeo. O principal objetivo do desenvolvimento do H.264/AVC foi aumentar em 50% a taxa de compressão quando comparado com o padrão anterior (MPEG-2) (RICHARDSON, 2003). Para atingir tal objetivo, varias novas ferramentas de algoritmos de compressão de vídeos foram introduzidos no padrão e o H.264/AVC obteve resultados satisfatórios em termos de taxa de compressão. Entretanto, nós últimos anos a demanda por qualidade visual e alta resolução continuou crescendo. Para atender essa demanda, os especialistas em compressão de vídeo continuaram as pesquisas em novos algoritmos com o objetivo de criar um novo padrão capaz de superar o H.264/AVC.

Assim, em Janeiro de 2010 um novo time de especialistas em compressão de vídeos foi formado com o nome de JCT-VC (Joint Collaborative Team on Video Coding) com o objetivo de criar o padrão de compressão de vídeo para atender a nova geração de vídeos digitais (com resolução de até 4K – 4096x2048 pixels) que foi batizado de HEVC (High Efficiency Video Coding). O principal objetivo do novo padrão era novamente superar o padrão atual (H.264/AVC) em 50%.

Embora o HEVC tenha atingido os objetivos em relação a taxa de compressão, a complexidade computacional envolvida no processo de compressão também cresceu. O padrão HEVC incluí novas estruturas de dados, novos algoritmos de compressão e novas opções de compressão quando comparado com o H.264/AVC. Unidades de codificação (CU – Coding Unit) formando quad-trees são as novas estruturas de dados e o número de possibilidades para codificar um bloco aumentou. Assim, o processo de escolha do melhor modo de compressão em codificadores HEVC é muito mais complexo do que era no H.264/AVC.

Focando no processo de predição intra do HEVC (foco deste trabalho), uma unidade de codificação (CU) pode ser codificada considerando quatro tamanhos diferentes (8x8, 16x16, 32x32 e 64x64 pixels). Além disso, cada uma dessas CUs pode ser particionada em unidades de predição (PU) que suportam até 34 diferentes modos de predição intra. A grande quantidade de possibilidades de codificação impõe um grande desafio ao codificador HEVC se todas as direções e tamanhos de bloco forem avaliados pela técnica RDO (rate-distortion optimization) (SULLIVAN e WIEGAND, 1998). Assim, é necessário o desenvolvimento de mecanismos rápidos para a predição intra para permitir a utilização de codificadores HEVC em aplicações que tenha como requisito processamento de vídeos de alta resolução em tempo real.

Na literatura, existem duas classes de trabalhos que tentam lidar com a questão da complexidade da predição intra. A primeira é relacionado com o desenvolvimento de arquiteturas de hardware dedicadas para o processo de predição intra (DINIZ, ZATT, *et* *al.*, 2009). A segunda classe de trabalhos tenta diminuir o número de possibilidades que podem ser avaliadas pela predição intra através de pré-avaliações ou avaliações em tempo de execução. Trabalhos anteriores como (BHARANITHARAN, LIU, *et al.*, 2008) (HUANG, HSIEH, *et al.*, 2005) (HUANG, OU e CHEN, 2010) (PARLAK, ADIBELLI e HAMZAOGLU, 2008) (WANG, WANG, *et al.*, 2007) propõe algoritmos rápidos usando diferentes estratégias para o processo de decisão de modo intra. Entretanto, todos estes trabalhos são relacionados com o padrão H.264/AVC e não podem ser diretamente aplicados ao HEVC, já que o novo padrão possui novas estruturas de dados, mais tamanhos de blocos e modos de codificação intra a serem considerados no processo de decisão de modo.

Nesse contexto, o principal objetivo desta dissertação é desenvolver um mecanismo de decisão rápido para a predição intra do padrão HEVC que integrado a uma arquitetura de hardware reduza a complexidade computacional do processo de predição intra e também, atinja altas taxas de processamento. Neste trabalho nós propomos o desenvolvimento em conjunto dos dois métodos propostos na literatura – arquitetura dedicada e algoritmo rápido de decisão de modo – em uma só solução para a predição intra do padrão de codificação de vídeo HEVC.

## B 3. Predição intra e modo de decisão no padrão HEVC

O padrão HEVC mantém a codificação hibrida baseada em blocos que já era utilizada no H.264/AVC. Um quadro é dividido em uma sequência quadrada de unidades chamadas de *tree blocks*. Cada uma dessas unidades é formada por NxN amostras de luminância e mais duas de crominância. O tamanho dos blocos de crominância varia de acordo com a sub-amostragem utilizada. Na versão atual do HEVC (JCT-VC, 2012), o tamanho máximo permitido para uma *tree block* é 64x64 pixels.

Cada *tree block* é formado por uma ou mais CUs, que são unidades semelhantes aos macroblocos do H.264/AVC. A grande inovação do padrão HEVC é que uma CU pode ter diferentes tamanhos de acordo com a região que está sendo codificada enquanto um macrobloco possuía tamanho fixo de 16x16 pixels. Cada *tree block* pode ser recursivamente dividido em quatro blocos de tamanho igual. Esse processo de divisão em recursão constrói uma estrutura de arvore quaternária (*quad tree*) composta por CUs, que podem ter tamanhos que variam de 64x64 até 8x8. A figura B.3.1 mostra uma possível *quad tree* após consecutivas divisões.

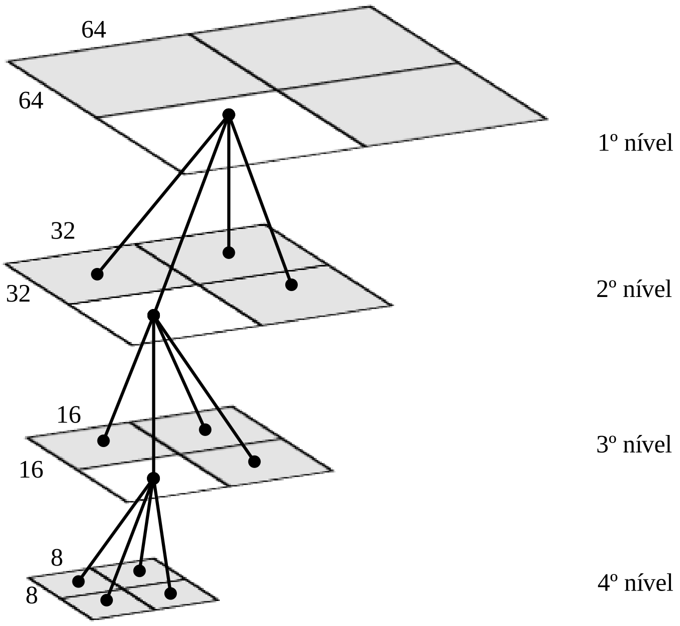


Figura B.3.1: Exemplo de divisão e um *tree block*.

Após a geração da *quad tree*, cada CU pode ser particionada em unidade de predição (PU). A PU pode ser dividida em partes tanto quadradas quanto retangulares. O particionamento das CUs pode ser realizado de acordo com a figura B.3.2. As PUs são as estruturas onde os modos algoritmos de predição serão aplicados para que a codificação dessa região seja realizada. O **N** na figura B.3.2 representa metade do tamanho da PU.

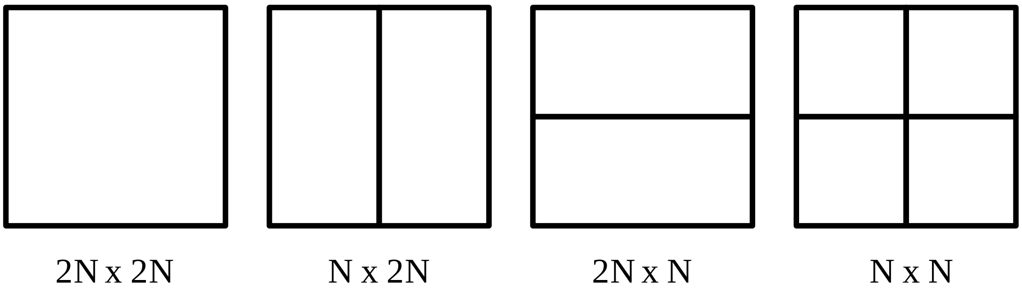


Figura B.3.2: Exemplo de divisão e um *tree block*.

### B.3.1 Predição Intra no HEVC

A predição intra no padrão HEVC é responsável por reduzir ou eliminar a redundância espacial em um quadro. Este tipo de predição foi uma inovação do padrão H.264/AVC, onde foi usada pela primeira vez em esquemas de codificação de vídeo (RICHARDSON, 2003). O processo de predição intra é realizado no domino espacial, onde as amostras vizinhas (já codificadas) ao bloco que está sendo codificado são utilizadas como referência para formar o bloco predito.

No padrão HEVC a predição sobre o bloco atual pode ser realizada de acordo com 33 diferentes modos direcionais, mais o modo DC. A figura B.3.3 mostra os 33 possíveis modos direcionais usando como exemplo um bloco de tamanho 8x8.

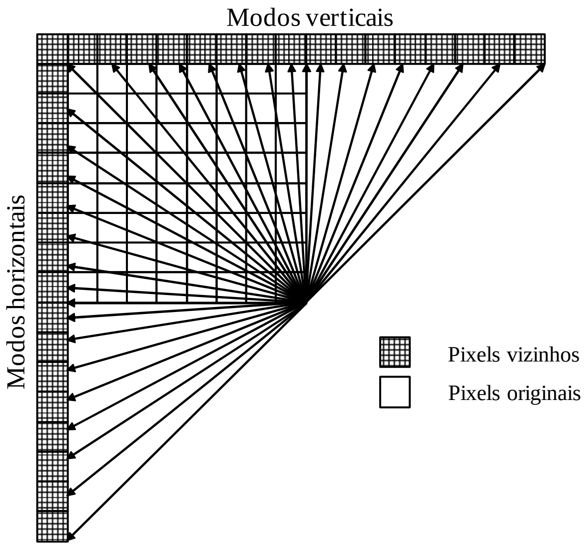


Figura B.3.3: 33 modos direcionais para a predição intra do padrão HEVC.

Apesar dos 33 modos direcionais de predição intra possíveis, o número de modos que é avaliado mudo de acordo com o tamanho da PU que está sendo codificada e de acordo com o número de amostras vizinhas disponíveis. A tabela B.3.1 mostra o número de modos de predição intra para cada tamanho de PU no HEVC em comparação com o H.264/AVC. O modo DC também é considerado.

Tabela B.3.1: Número de modos de predição intra para cada tamanho de PU.

| **Tamanho PU** | **Número de modos Intra HEVC** | **Número de modos Intra H.264/AVC*** |
|---|---|---|
| 64x64 | 3 | - |
| 32x32 | 34 | - |
| 16x16 | 34 | 4 |
| 8x8 | 34 | - |
| 4x4 | 17 | 9 |

*H.264/AVC no perfil *Main*

### B.3.2 Decisão de modo de predição intra no HEVC

O processo de decisão de modo intra implementado na versão do software de teste do HEVC, o HM 5.1 (ISO/IEC-JTC1/SC29/WG11, 2011) segue a ordem de busca em profundidade na estrutura *quad tree.* A avaliação segue a abordagem *top-down*, ou seja, PUs em nodos mais acima na *quad tree* são avaliados primeiro pela técnica RDO. A figura B.3.4 mostra um exemplo da ordem usada no HM para avaliar PUs em cada nodo, os números dentro dos nodos representam a ordem de avaliação dos nodos.

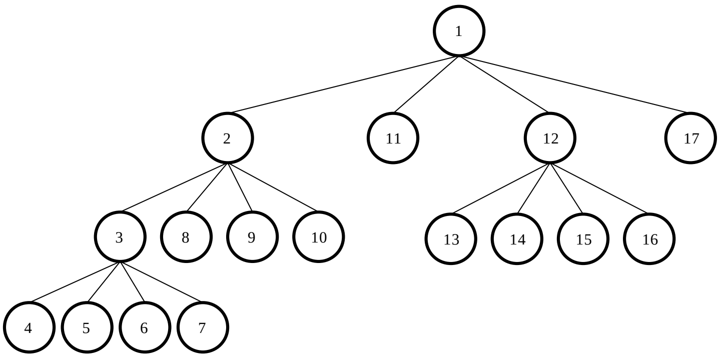


Figura B.3.4: Ordem de avaliação do processo de decisão de modo na *quad tree* no HM 5.1.

Na configuração padrão do software de referência HM 5.1 a avalição RDO completa não é realizada para todos os modos de predição intra, já que isso seria uma tarefa muito complexa para o codificador. De acordo com o tamanho da PU um número limitado de modos intra é completamente avaliado pela técnica RDO. Esta idéia foi inicialmente proposta no trabalho de (ZHAO, ZHANG, *et al.*, 2011) e adotada pela primeira vez na versão 2.0 do HM. A tabela B.3.2 mostra o número de avaliação RDO completas para cada tamanho de PU.

Tabela B.3.2: Numero de modos completamente avaliados pelo RDO (ZHAO, ZHANG, *et al.*, 2011).

| **Tamanho PU** | **Número de modos completamente avaliados RDO** | **Número de modos total** |
|---|---|---|
| 64x64 | 3 | 3 |
| 32x32 | 3 | 34 |
| 16x16 | 3 | 34 |
| 8x8 | 8 | 34 |
| 4x4 | 8 | 17 |

Mesmo com essa simplificação adotada como padrão no software HM, a predição intra ainda é um passo que demanda de alta complexidade computacional em codificadores HEVC, já que a avaliação RDO completa é realizada muitas vezes em uma *quad tree* completa para escolher os melhores tamanhos de CU e PU, além de escolher o modo de predição intra que irá gerar o melhor custo de taxa-distorção do vídeo codificado.

## B 4. Algoritmos rápidos desenvolvidos

A ideia base utilizada neste trabalho para o desenvolvimento dos algoritmos rápidos para a decisão de modo intra explora a correlação de modos presente na estrutura de *quad tree* utilizado no HEVC. Esta correlação está presente nos modos de predição intra escolhidos entre diferentes níveis da *quad tree*, que podem auxiliar na aceleração do processo de decisão. Essa correlação pode ser exemplificada da seguinte maneira: se os modos de predição escolhidos para quatro PUs de tamanho 8x8 (que formam uma PU de tamanho 16x16) forem similares, existe uma grande chance de a PU de tamanho 16x16, que representa a mesma região da imagem, use um modo intra similar.

Sendo assim, uma avaliação para medir o nível desta correlação foi realizada utilizando várias amostras de vídeo, seguindo as condições de teste especificadas pelo

JCT-VC (ISO/IEC-JCT1/SC29/WG11, 2011). As tabelas B.4.1 e B.4.2 mostram os resultados desta avaliação considerando diferentes classes de vídeos (A até E) e os quatro parâmetros de quantização (QP) recomendados, para as duas configurações estruturais (*High Efficiency* e *Low Complexity* respectivamente).

Tabela B.4.1: Correlação entre modos intra de diferentes níveis na *quad tree* (High Efficiency).

| QP | A | B | C | D | E |
|---|---|---|---|---|---|
| 22 | 50,42% | 50,70% | 47,12% | 41,87% | 57,02% |
| 27 | 54,72% | 58,29% | 49,96% | 45,19% | 63,33% |
| 32 | 58,99% | 64,60% | 53,52% | 51,21% | 68,21% |
| 37 | 63,04% | 69,98% | 58,69% | 57,93% | 72,97% |

Tabela B.4.2: Correlação entre modos intra de diferentes níveis na *quad tree* (Low Complexity).

| QP | A | B | C | D | E |
|---|---|---|---|---|---|
| 22 | 47,24% | 47,56% | 45,15% | 40,30% | 55,62% |
| 27 | 52,33% | 55,70% | 48,16% | 43,13% | 62,37% |
| 32 | 57,28% | 62,46% | 51,80% | 49,36% | 67,26% |
| 37 | 62,10% | 68,64% | 58,03% | 55,26% | 71,66% |

Os valores em porcentagem apresentado nas tabelas foram calculados da seguinte forma: se entre o conjunto de melhores modos intra de quatro PUs de tamanho NxN que formam uma PU de tamanho 2Nx2N (nível acima na *quad tree*) está o modo que foi escolhido como melhor para a PU maior, então a correlação existe.

Considerando os resultados apresentados nas tabelas, a correlação entre modos de diferentes níveis na *quad tree* pode chegar a aproximadamente 72%. Entretanto, a existência dessa correlação não é explorada pelo HM para o processo de decisão intra.

Nesse contexto, neste trabalho o uso dessa correlação é proposto para ser usado como base para acelerar o processo de decisão de modo intra no HEVC. Assim, usando os modos escolhidos como melhores em níveis mais abaixo na *quad tree* como referência para a decisão de modo do nível imediatamente acima, é possível diminuir o número de candidatos a serem avaliados pela técnica RDO, acelerando o processo de decisão como um todo.

### B.4.1. Nova ordem de avalição na quad tree

Como mostrado anteriormente, o processo de decisão de modo no software HM realiza uma avalição da *quad tree* seguindo a ordem de busca em profundidade com uma abordagem top-down, ou seja, os nodos mais acima são avaliados antes dos nodos abaixo. Entretanto, essa abordagem não permite que os modos intra escolhidos nos níveis abaixo, sejam utilizados como referência para a decisão do nível imediatamente acima, já que a decisão para o nodo mais acima já foi realizada. Desse modo, neste trabalho, foi utilizada uma nova abordagem na ordem de avaliação da *quad tree*, possibilitando o uso da correlação mencionada na seção anterior para acelerar o processo de predição intra. A figura B.4.1 mostra a ordem de avaliação utilizada neste trabalho. Os números representam a ordem em que os nodos são avaliados.

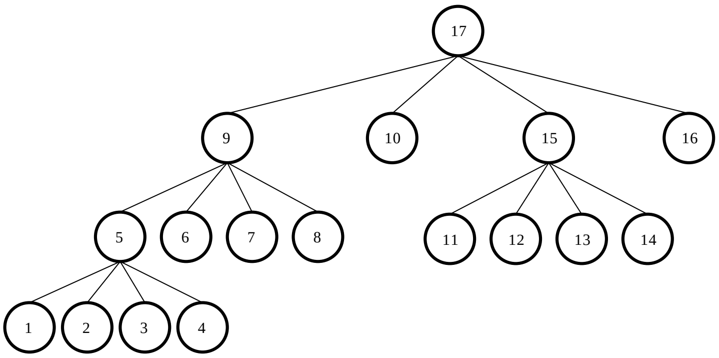


Figura B.4.1: Nova ordem de avaliação na *quad tree* para o processo de decisão intra.

Com essa nova ordem de avaliação na *quad tree* é possível usar os melhores modos intra dos níveis mais abaixo como referência para realizar a decisão do nível imediatamente acima. Considerando o exemplo da figura B.4.1, os modos intra escolhidos nas decisões dos nodos 1, 2, 3 e 4 podem ser utilizados como referência para o processo de decisão de modo para o nodo 5. Utilizando essa estratégia como base, três algoritmos diferentes foram propostos neste trabalho e estão detalhados nas seções seguintes.

### B.4.2. First Decision Reuse (FDR)

O primeiro algoritmo desenvolvido neste trabalho foi baseado na correlação das amostras vizinhas que são utilizadas no processo de predição intra. A figura B.4.2 mostra essa correlação.

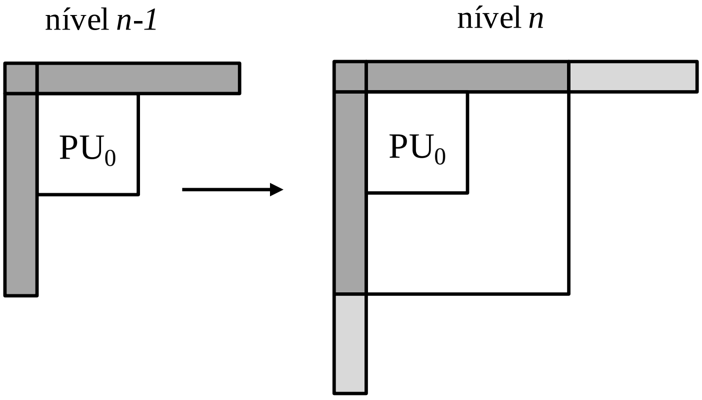


Figura B.4.2: Correlação entre amostras vizinhas utilizadas na predição intra entre diferentes níveis na *quad tree*.

As amostras vizinhas utilizadas para o processo de predição intra da $PU_0$ (nível *n-1*) são um subconjunto das amostras que serão utilizadas para realizar a predição da PU maior (nível *n*). Sendo assim, o primeiro algoritmo, chamado de FDR (*First Decision Reuse*) considera que entre todas as quatro PUs do nível *n-1*, o modo intra da $PU_0$ será utilizado para realizar a codificação da PU maior no nível *n*.

### B.4.3. Majority Decision Reuse (MDR)

O algoritmo MDR (*Majority Decision Reuse*) desenvolvido neste trabalho foi baseado no modo de predição intra mais utilizado nas PUs imediatamente abaixo, ou seja, os modos candidatos a serem avaliados para a predição de uma PU são aqueles que mais ocorrem entre as PUs no nível abaixo na *quad tree*. Assim, a moda estatística dos modos das PUs no nível abaixo é utilizada para definir quais ou qual modo será avaliado pela técnica RDO no nível imediatamente acima. A figura B.4.3 mostra um exemplo onde o conjunto de modos candidatos é formado por dois modos (13 e 0), já que esses são os modos que mais aparecem como melhores para a decisão das PUs no nível imediatamente abaixo.

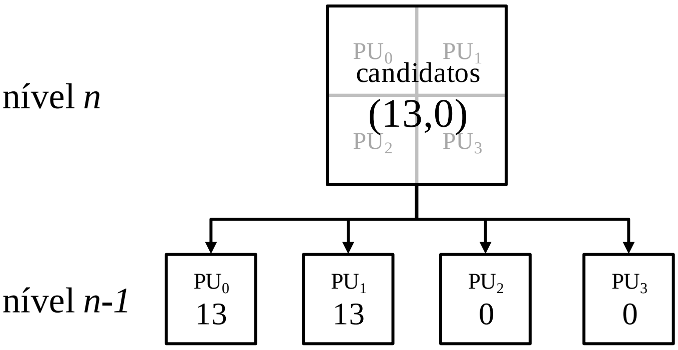


Figura B.4.3: Seleção dos candidatos para a decisão de modo intra baseado na moda estatística.

### B.4.4. Complete Decision Reuse (CDR)

O algoritmo CDR (*Complete Decision Reuse*) que também usa a correlação entre modos em níveis vizinhos na *quad tree* apresentada anteriormente, entretanto, não tem nenhum limitação em relação a quais os modos serão utilizados como referência para formar o conjunto de candidatos que serão completamente avaliados pela técnica RDO. Sendo assim, todos os modos usados nas PUs do nível imediatamente abaixo são utilizados como referência para a decisão de modo intra da PU no nível acima. A figura B.4.4 mostra um exemplo de como o conjunto de modos candidatos é formado.

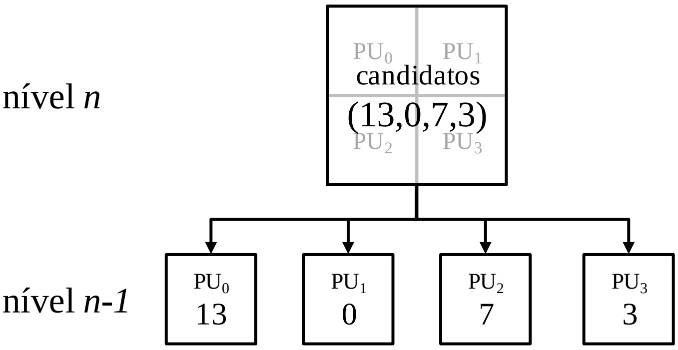


Figura B.4.4: Seleção dos candidatos para a decisão de modo intra baseado no algoritmo CDR.

### B.4.5. Resultados

Os resultados obtidos com os algoritmos propostos foram avaliados no software HM versão 5.1. A avaliação for realizada de acordo com as sequencias de teste e parâmetros de quantização (QP) especificados em (ISO/IEC-JCT1/SC29/WG11, 2011), considerando as duas configurações estruturais (*High Efficiency* e *Low complexity*). A tabela B.4.3 mostra os resultados obtidos considerando a variação (delta **Δ**) de taxa de bits (%), PSNR (dB) e tempo de execução (%) em relação a referência, que é o HM sendo executado com as configurações padrão.

Além disso, todos os algoritmos foram avaliados considerando a variação de um parâmetro proposto neste trabalho, o nível limite de ação do algoritmo que pode ter valores iguais a 8, 16, 32 e 64. Cada valor está associado com o nível limite de ação do algoritmo na estrutura *quad tree*. Por exemplo, se o nível limite é 16, o algoritmo (seja qualquer um dos três) será utilizado para definir o conjunto de modos a serem avaliados nas PUs de tamanho 8x8 e 16x16 e para as PUs acima (32x32 e 64x64) o algoritmos não é considerado e o modo de decisão padrão do HM é utilizado.

Tabela B.4.3: Resultados dos algoritmos FDR, MDR e CDR.

| **Algoritmo** | **Nível limite** | ***High Efficiency*** | | | ***Low Complexity*** | | |
|---|---|---|---|---|---|---|---|
| | | **Δ taxa de bits (%)** | **Δ PSNR (dB)** | **Δ tempo (%)** | **Δ taxa de bits (%)** | **Δ PSNR (dB)** | **Δ tempo (%)** |
| FDR | 8 | +1,69 | -0,05 | -9,52 | +1,67 | -0,08 | -14,48 |
| | 16 | +4,44 | -0,08 | -13,92 | +4,32 | -0,11 | -21,72 |
| | 32 | +6,02 | -0,08 | -18,50 | +5,77 | -0,12 | -28,89 |
| | 64 | +6,99 | -0,08 | -23,10 | +6,62 | -0,11 | -35,80 |
| MDR | 8 | +1,31 | -0,05 | -8,93 | +1,28 | -0,07 | -13,90 |
| | 16 | +2,91 | -0,06 | -13,05 | +2,75 | -0,10 | -20,71 |
| | 32 | +3,76 | -0,07 | -17,27 | +3,51 | -0,10 | -27,31 |
| | 64 | +4,16 | -0,06 | -21,54 | +3,85 | -0,10 | -33,81 |
| CDR | 8 | +1,03 | -0,04 | -8,33 | +1,00 | -0,06 | -12,91 |
| | 16 | +2,10 | -0,05 | -11,85 | +2,00 | -0,07 | -18,78 |
| | 32 | +2,66 | -0,05 | -15,29 | +2,50 | -0,08 | -24,21 |
| | 64 | +2,91 | -0,05 | -18,53 | +2,71 | -0,08 | -29,37 |

## B 5. Arquitetura para predição intra

Nesta seção é apresentada a arquitetura de hardware dedicada para a predição intra. A arquitetura foi desenvolvida utilizando como base o algoritmo CDR (*complete decision reuse*) que foi apresentado anteriormente. A arquitetura é capaz de realizar a predição intra considerando todos os modos direcionais e todos os tamanhos de PU (4x4, 8x8, 16x16, 32x32 e 64x64).

A figura B.5.1 mostra um diagrama em blocos da arquitetura para a predição intra desenvolvida neste trabalho. A arquitetura foi dividida em duas partes (*data path* vertical e *data path* horizontal), assim os modos verticais e horizontais são processados em paralelo. A arquitetura é composta basicamente por três partes principais: predição das amostras, cálculo do resíduo e *buffers* de armazenamento.

A predição das amostras é realizada de acordo com a posição da amostra na PU que está sendo codificada e de acordo com o modo de predição que está sendo processado. Cada *data path* é capaz de processar quatro amostras por ciclo.

O resíduo (diferença entre a PU predita e a PU original) é calculado pelo modulo SAD. Esse módulo realiza o cálculo de quatro amostras originais e preditas por ciclo. Como todos os tamanho de bloco são formados por mais que quatro amostras, o SAD *buffer* é utilizado para armazenar os resultados parciais do cálculo residual.

Os *buffers* de vizinho e das amostras originais são utilizados para manter os dados constantes internamente na arquitetura. Assim, todos os cálculos necessários podem ser realizados e os dados só são carregados das memórias externas quando os dados armazenados internamente não serão mais utilizados.

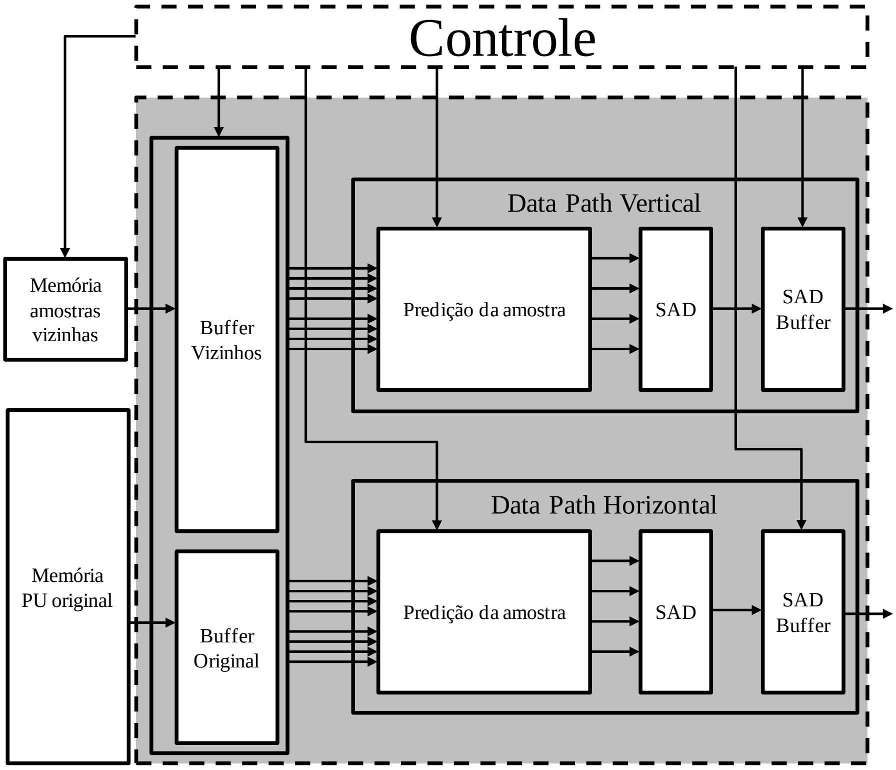


Figura B.5.1: Diagrama em blocos da arquitetura para a predição intra.

### B.5.1. Resultados de síntese e avaliação de desempenho

A arquitetura completa foi descrita na linguagem VHDL e a síntese lógica para a biblioteca de células da tecnologia IBM 65nm foi realizada utilizando a ferramenta RTL Compiler da Cadence. A tabela B.5.1 mostra os resultados de síntese considerando o número de portas equivalentes e frequência máxima de operação atingida.

Tabela B.5.1: Resultados de síntese da arquitetura para a predição intra.

| Technology | IBM 65nm |
|---|---|
| Gate Count | 36,734 |
| Max Frequency (MHz) | 500 |
| PU size supported | All |

A arquitetura para a predição intra foi projetada para suportar processamento em tempo real para vídeos de alta resolução, como HD1080p. Para atingir tal objetivo, a arquitetura necessita ser capaz de processar 15300 *quad trees* por segundo. Esse requisito é dependente do número de ciclos que a arquitetura usa para processar uma *quad tree* e da frequência de operação máxima. Sendo assim, uma avaliação do desempenho da arquitetura foi realizada comparando duas possíveis configurações: (1) sem nenhum tipo de algoritmo rápido de decisão associado, ou seja, realizando a predição intra para todos os modos possíveis, e (2) utilizando o algoritmo CDR (*Complete Decision Reuse*) como base, ou seja, processando a predição intra sobre somente os candidatos selecionados pelo algoritmo. A tabela B.5.2 apresenta os resultados desta comparação considerando diferentes resoluções de vídeos (VGA, HD720p, HD1080p e 2Kx1K), a taxa de processamento atingida por cada configuração (*quad trees* por segundo), número de quadros por segundo (a 500 MHz), além do frequência mínima necessária para atingir processamento em tempo real para cada uma das resoluções apresentadas.

Tabela B.5.2: Avaliação de desempenho da arquitetura.

| Resolução do vídeo | Arquitetura sem algoritmo rápido | | | Arquitetura baseada no CDR | | |
|---|---|---|---|---|---|---|
| | *quadtrees*/s (500 MHz) | Quadros/s (500 MHz) | Min. Freq. 30 quadros/s (MHz) | *quadtrees*/s (500 MHz) | Quadros/s (500 MHz) | Min. Freq. 30 quadros/s (MHz) |
| 2Kx1K (2048x1024) | 6.785 | 13 | 1.131,75 | 18.115 | 35 | 423,9 |
| HD1080p (1920x1088) | | 13 | 1.127,33 | | 35 | 422,3 |
| HD720p (1280x720) | | 30 | 497,35 | | 80 | 186,3 |
| VGA (640x480) | | 90 | 165,78 | | 241 | 62,1 |

A tabela B.5.2 mostra que a arquitetura para a predição intra desenvolvida neste trabalho atingiu o requisito de processar vídeos de alta resolução em tempo real, sendo capaz de processar até 35 quadros por segundo quando a frequência de operação é 500 MHz (máxima frequência de operação atingida quando sintetizada para a tecnologia IBM 65 nm). Outro resultado importante demonstrado na tabela é que a arquitetura utilizando o algoritmo CDR como base atinge uma taxa de processamento aproximadamente três vezes maior do que uma arquitetura sem nenhum algoritmo rápido. Com o algoritmo CDR diminuindo o número de modos a serem avaliados, a arquitetura é capaz de processar tanto vídeos de resolução HD1080p como 2Kx1K à 30 quadros por segundo, o que não é possível quando comparado com a arquitetura sem nenhum algoritmo associado, que é capaz de processar somente 13 quadros por segundo nas resoluções mais altas.

## B 6. Conclusões

Este trabalho apresentou uma arquitetura de hardware dedicada para a predição intra do novo padrão de codificação de vídeo HEVC. O principal objetivo deste trabalho foi desenvolver uma arquitetura de hardware para a predição intra com base em um algoritmo rápido de decisão de modo intra capaz de aumentar desempenho da predição intra em codificadores HEVC. As principais contribuições deste trabalho foram:

- Três algoritmos rápidos de decisão de modo foram desenvolvidos: FDR, MDR e CDR. Os resultados mostraram que os algoritmos podem atingir uma redução de até 35% na complexidade computacional envolvida na codificação intra com baixas penalidades no desempenho de compressão (taxa de bits e qualidade visual).
- Uma arquitetura de hardware capaz de processar todos os tamanhos de PU e todos os modos de predição intra direcionais. A arquitetura foi desenvolvida com base no algoritmo rápido CDR (*Complete Decision Reuse*) para possibilitar o processamento de vídeos de alta resolução em tempo real.

Os resultados de avaliação de desempenho mostraram que a arquitetura baseada no algoritmo CDR possibilitou um aumento de quase três vezes (3X) na taxa de processamento, quando comparado com uma arquitetura sem algoritmo associado. O aumento na taxa de processamento foi obtido com pequenas penalidades na taxa de bits (2,71%) e na qualidade visual (0,08 dB) do vídeo codificado quando comparado a solução sem algoritmo. Além disso, os resultados também mostraram que a arquitetura baseada no CDR é capaz de processar vídeos de resoluções como HD1080p e 2Kx1K em tempo real.

## B 7. Referências

BHARANITHARAN, K. et al. A Low Complexity Detection of Discrete Cross Differences for Fast H.264/AVC Intra Prediction. **IEEE Transactions on Multimedia**, v. 10, p. 1250-1260, Novembro 2008.

DINIZ, C. et al. **A Real Time H.264/AVC Intra frame Prediction Hardware Architecture for HDTV 1080p**. IEEE International Conference on Multimedia and Expo. [S.l.]: [s.n.]. 2009. p. 1138-1141.

GHANBARI, M. Standard Codecs: Image Compression to Advanced Video Coding. **IEEE Press**, 2003.

HUANG, Y. et al. Analysis, Fast Algorithm, and VLSI Architecture Design for H.264/AVC Intra Frame Coder. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 15, p. 378-401, Março 2005.

HUANG, Y.; OU, T.; CHEN, H. Fast Decision of Block Size, Prediction Mode, and Intra Block for H.264 Intra Prediction. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 20, p. 1122-1132, Agosto 2010.

ISO/IEC-JCT1/SC29/WG11. **Common test conditions and software reference**. Daegu, Korea. 2011.

ISO/IEC-JTC1/SC29/WG11. **High Efficiency Video Coding (HEVC) Test Model 3 Encoder Description**. [S.l.]. 2011.

JCT-VC. High Efficiency Video Coding, 2012. Disponivel em: <http://hevc.hhi.fraunhofer.de/>. Acesso em: Outubro 2012.

JCT-VC. **High Efficiency Video Coding (HEVC) text specification draft 8**. Stockholm. 2012.

JM. H.264/AVC Software Coordination, 2012. Disponivel em: <http://iphome.hhi.de/suehring/tml/>. Acesso em: Outubro 2012.

JVT. **Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264/AVC or ISO/IEC 14496-10 AVC)**. [S.l.]. 2003.

LI, F.; SHI, G. **An Efficient VLSIArchitecture for 4x4 Intra Prediction in the HighEfficiency Video Coding (HEVC) Standard**. IEEE International Conference on Image Processing. [S.l.]: [s.n.]. 2011. p. 381-384.

MIN, J. H. **Unification of the directional intra prediction**. [S.l.]. 2010.

PALOMINO, D. et al. **A Memory Aware and Multiplierless VLSI Architecture for the Complete Intra Prediction of the HEVC Emerging Standard**. IEEE International Conference on Image Processing. Orlando: [s.n.]. 2012.

PARLAK, M.; ADIBELLI, Y.; HAMZAOGLU, I. A Novel Computational Complexity and Power Reduction Technique for H.264 Intra Prediction. **IEEE Transactions on Consumer Electronics**, v. 54, p. 2006-2014, Novembro 2008.

RICHARDSON, Y. **H.264/AVC and MPEG-4 Video Compression - Video Coding for Next-Generation Multimedia**. [S.l.]: [s.n.], 2003.

SULLIVAN, G.; WIEGAND, T. Rate-distortion optimization for Video Compression. **IEEE Signal Processing Magazine**, v. 15, p. 74-90, 1998.

SUNNA, P. **AVC/H.264 - an advanced video coding system for SD and HD broadcasting**. Ebu Technical Review. [S.l.]: [s.n.]. 2005.

WANG, J.-C. et al. A Fast Mode Decision Algorithm and Its VLSI Design for H.264/AVC Intra-Prediction. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 17, p. 1414-1422, Outubro 2007.

WIEGAND, T. Overview of the H.264/AVC Video Coding Standard. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 13, p. 560-576, July 2003.

ZHAO, L. et al. **Fast Mode Decision Algorithm for Intra Prediction in HEVC**. IEEE Visual Communications and Image Processing. Taiwan: [s.n.]. 2011.