博士論文

Real-time super high resolution video encoder VLSI architecture and its power reduction

（超高解像度映像のリアルタイム符号化に向けた VLSI アーキテクチャとその低電力化手法）

大西　隆之

Abstract

The use of ultra-high definition video is steadily expanding, such as the 4K / 8K satellite broadcast launch in December 2018, 4K support for various video distribution services, and the rapid increase in the number of 4K TVs for consumers. However, there are many technical issues to achieve super-high-resolution real-time encoders, because of the increasing pixel rate of 8x and 32x compared to current HDTV and because of the recent video coding standard such as H.265/HEVC which requires a huge amount of computation load in exchange for high compression efficiency. Furthermore, the increase in power consumption makes it difficult to apply to the mobile field such as remote news gathering and mobile camcorders, and the reduction of the encoder circuit size and power consumption is also urgently needed.

This dissertation aims to establish a real-time ultra-high definition video coding hardware architecture for solving the above-mentioned issues, in order to develop real-time encoder VLSIs that achieve both high image quality and high compression ratio for broadcast and distribution industries, by reducing computational complexity efficiently while maintaining coding quality.

In order to achieve the above goal, this dissertation focuses on the following three points and makes proposals for solution.

The first is the configuration of a prediction core that determines prediction modes for encoding. The main cause of the computational complexity is that the number of combinations of motion vectors and coding mode candidates increases explosively with the recent coding standards. Therefore, the correlation between the candidates that are spatially and temporally adjacent or included is utilized. By adaptive switching of motion estimation and pruning of prediction modes based on statistical analysis, and sharing of mode evaluation results over multiple block sizes, the amount of operation required for evaluation is greatly reduced while securing wide search range and diversity of coding modes.

Furthermore, in order to respond to the heavy requests of reference image transfer from the motion estimation engines operating in parallel, a reference image cache provided with a high-speed transfer bus is installed to process transfer requests in a time division manner. In the final mode decision, high-speed sequential operation conforming to the procedure of video coding standards is utilized to prevent coding efficiency degradation, and also to enable flexible re-arrangement of processor-controlled operations.

H.264/AVC and H.265/HEVC real-time encoder VLSIs which apply the prediction core

architecture with the above-mentioned features are designed and manufactured in 90 nm and 28 nm CMOS processes, and are installed in encoder devices for professional users in broadcast and distribution industries. Quality evaluation of HDTV / 4K / 8K coded image clarifies that these VLSI achieves high coding efficiency while realizing real-time encoding of ultra-high definition videos.

The second is a parallel coding configuration for higher resolution and multiple channels. Parallel and collaborative processing with multiple VLSIs is an essential function for large-scale applications such as 8K and digital cinema that are not practical to be processed in one chip. Therefore, inter-chip image exchange mechanism for realizing parallel coding by screen division is provided within the VLSI, and a multiplexing unit that outputs a coded stream is also distributed that allows flexible cooperation of multiple multiplexers according to various parallel encoding applications. The MPEG-2 and H.265/HEVC real-time encoder VLSIs with these mechanisms are designed and manufactured in 0.18 μm and 28 nm CMOS processes, and they are adopted in ultra-high definition video encoder devices. Analysis of output streams reveals that these VLSIs can generate high quality transport streams while performing chip-to-chip data transfer in real time.

The third is circuit scale and power reduction technology for motion estimation engines to further reduce the power consumption of the real-time encoder VLSIs. For example, in the H.265/HEVC video encoding VLSI, the functional blocks that perform motion estimation (motion search) consumes over 60% of the energy consumption in the entire prediction core. Power efficiency improvement here practically contributes to the power reduction of the entire VLSIs. However, since the reduction in calculation accuracy and in motion search range directly lead to image quality degradation, it is not suitable for encoder VLSIs that require high image quality for broadcast and distribution use.

Therefore, in the sum of absolute difference (SAD) calculation which occupies most of the motion estimation calculation, an adaptive bit reduction SAD calculation method is proposed in which the bit extraction position is made variable according to the flatness of the image. In a flat area where the MSB-side bits of the pixel luminance values are uniform, the bits on the LSB-side are selectively extracted to perform the SAD operation, and in the other areas, the MSB side bits are extracted to perform the SAD operation.

The proposed method is implemented in the H.265/HEVC real-time encoder VLSI design file, and as a result of performing power simulation in 28 nm CMOS process, it is clarified that the energy consumption of each functional block can be reduced by 18 to 39%. In addition, it was clarified in the HDTV/4K/8K encoding image quality evaluation, including high dynamic range (HDR) videos, that are spreading in recent years, that high quality encoding is possible by suppressing degradation of coding performance by the adaptive bit reduction technique.

With the above-mentioned proposed technologies, this dissertation demonstrates that real-time encoding of 4K/8K ultra-high definition video is realized with the video coding VLSIs, and indicates the possibility of expanding the application to mobile applications by further reducing power consumption.

Also, in the next-generation video coding standards such as versatile video coding (VVC) which are currently under standardization, technical policies stay that they improve compression efficiency by further increasing coding mode candidates under the same processing flow. Therefore, the techniques and ideas proposed in this dissertation are equally applicable to real-time encoder VLSI design for the next generation standards.

Acknowledgments

There are numerous people who contributed to making this dissertation materialize. It is a great pleasure to acknowledge the encouragement and support that I have received from these people.

First, I would like to express my deepest gratitude to my committee chair, Professor Hiroshi Nakamura, for leading the very late writing, slow contact and slow study student for as long as nine years with quite accurate, effective and highly suggestive research instructions. Without your tutelage and clemency, I would not be able to execute and continue the research activities resulting in this dissertation. I would also like to express my gratitude to my committee members, Professor Masahiko Inami, Professor Hitoshi Aida, Professor Masahiro Fujita and Associate Professor Masaaki Kondo, for making a lot of fruitful and sound advices.

I would also like to thank Professor Emeritus Mitsutoshi Hatori and Professor Kiyoharu Aizawa for starting my career in image engineering. I am also very grateful to Professor Emeritus Tadao Saito and Professor Hitoshi Aida again for adding my expertise in video coding and telecommunication. I would like to thank all other members I studied with at Hatori-Aizawa laboratory and Saito-Aida laboratory. I was so lucky to be surrounded by excellent researchers at the very first time of my career. And I would appreciate all the members at Nakamura-Kondo laboratory during my doctoral course, who have encouraged me with lots of intriguing discussions during the lab meetings. These times have given me the primal joy toward a new journey of knowledge, even in my middle age.

I would also like to express my appreciation to managers at Nippon Telegraph and Telephone Corporation (NTT) Cyber Space Laboratories and Media Intelligence Laboratories, Prof. Takeshi Ogura now with Ritsumeikan University, Mr. Makoto Endo now at NTT Electronics, Prof. Jiro Naganuma now with Shikoku University, Prof. Yoshiyuki Yashima now with Chiba Institute of Technology, Prof. Kazuto Kamikura now with Tokyo Polytechnic University, Mr. Atsushi Shimizu now at NTT TechnoCross, for guiding my research. I would also like to thank my supervisors and colleagues at NTT Cyber Space Laboratories and Media Intelligence Laboratories, especially Dr. Hiroe Iwasaki, for her continuing care and support for my research, schooling and career decision-making before and during my doctoral course. I would also like to thank Dr. Mitsuo Ikeda, Dr. Koyo Nitta and Mr. Ken Nakamura for their helpful support and valuable discussion. Your advices on both research as well as on my career were priceless. And a special thanks to my colleagues, Mr. Takuro Takahashi, Mr. Yasuhiko Sato and other members at NTT Electronics. Without their prominent skills in hardware design, some works presented here could not be accomplished.

Finally, I would like to thank my beloved wife Akiko, more than words, not only as my spouse but also as my age peer at NTT laboratories who has continuously encouraged me never to give up my doctoral course. And let me also thank my daughter Mahoro for her sincere love, patience and support. Without encouragement from you two, I would not have been to accomplish the researches presented here.

Takayuki Onishi

Tokyo, August 2019.

Contents

List of Figures

List of Tables

List of Acronyms

4K	4000 (horizontal pixels, precisely 3840 horizontal pixels)
8K	8000 (horizontal pixels, precisely 7680 horizontal pixels)
AI	Artificial Intelligence
AVC	Advanced Video Coding
CABAC	Context Adaptive Binary Arithmetic coding
CIF	Common Intermediate Format
CPU	Central Processing Unit
CTU	Coding Tree Unit
CU	Coding Unit
DC	Direct Current
DCT	Discrete Cosine Transform
DDR-SDRAM	Double Data Rate SDRAM
DEMUX	Demultiplexer
DF	Deblocking Filter
DNN	Deep Neural Network
DRAM	Dynamic Random Access Memory
DSCQS	Double Stimulus Continuous Quality Scale
DVD	Digital Versatile Disc
ES	Elementary Stream
FCBGA	Flip Chip Ball Grid Array
FME	Fractional Motion Estimation or Full-pel Motion Estimation
FPS	Frames Per Second
FPU	Field Pick-up Unit
FTTH	Fiber-To-The-Home
GAN	Generative Adversarial Network

GOP	Group of Pictures
HDR	High Dynamic Range
HDTV	High Definition Television
HEVC	High Efficiency Video Coding
IFE	Image Feature Extraction
IIM	Intra Inter Mode decision
IME	Integer Motion Estimation
IPD	Intra Prediction
ISO/IEC	International Organization for Standardization / International Electrotechnical Commission
ISSCC	International Conference on Solid-State Circuits
ITU-R	International Telecommunication Union - Radiocommunication Standardization Sector
ITU-T	International Telecommunication Union - Telecommunication Standardization Sector
JM	Joint Model
LSB	Least Significant Bit
MBAFF	Macroblock Adaptive Frame/Field
MBUS	Memory Bus
MED	Multi-block-size Edge Detector
MME	Multi-block size Motion Estimation
MPEG	Moving Picture Experts Group
MSB	Most Significant Bit
MUX	Multiplexer
MC	Motion Compensation
ME	Motion Estimation
MV	Motion Vector

NTT	Nippon Telegraph and Telephone corporation
PAFF	Picture Adaptive Frame/Field
PES	Packetized Elementary Stream
PCR	Program Clock Reference
PID	Program Identification
PSI	Program Specific Information
ES	Elementary Stream
PMV	Prediction Motion Vector
PSNR	Peak Signal-to-noise Ratio
QoS	Quality of Service
RISC	Reduced Instruction Set Computing
SAD	Sum of Absolute Difference
SAO	Sample Adaptive Offset
SATD	Sum of Absolute Transformed Difference
SDR	Standard Dynamic Range
SDRAM	Synchronous DRAM
SDTV	Standard Definition Television
SIMD	Single Instruction Multiple Data
SNR	Signal to Noise Ratio
SOP	Structure of Pictures
SRAM	Static Random-Access Memory
STC	System Time Clock
SU	Search Unit
TME	Twice-pel motion estimation
TQ	Transform and Quantization
TS	Transport Stream
VIF	Video Interface

VLSI	Very Large Scale Integrated circuit
VoD	Video on Demand
VR	Virtual Reality
VVC	Versatile Video Coding
WCG	Wide Color Gamut
WME	Wide-range Motion Estimation
WP	Weighted Prediction
WPP	Wave-front Parallel Processing

Chapter 1 Introduction

1.1 Demands and requirements for digital video encoding

Digital video and its compression technology have been dramatically improved over the last three decades, interrelatedly. In order to shoot, edit, distribute, and deliver high resolution, smooth and realistic digital video, advancement of camera sensors and display panels required for shooting and display is not sufficient. Compression techniques for storing and transmitting digital video with a realistic amount of information is quite essential for the spread of high definition videos.

Digital video compression technologies originated from the world of video telephony, which is well known as H.261 [1] standardized by ITU in 1990. At that time the maximum picture size was 352×288 pixels common Intermediate Format (CIF) with up to 15 frames per second, but they soon became applied to the digital TV world and since then digital video coding standards and various codec appliances complied with them contribute to the growth of digital video world today. The basic video compression idea of motion compensation and DCT-based transform has not been changed since H.261, however, as more computing power is allowed with the progress of VLSI and processor technologies, more complex coding tools which had been rejected due to complexity issues have been added for more compression efficiency. Each standard was aimed to achieve double the coding efficiency (i.e. half the bit rate for the same picture signal to noise ratio (SNR)) compared to the previous standard, and the advancements in coding standards have allowed digital video to adopt higher picture size, from standard definition TV (SDTV) of 720×480 pixel to high definition TV (HDTV) of 1920×1080 pixel. Now the latest coding standard HEVC (high efficiency video coding) [2] now has the potential to encode 4K (3840×2160 pixel) and 8K (7680×4320 pixel) programs with 60 frames per second into reasonable bit rates which can be transferred via broadcasting and IP networks.

However, when developing video coding appliances complying with these standards, especially real-time encoders for high definition images, there are many challenges to be solved. One of them is a hardware architecture to achieve real-time processing. Software-based reference encoders used for standardization puts emphasis heavily on maximizing coding quality, and it is not realistic to employ the same algorithms to hardware-based real-time encoders, due to limited computational resources and narrow external memory bandwidths. Real-time encoder architecture should be configured so as to effectively reduce the computational complexity while maintaining good picture quality which could meet the high demand for professional broadcasting services. Another very promising approach for real-time encoding of high definition video is parallel processing of multiple VLSI encoders. The architecture should well take care of its multi-chip configuration for higher resolution encoding or multiplexed encoding of more video channels, ideally without any peripheral devices. And even when the above-mentioned two requirements are fulfilled, deeper circuit scale and power reduction should also be sought, because nowadays power-aware mobile devices even have super-high-resolution video capturing cameras and transmitting applications for social media and over-the-top video services.

This dissertation deals with these three major issues with current real-time super-high-resolution encoders and proposes various techniques to overcome each problem, concluding in commercial encoder VLSIs widely going into market and power simulations based on these VLSIs.

At first, in the following section, history and progress in recent video coding standards are addressed and then their characteristics in computational complexity are described.

1.2 Progress in video coding standards

Figure 1.1 describes the history of recent video coding standards and their applications, mainly for broadcast and distribution in Japan. After the H.261 for videophones, the first standard provided by the ISO/IEC moving pictures experts group was MPEG-1 [3], which is mainly used for package contents with video-CD or digital video files with PCs, not for broadcasting or distribution purposes. The first success in digital broadcasting was MPEG-2 [4], also known as H.262 in ITU-T, which is in Japan first employed for digital HDTV

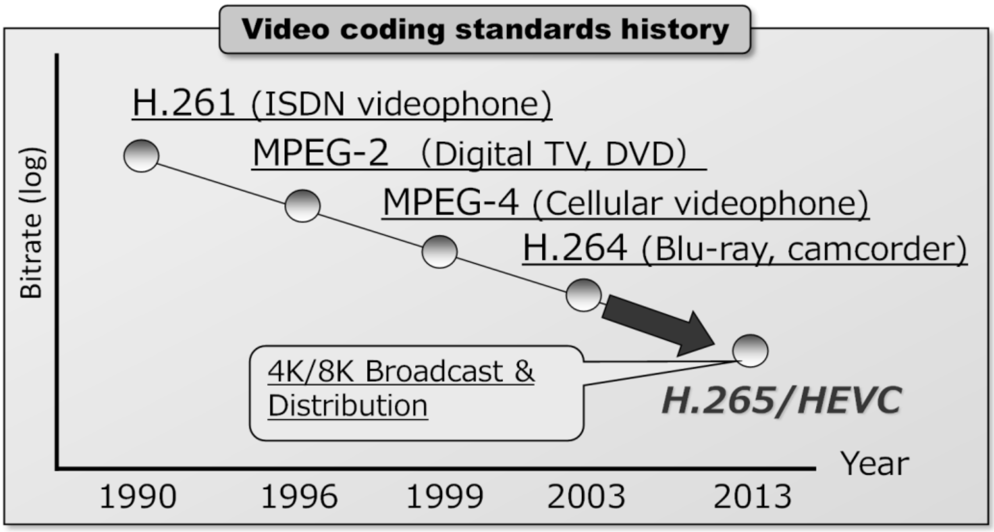


Figure 1.1 History of video coding standards.

broadcasting satellite in the year 2000. It was also employed in digital terrestrial broadcasting with HDTV in 2003 and has become one of the most successful video coding standards nowadays, also with packaged contents provided with DVDs. In the meanwhile H.263 [5] and MPEG-4 [6] are standardized and partly used for video compression for digital cameras and mobile phones. However, they did not reach a big epidemic.

The succeeding video coding standard AVC (advanced video coding) [7], also known as H.264 in ITU-T, becomes quite popular with HDTV package contents provided with Blu-ray discs and both professional / consumer-use HDTV camcorders. The H.264 standard was, however, widely adopted for optical IP network distribution, among them the typical service is a digital HDTV terrestrial broadcast retransmission launched in May 2008.

And the latest coding standard HEVC [8], also known as H.265 in ITU-T, is at the forefront of super-high-resolution 4K and 8K video broadcasting. 4K and 8K commercial satellite broadcasts in Japan started in December 2018. Although it has licensing issues and threats of other over-the-top company-based video codecs such as VP9 format by Google and AV1 format by the Alliance for Open Media, however, it still remains the most promising standard of prevailing super-high-resolution 4K and 8K contents.

Currently the standardization of the next promising coding standard of VVC (versatile

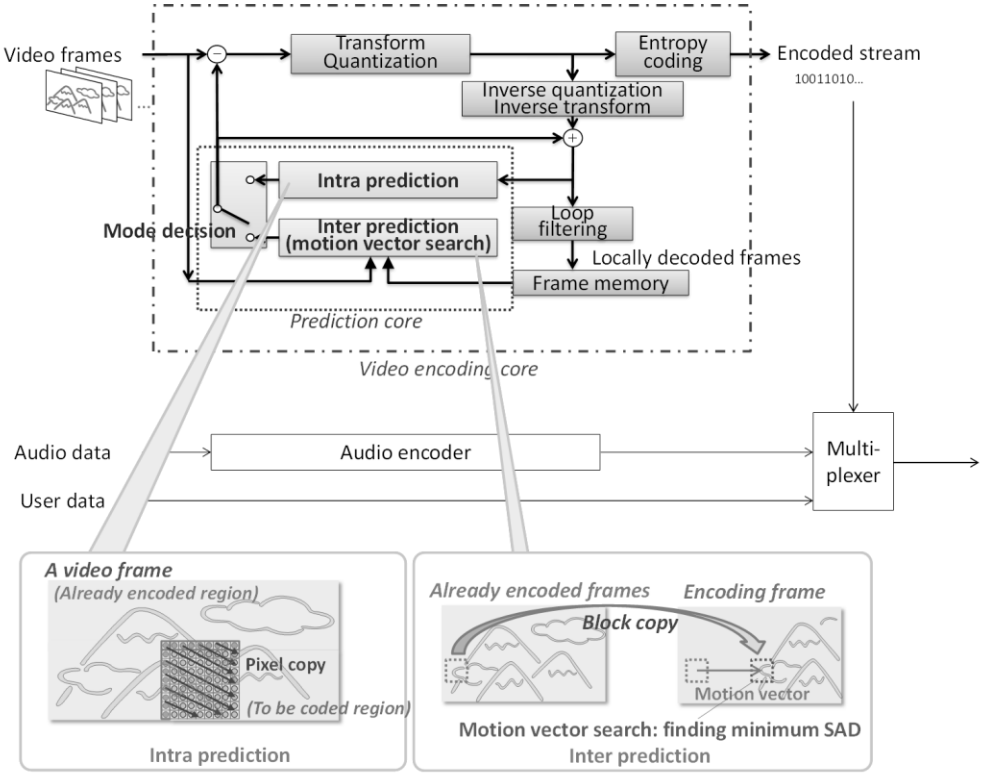


Figure 1.2 Basic structure of video encoders.

video coding) is underway with ITU-T and ISO/IEC, which is planned to be fixed in the year 2020.

1.3 Video coding structure

Figure 1.2 shows the basic structure of video encoders. Video coding standards have been progressed, however, this basic structure has not been changed as a fundamental process of video encoding.

Video compression is fundamentally achieved by “prediction” of pixel luminance and chrominance values, which means, luminance/chrominance value of one pixel is derived from the value of other spatially / temporarily surrounding pixels. When the prediction is more precise and the difference between the predicted value and the actual value is smaller, less coding bits are needed for expressing the difference and coding efficiency gets higher.

The coding standards, therefore, aim to make the prediction more precise.

Intra prediction is a prediction within a frame (the same picture), which means, luminance/chrominance value of one pixel is derived from spatially surrounding pixels which are previously encoded. On the other hand, inter prediction or inter-frame prediction predicts pixel values in certain size of square or rectangle blocks from blocks of temporarily distant picture frames which are previously encoded, with "motion vectors" which denote a motion of objects in the picture frames. Conventionally, the word "P-picture" is used for pictures which use temporarily past pictures for reference, and the word "B-picture" is used for pictures with bi-directional prediction from temporarily past and future reference pictures[1].

Both intra and inter prediction has many prediction modes with multiple combination of motion vectors, the most promising prediction modes (which means, a prediction mode that seems to achieve the least coding bits with the same PSNR) must be chosen. This process is called "mode decision" function in the dissertation.

When prediction is done, at first, accurate predicted image with the decided prediction mode is constructed as motion compensation (MC) process[2]. After that, the differences between the predicted values and the actual pixel values are treated as "residuals of the prediction" and need to be encoded together with the prediction modes which are used. These pixels residual values are transformed with the discrete cosine transform (DCT) or near-DCT integer transform. Then the values in the frequency domain are quantized for effective compression, using the phenomenon that human eyes are sensitive to low frequency signals, not to high frequency signals. Bit rate control of the encoder is basically done here, by changing the amount of quantization. Quantized residuals together with the information of coding modes are further compressed by entropy coding, where Huffman coding has been utilized historically and the new context adaptive arithmetic binary coding

[1] Reference pictures denote the previously encoded pictures which are used as source pictures of block copy, with motion vectors.

[2] When accurate predicted image is obtained during the intra and inter-frame prediction process, this MC process is not required and therefore not shown in Figure 1.2. However, usually during the prediction process, inaccurate predicted image is usually used due to computational complexity reduction and memory bandwidth reduction, or predicted images during the prediction process are discarded due to limited buffer memory. In these cases, MC process is required after the prediction process.

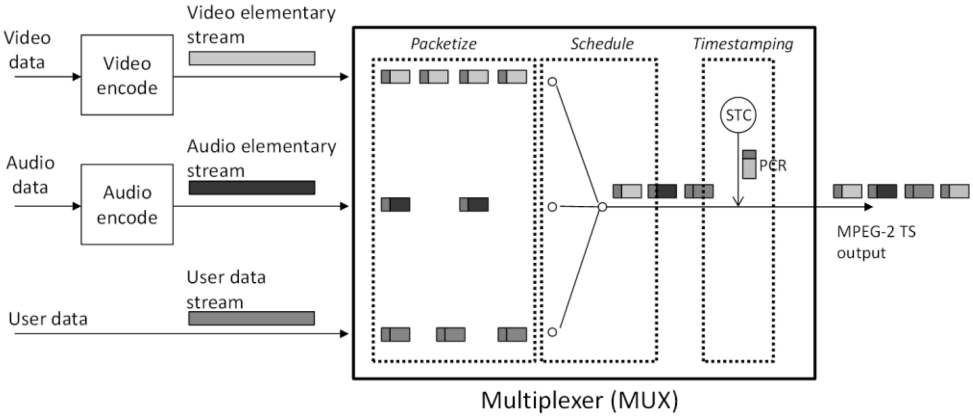


Figure 1.3 Operations of a multiplexer.

(CABAC) has been introduced since H.264/AVC. Entropy-coded data are output as an encoded video stream.

In parallel, the quantized residuals are inverse quantized and inverse transformed to create decoded frames, which are identical to decoded frames at the decoder's side, and are stored in frame memories to be used for the next prediction process. In-loop filtering is inserted from H.264/AVC for reducing artifacts caused by quantized frequency information.

Finally, encoded streams are multiplexed with audio encoded data and other user data into one MPEG-2 transport stream [9] to be transmitted outside of the encoder. The operations of a multiplexer are illustrated in Figure 1.3, where video, audio and user data elementary streams are packetized into 188 byte transport packets, scheduled to mix into one transport stream, timestamped in program clock reference (PCR) fields with an internal system time clock (STC) and then sent out.

1.4 Increasing coding complexity

Figure 1.4 shows the intra (within each picture frame) encoding and inter-frame (among multiple picture frames) encoding modes of each video coding standard. As for intra encoding, MPEG-2 only predicts a DC value, which means the average of each block's luminance values. Other AC values, which mean a transition element of pixel luminance

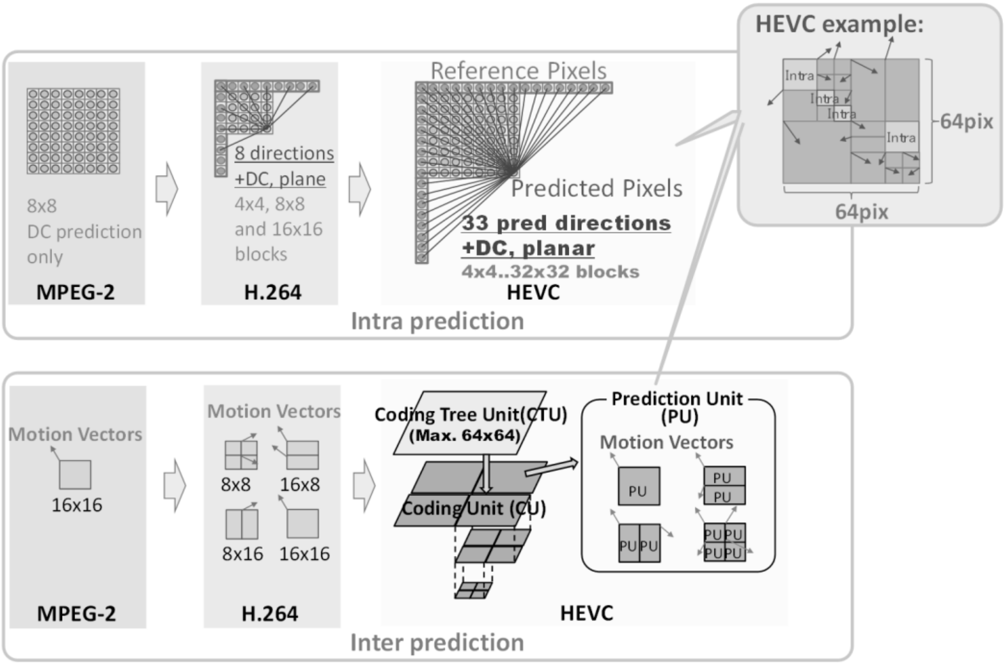


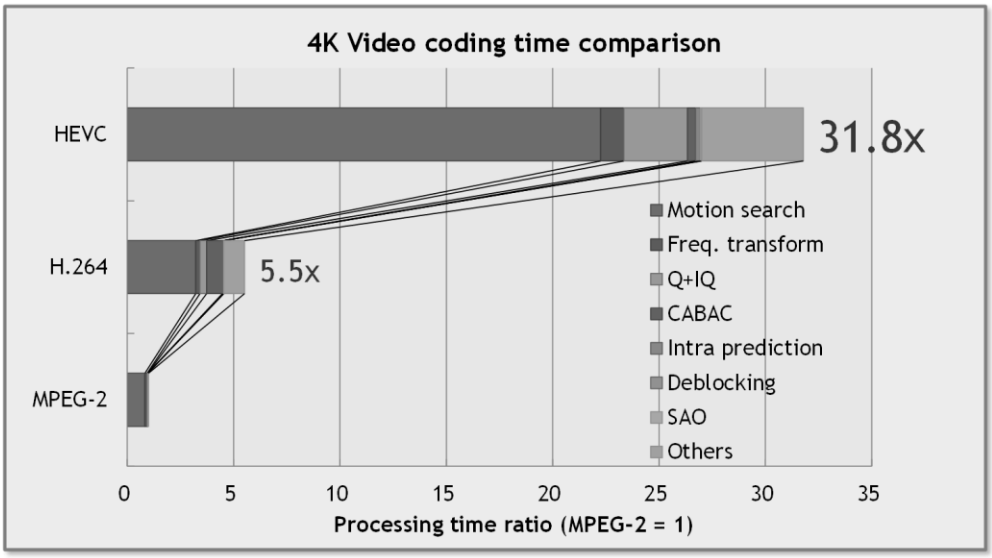
Figure 1.4 Coding modes comparison of MPEG-2, H.264/AVC and H.265/HEVC.

values within a block, have to be expressed and coded individually in the stream as coefficients of 8×8 discrete cosine transforms (DCTs). In H.264/AVC, pixel value prediction from adjacent blocks' pixel values with angular directions is introduced with eight directions with three types of 4×4, 8×8 and 16×16 pixel block sizes. In H.265/HEVC, these angular directions are enriched up to 33 directions and block sizes are also expanded to arbitrary combination of 4×4, 8×8, 16×16 and 32×32 pixels, the combination of intra encoding modes and block sizes becomes quite large, compared to the previous standards.

Inter-frame encoding modes have blocks with motion vectors, which represent a motion of objects. In the figure field coding modes are omitted and only frame coding modes are illustrated for simplicity. In MPEG-2, there is only 16×16 pixel block size[3]. In H.264/AVC, however, block size variation is introduced and four combinations of 16×16, 16×8, 8×16, 8×8[4] blocks can be selected for each 16×16 blocks. In the latest standard H.265/HEVC, the

[3] In this figure and explanation, frame coding mode is considered and field coding mode (for interlaced video) with 16×8 pixel block size is not dealt with, as declared in the text.

[4] In H.264/AVC, inter blocks under 8×8 are also defined but they are mainly for pictures



Reference software: MPEG-2_MSSG H.264_JM18.5 HEVC_HM12.1

Figure 1.5 4K video coding time comparison.

concept of coding unit (CU) prediction unit (PU) is introduced, and within a coding tree unit (CTU) which is as large as 64×64 pixel block can be separated into multiple blocks of 32×32, 16×16 and 8×8 CUs. Each CU can have one PU, or further be partitioned into two rectangles or four squares of PUs, and rectangles could be divided one to one or one to three ratios. Each PU has its own motion vector and therefore the number of combinations becomes quite large. To find the best combination in HEVC CTU, ideally all combinations of PU block sizes should be under motion estimation process and the best motion vectors found are secondly go through a coding cost evaluation phase to find the best combination. It is easily understandable that the most of the motion estimation results are discarded in vain during this best combination finding process, and reducing this redundancy helps reducing the coding complexity of the latest coding standard.

Figure 1.5 shows the encoding time required for the same 4K content with reference software encoders which are used for coding quality assessment during the standardization process. It is obvious that as the video coding standards proceed, coding complexity becomes much higher and it takes approximately six- or thirty-two-times longer time to

smaller than SDTV and therefore not mentioned here.

encode, expressly indicating that novel techniques are essential for keeping real-time encoding functionality with more recent standards.

It should also particularly be noted that with all standards the most time-consuming task is the motion estimation (motion search[5]), which takes about 70% of all the encoding tasks. This is mainly due to the accelerated variations of encoding modes.

1.5 Circuit scale of video encoder VLSIs

The author is with Nippon Telegraph and Telephone Corporation (NTT) and has developed several video encoder LSIs mainly targeted for professional markets such as broadcasters, distributors and content creators. The first generation ENC-C and ENC-M [10] [11] [12] in 1995 requires the combination of two LSIs of one for motion estimation and another for the remaining encoding tasks in order to process encoding tasks conforming to MPEG-2, however, soon they were integrated into one VLSI of SuperENC [13] [14] [15] [16] [17] [18] in 1997 as a single-chip MPEG-2 video encoder of SDTV and its next version of SuperENC-II [19] was also developed in 2000. The SuperENC is mainly targeted at consumers' digital video use such as a PC card video encoder for notebook PCs [20], however, it also had the ability to support 4:2:2 chroma format, which has double density chrominance (color) information compared to 4:2:0 format and mandatory for broadcasters and distributors in order to prevent picture quality degradation due to multiple duplication of encoding and decoding. It also had a basic multi-chip configuration [21] [22] functionality to support HDTV encoding and it opened up the door to professional digital video markets. To meet the needs of encoding requirements for digital terrestrial TV broadcasting started in the year 2000, HDTV MPEG-2 encoder VLSI named VASA [23] [24] [25] [26] was developed in 2002, in order to be integrated into digital TV broadcasting contribution networks connecting among broadcasting stations and it was the world-first one-chip VLSI to support HDTV MPEG-2 encoding. It also had a flexible multi-chip connection capability which supported super-high-resolution encoding and also

[5] In video coding, "motion estimation (ME)" is a proper word for expressing the process of finding motion vectors (MVs) of moving objects. The word "motion search" approximately means more detailed process of doing pattern matching between two pictures, however, in this dissertation the two words are used for almost the same meaning.

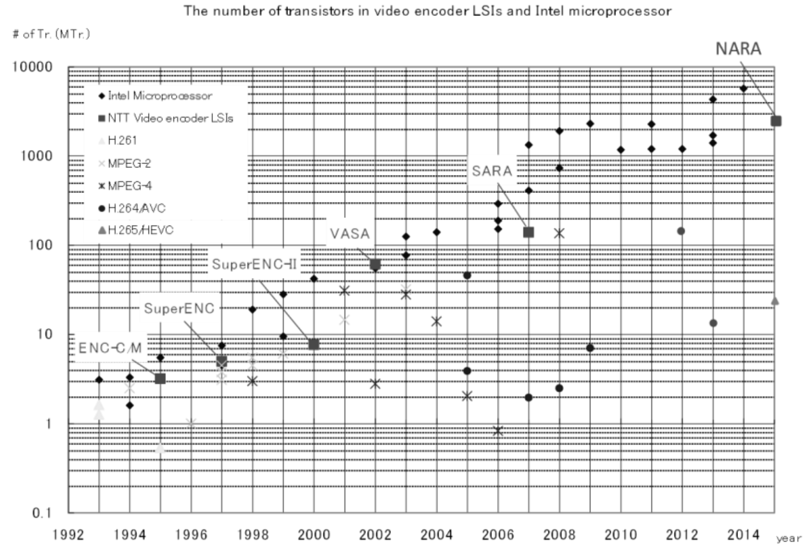


Figure 1.6 The number of transistors in video encoder LSIs and Intel microprocessor.([70], extended by the author)

multi-channel outputs with simple inter-chip output connection which will be proposed and discussed later in the dissertation.

As the next standard H.264/AVC emerged and was anticipated to be a promising standard for high compression of HDTV videos for network distribution and Blu-ray disc recording, H.264/AVC encoder chip SARA [27] [28] [29] [30] [31] was developed in 2007 for re-transmission of digital terrestrial HDTV broadcasting over next generation fiber-to-the-home (FTTH) IP network, which could encode HDTV videos with six multi-chip configuration. And then the successor of H.264/AVC with double the coding efficiency, H.265/HEVC was standardized and defined to be a mandatory video format for compressing 4K/8K to be broadcast over the satellite. H.265/HEVC encoder VLSI named NARA [32] [33] was thus developed in 2015, with the single-chip encoding capability of 4K and with parallel encoding configuration of four chips for 8K encoding in real time.

Encoder appliances using these video encoder LSIs are mainly targeted on broadcasters and distributors, usage of which are content broadcast and distribution encoding and contribution (transmission of contents from sports stadiums or news sites to the broadcast stations) for content creation. The video encoder VLSIs therefore must have strict real-time operation functionalities, good picture quality which meets the need for broadcasting use, adaptability to various video formats (e.g. progressive / interlaced video, 3-2 pulldown video for cinema contents) and on-chip filtering capabilities, wide variety of encoding configurations (e.g. high compression for broadcasting vs. high bit rate and ultra-low delay compression for contribution), a wide range of audio encoding channels and formats, and overall multiplexing output functionalities. These requirements for professional use continue to be a burden for LSI development.

Figure 1.6 illustrates a comparison of the number of transistors in NTT's video encoder VLSIs, video codec VLSIs which were presented at the International Conference on Solid-State Circuits (ISSCC), and for the reference of the largest state-of-the art VLSIs, in Intel microprocessors. NTT's video encoder in each generation has the scale approaching Intel's microprocessors, showing that the design and manufacture cannot be done without cutting-edge technology and circuit scale at the time. This limitation causes very high manufacturing cost, high power consumption and high heat generation, which prevent the encoders to prevail or extend their field to mobile applications such as camcorders and mobile video transmitters. It can also be said that although video encoder VLSIs for professional use can be manufactured with the highest state-of-the-art technologies, as the video coding standards become more complex in the future, video encoder VLSIs may reach the glass ceiling and cannot exploit the full advantage of newer standards. Further circuit scale and power reduction techniques while maintaining picture quality are thus mandatory for video encoder LSI development in the future.

1.6 Objectives of the dissertation

The main objective of this dissertation is to realize super-high-resolution real-time video encoder VLSIs with high image quality. Encoding capability of super-high-resolution of 4K and 8K with broadcast image quality continues to be a driving force for a better consumer experience with finer quality broadcasting, distribution and packaged contents. Apart from digital TVs, even more resolution such as 16K and 32K is required for very natural 360 degree virtual reality (VR) experience [34] with wearing VR glasses, yet higher resolution beyond 8K therefore will soon be expected to appear in the near future.

To achieve the above-mentioned high-quality video encoder VLSIs, the author finds three major issues to be solved, which are listed below and are illustrated in Figure 1.7:

1) Real-time processing yet maintaining high image quality

2) Support for higher resolution over single-chip capability

3) Deeper reduction in circuit scale and power consumption

As for 1), as mentioned in Section 1.4, video coding standard has become much more complex and resource-consuming in exchange for better coding efficiency. Here, “complex” means more types of prediction measures and more combination of motion vectors and coding modes with more block size patterns.

It should be noted that in each video coding standard, what is defined is how to decode the coded bit stream back to video frames, not how to encode the video frames into bit streams. This means that there is no restriction for encoders to choose the combination of prediction types, motion vectors and coding modes as long as generated bit streams can be decoded with the conformant decoder defined by the standard. Each encoder is expected to choose the best motion vectors and the best coding modes to maximize the coding efficiency, and the reference software encoders which are developed during the standardization process usually adopt brute force search of all coding modes for achieving the best coding efficiency, in exchange for computational complexity unrealistic for real-time processing. Real-time encoder architecture therefore should seek for techniques for reducing the computational complexity dramatically to meet the real-time operations, yet minimizing the decrease in coding efficiency by utilizing statistical correlations between multiple coding modes and motion vectors of adjacent blocks. This will be discussed as video prediction core architecture in Chapter 2.

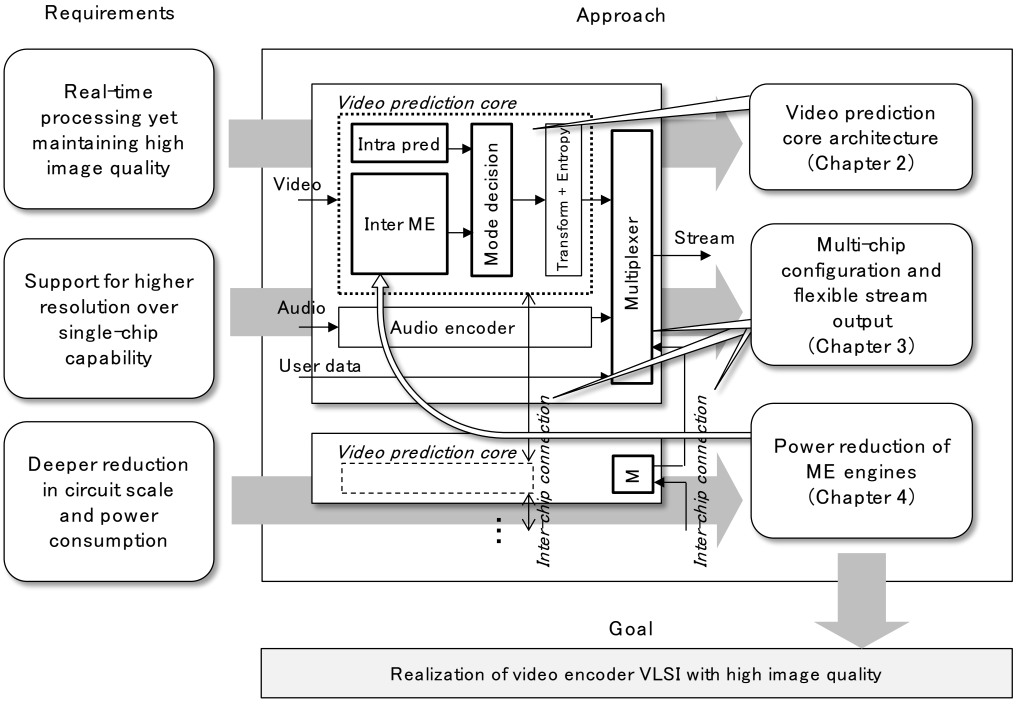

Requirements
Approach
Real-time processing yet maintaining high image quality
Support for higher resolution over single-chip capability
Deeper reduction in circuit scale and power consumption
Video prediction core
Intra pred
Inter ME
Mode decision
Transform + Entropy
Multiplexer
Video
Audio
Audio encoder
User data
Stream
Video prediction core
M
Inter-chip connection
Inter-chip connection
Video prediction core architecture (Chapter 2)
Multi-chip configuration and flexible stream output (Chapter 3)
Power reduction of ME engines (Chapter 4)
Goal
Realization of video encoder VLSI with high image quality

Figure 1.7 Objectives of the dissertation.

About item 2), cutting-edge applications which require the highest resolution videos (such as HDTV and 8K systems in the early days, digital cinema solutions, multi-channel and multi-angle public viewing systems) allow certain size of encoding devices because they are installed in indoor chassis and do no go mobile. However, these applications tend to be difficult to be developed with a single-chip VLSI because of mainly two reasons: operation scale is too large to fit into one-chip VLSI with current CMOS technology, and, the number of devices required is too small to develop and manufacture dedicated VLSIs. Considering the characteristics above, parallel processing of video encoder VLSIs with multi-chip configuration is a promising solution.

To achieve flexible multi-chip solution mentioned above, inter-chip connection framework in order to support super-high-resolution and multi-channel encoding is proposed. When external devices for multi-chip control and operation are required, they could cause extra burden and cost for encoding device development, flexible inter-chip connection without external devices is desirable and it is proposed in Chapter 3.

About item 3), even if real-time encoder VLSI can be developed with the above two techniques, super-high-resolution video encoders are still circuit-scale and power consuming appliances, hard to adapt 4K and 8K encoders to mobile applications such as 4K/8K field pick-up units (FPUs) for on-site news gathering for broadcasters and 4K/8K cameras with wireless connections for real-time video uploads to clouds. Furthermore, to think about the next-generation video coding standards such as VVC, coding complexity will become much higher and further power reduction techniques in power-consuming tasks are quite essential for next generation video encoding VLSIs, while maintaining coding efficiency. To meet these requirements, bit reduction technique in most power-consuming motion estimation (ME) is proposed. Sum of absolute differences (SAD) calculation in ME engines is bit-shortened. However, by changing the bit extraction positions due to the picture characteristics of input videos, SAD calculation precision in flat luminance regions is preserved and coding quality is maintained, which is proposed in Chapter 4.

1.7 Overview of the dissertation

The rest of the dissertation is structured as follows. In Chapter 2, the proposed video encoding core architecture which achieves real-time processing of the H.264/AVC standard and the successive standard H.265/HEVC is presented, which achieves significant computational complexity reduction for real-time processing while preserving coding efficiency by fully utilizing the statistical correlation between adjacent blocks. The coding efficiency of the proposed video encoding core architecture is evaluated with objective and subjective evaluation of encoded videos. In Chapter 3, inter-chip connection framework in order to support yet higher resolution and multi-channel encoding with parallel operation of multiplexers is proposed, and multiplexed output quality of the proposed technique is assessed by the analysis of output streams. In Chapter 4, motion estimation engines with adaptively bit-reduced SAD calculation is proposed, and reduction effect on circuit scale and power consumption is assessed with power simulation of HEVC video encoder VLSI design. Effect of preserving coding efficiency with the proposed technique is also evaluated with a software simulator of the encoder. Chapter 5 concludes the dissertation with the summary of the results in the researches, with future works to be undertaken in the future.

Chapter 2

Video encoding core architecture

2.1 Introduction

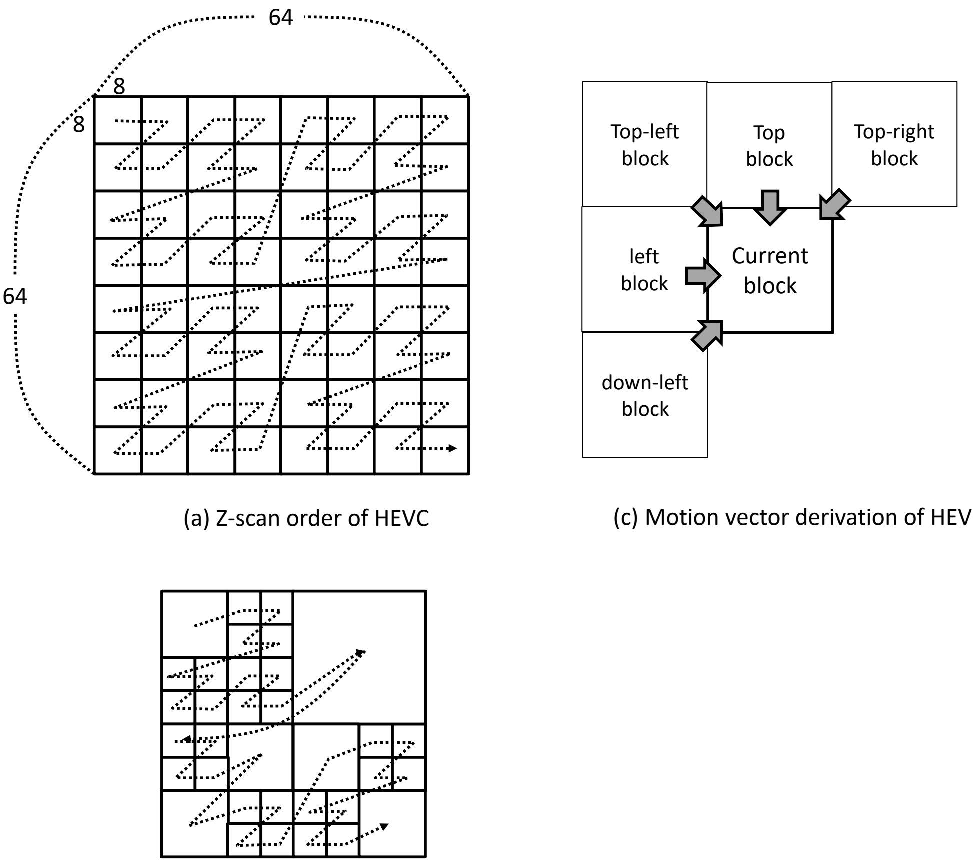
In this section, the video encoding core architecture of real-time video encoder VLSIs aiming at broadcast-level video quality is introduced. In this dissertation, intra prediction and inter-frame prediction blocks, including motion estimation blocks and succeeding encoding mode decision blocks, are from now on called "prediction core" and especially targeted. It is because this prediction core has the function of determining the final encoding modes with motion vectors, which is directly linked to the coding efficiency of the encoder VLSIs and which is a source of competitiveness. In addition, as discussed previously in Section 1.4, increase in coding complexity is due to the increasing number and combination of motion vectors and coding modes which need to be evaluated, real-time processing of the prediction core is crucial for video coding core design.

Video encoder VLSIs, at which this dissertation is going to target, are expected to play an important role in the field of television infrastructures, such as digital terrestrial, satellite, IPTV broadcasting, real-time video contribution and distribution over comsat/IP networks, For these applications, in order to provide professional-quality images with lower bit rate, high performance encoders should care about the following performance requirements:

(1) Support for motion vectors with large motion

Video contents to be broadcasted or distributed contain large motion of objects and professional quality video encoders are expected to deliver the contents without severe degradation due to large motion. For example, as for 4K HEVC encoding, according to the author's previous findings, when the search rage between frames that are 1/60 second apart is ±48, sufficient search performance is obtained for most broadcast program materials with horizontal pans [33]. Motion estimation engines in the prediction core should therefore support the motion search range to meet at least the above-mentioned requirements.

(2) Support for motion vectors of multiple block sizes



(a) Z-scan order of HEVC

(c) Motion vector derivation of HEVC

(b) Z-scan example(mixture of block sizes)

Figure 2.1 Example of encoding mode derivation.

Finding motion vectors for huge combinations of block sizes is another major issue to be solved for real-time processing. It is expected that adjacent blocks or overlapped blocks with different block sizes have strongly correlated motion vector values. The solution of this issue therefore is the combination of two types of motion estimation blocks. The first one is a wide-range motion search with limited block sizes. The second one is a neighboring motion estimation from the results of the first wide-range motion estimation, with multiple block sizes.

(3) Support for precise encoding mode decision

Video encoding standards usually take measures of describing motion vectors and encoding modes as the difference from those of adjacent blocks. For example, Figure 2.1 shows the encoding order and the motion vector derivation manner from adjacent blocks in HEVC. In an HEVC's CTU, which typically comprises 64×64 pixel block, 8×8 blocks are sequentially coded in z-scan order, which goes from top-left block to down-right block as illustrated in Figure 2.1 (a). When multiple block sizes over 8×8 block are mixed to form the 64×64 pixel CTU, z-scan order of larger blocks are skipped as shown in Figure 2.1 (b). And Figure 2.1 (c) illustrates the example of motion vector derivation in the H.265/HEVC standard, showing that when the center block is being encoded, motion vector predictions are derived from motion vectors of adjacent blocks, and only the difference values from the predictions are written in the encoded stream. Adjacent blocks which are not yet encoded in z-scan order are obviously non-existent and omitted from the motion vector prediction calculation.

Coding cost of motion vectors increases as the difference gets larger, therefore in order to get a precise coding cost evaluation for encoding mode decision, motion vectors and coding modes of previously-encoded blocks (in z-scan order) should be fixed. If encoding mode decision is performed in parallel or performed before ME is firmly fixed, coding cost evaluation becomes inaccurate because of inaccurate motion vector prediction and it results in a degradation in coding efficiency. For example, it is reported that when choosing a block coding mode after integer motion estimation (IME) and before fractional motion estimation (FME), it degrades coding performance by up to 1 dB [35]. In order to seek a precise encoding mode decision for maintaining coding efficiency, encoding mode decision, including final selection of motion vectors, should be done after adjacent blocks' motion vectors and encoding modes are all determined.

(4) Support for flexible mode decision adjustment

Mode decision algorithms and parameters have to be updated due to picture quality degradation in specific picture scenes, usually in order to respond to the requests and claims from professional customers from broadcasters and distributors. Mode decision algorithms and patterns also need to be changed in different coding modes such as frame/field adaptive coding for interlaced videos and 4:2:2 chrominance videos (double density chrominance signals for professional video sources). Mode decision processing therefore is required to be flexibly changed with firmware upgrade.

In order to solve the three requirements above, this section deals with two proposed

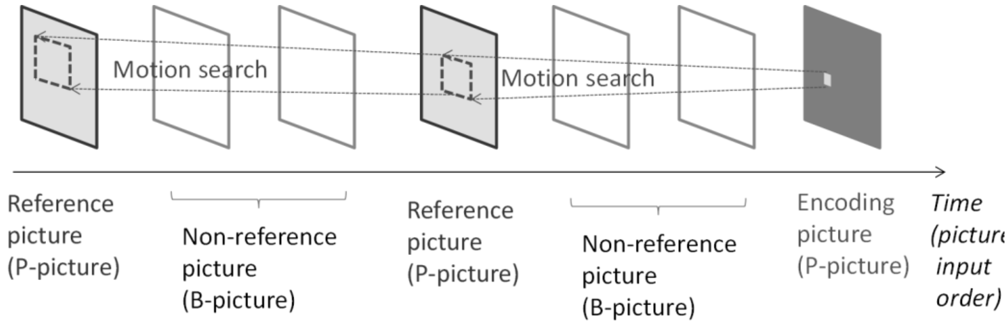

Motion search
Motion search
Reference picture (P-picture)
Non-reference picture (B-picture)
Reference picture (P-picture)
Non-reference picture (B-picture)
Encoding picture (P-picture)
Time (pictur input order)

Figure 2.2 Motion estimation example in H.264/AVC.

prediction core architecture: one is for H.264/AVC encoding and another is for H.265/HEVC encoding. These two have a common design philosophy of “wide motion search with limited block sizes, neighboring search with multiple block sizes, and precise final mode decision.” According to the features of each standard, prediction cores are designed in order to achieve real-time processing and also the above requirements. Furthermore, for H.265/HEVC encoding, combination of intra prediction modes also sharply increased as described in Section 1.4, intra prediction complexity reduction technique is also presented.

In the following section, proposed prediction core architecture with telescopic wide motion estimation and inclusive 8 × 8 through 16 × 16 second motion estimation with of H.264/AVC is first described.

2.2 H.264/AVC prediction core architecture

2.2.1 Motion estimation characteristics in H.264/AVC

In H.264/AVC, the concept of “multiple reference picture” is introduced. Figure 2.2 shows the example of multiple reference pictures. H.264/AVC’s picture reference structure is similar to that of MPEG-2. P-picture, which is used for reference picture, is placed in every three pictures, in between two non-referenced B-pictures are placed. When encoding a P-picture, referenced P-picture is located at three pictures distance in time and motion estimation should be executed between these P-pictures. In H.264/AVC’s multiple reference picture concepts, in addition, another P-picture which is located at six pictures distance in time can also be used for reference pictures, therefore motion estimation between P-pictures of six pictures distance should also be performed.

When picture distance becomes larger, motion of objects in pictures becomes also larger and motion search range also needs to be larger to correctly track the motion of objects. However, this causes the increase of motion search range in n^2 order and causes a severe increase in motion estimation processing. To solve this issue, the concept of telescopic search [10] is extended to accept the H.264/AVC's multiple reference pictures, which detail is described in the next subsection.

2.2.2 Telescopic primary motion estimation

The concept of telescopic search is shown in Figure 2.3. Motion search is first performed for the closest image in time from the image to be encoded, and then next motion search is performed for next farther image, through the farthest reference picture. The motion vector that is the search result of a certain image is used for search center of the next farther image. It should be noted that this telescopic search is performed with all successive pictures including non-reference pictures (such as non-reference B-pictures in Figure 2.2) in order to correctly track the motion of objects in pictures.

This way motion estimation is performed from the nearest picture through the farthest picture. Repeating the search to track the motion of objects in the picture makes it possible to obtain a larger motion vector for farther reference pictures, with relatively small search range and small picture load from each picture. This characteristic of telescopic motion estimation is suitable for H.264/AVC's multiple reference picture concepts, since motion vectors of far reference pictures can be obtained with this motion tracking attribute of the telescopic search.

In H.264/AVC encoding, variable block size of 8×8 or more should be supported by motion estimation engines. Therefore, as shown in Figure 2.4, the telescopic search is performed as the primary motion estimation, in units of 8×8 blocks. 1/2 reduced images are used for the search, and a double-pel precision motion vector is obtained using a 4×4-pixel template.

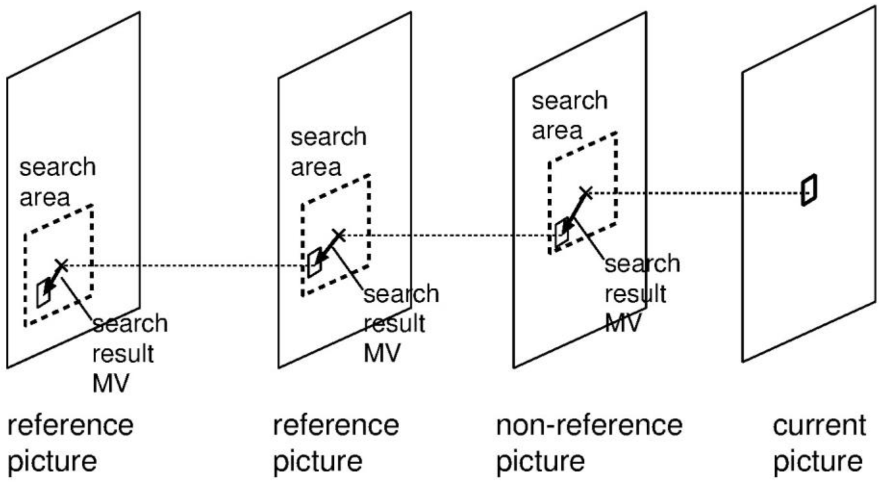


Figure 2.3 Concept of telescopic motion estimation.

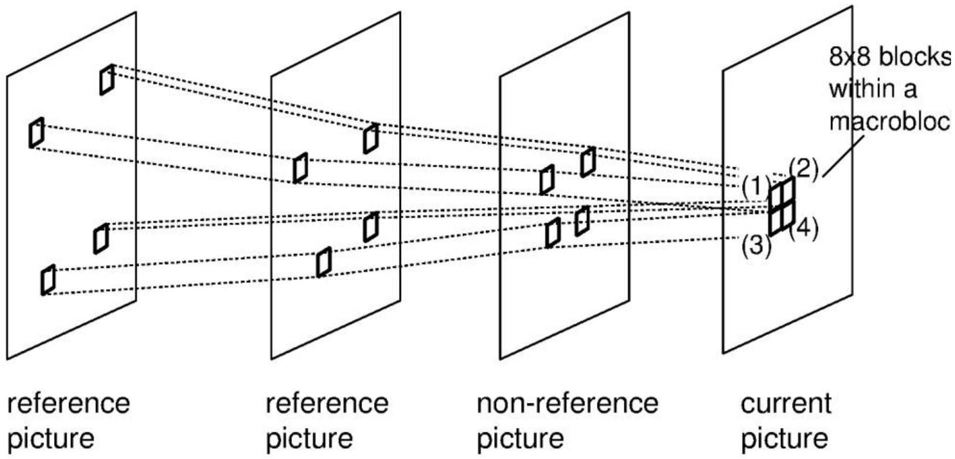


Figure 2.4 An 8×8-based telescopic motion estimation.

The search near the zero point (near the same position of the encoding block) and near the motion prediction vector (PMV) which is defined in the H.264/AVC standard is performed separately directly on the reference image. These additional searches are performed because if motion tracking with telescopic search fails with picture noises, picture occlusion or irregular motion of objects, motion vectors are hard to be adjusted. Motion vectors near zero-vector or PMV therefore should be searched individually.

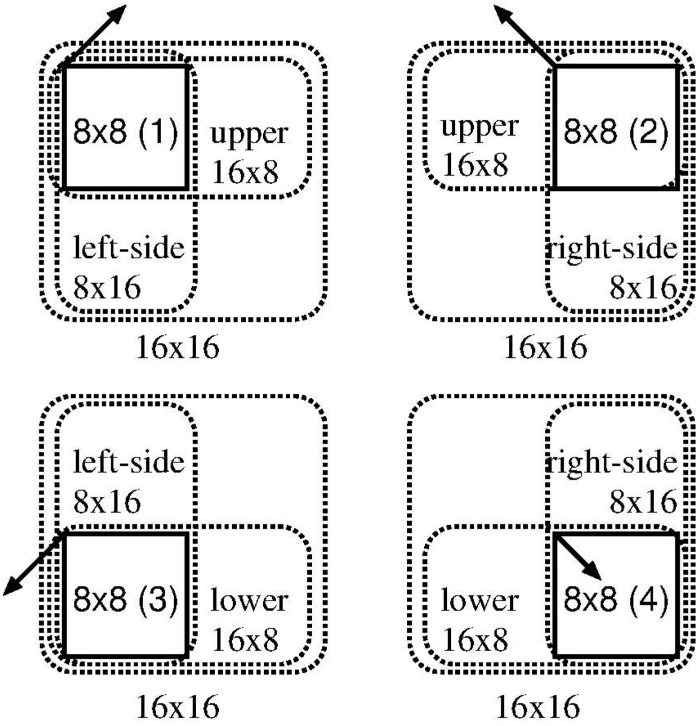


Figure 2.5 An 8×8-based “inclusive” neighboring motion estimation.

After that, the neighborhood search of ± 1 pixel is performed on the non-reduced image to obtain motion vectors with integer precision.

In this way, integer precision motion vector is obtained for all reference pictures of each 8×8 block. When encoding B-pictures, one reference picture for forward prediction and the other one for backward are selected for each 8×8 block. As for P-picture encoding, up to two reference pictures for forward prediction are selected and transferred to the following fractional motion estimation procedure.

2.2.3 Inclusive secondary motion estimation

Receiving the motion vector results of integer-pel precision from the primary telescopic motion estimation, the objective of second motion estimation is to:

- obtain quarter-pel motion vectors with a fractional motion estimation and for multi-block-size motion estimation,

- obtain motion vectors other than 8×8 blocks.

To achieve both of above, in the secondary search neighborhood search with 1/2-pixel and 1/4-pixel accuracy is performed for four block sizes of 16×16, 16×8, 8×16, and 8×8 with a proposed "inclusive" multi-block-size neighboring search manner. Figure 2.5 shows the concept of the inclusive search method using primary search results in 8×8 blocks. For the four 8×8 blocks, (1)-(4), the corresponding integer-pel motion vector results are used one by one as a search center. For larger blocks, for example an "upper 16×8" search, 8×8 (1) and (2), which are spatially included in the "upper 16×8" block can be candidate integer-pel motion vectors, and fractional motion estimation is done for each of them as a search center. In the same way, for a "lower 16×8" search, integer-pel motion vectors of 8×8 (3) and (4) are the candidates. In a "16×16" search, all 8×8 blocks from 8×8 (1) through (4) are spatially included, so all of them are used for fractional motion estimation.

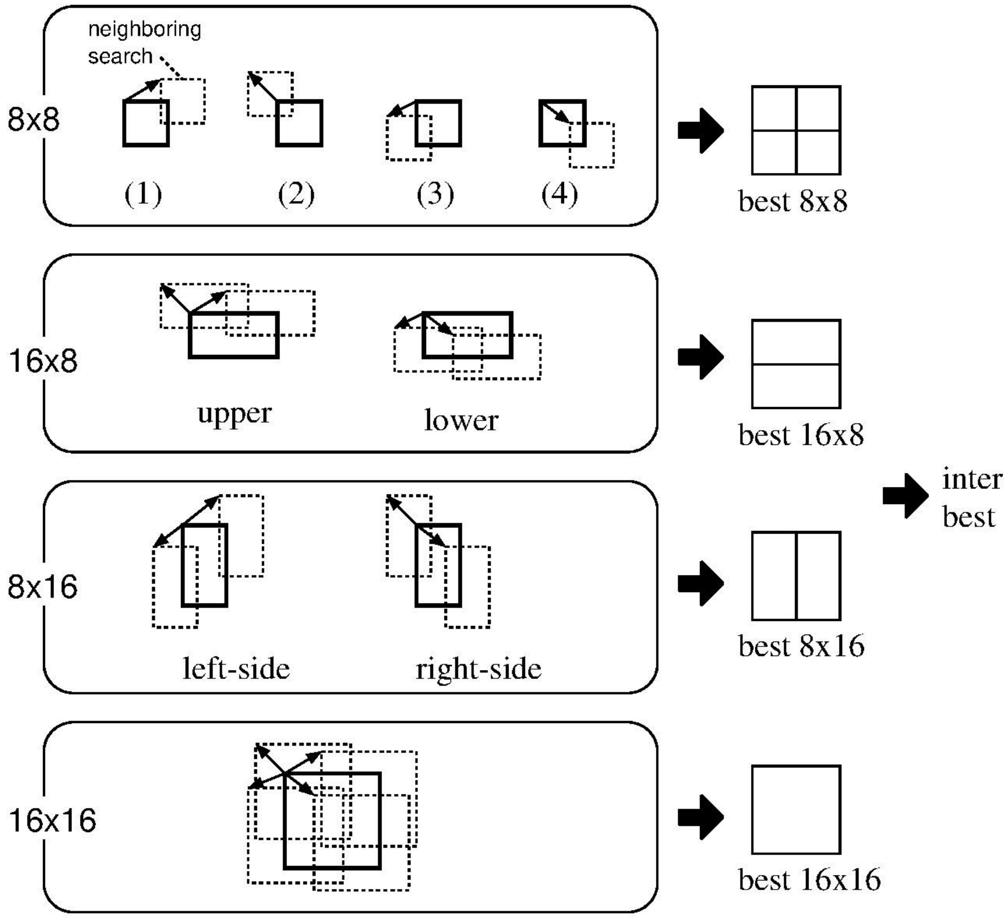


Figure 2.6 Fractional motion estimation process.

Then, as shown in Figure 2.6, the best MV and prediction direction are selected for each block size, and these are compared to determine the final coding mode.

As stated above, the performance of variable block size motion prediction is improved by performing an exhaustive search for all integer-pel precision motion vector candidates spatially included in each block size. On the other hand, the proposed algorithm requires a large amount of computation for neighborhood searches of multiple types of blocks.

Therefore, the next subsection describes the hardware configuration of the fractional motion estimation unit that enables this inclusive search. The proposed unit also achieves high search performance and precise mode decision by fully utilizing parallelism for multiple block sizes but maintaining serialized motion vector decision for each block size.

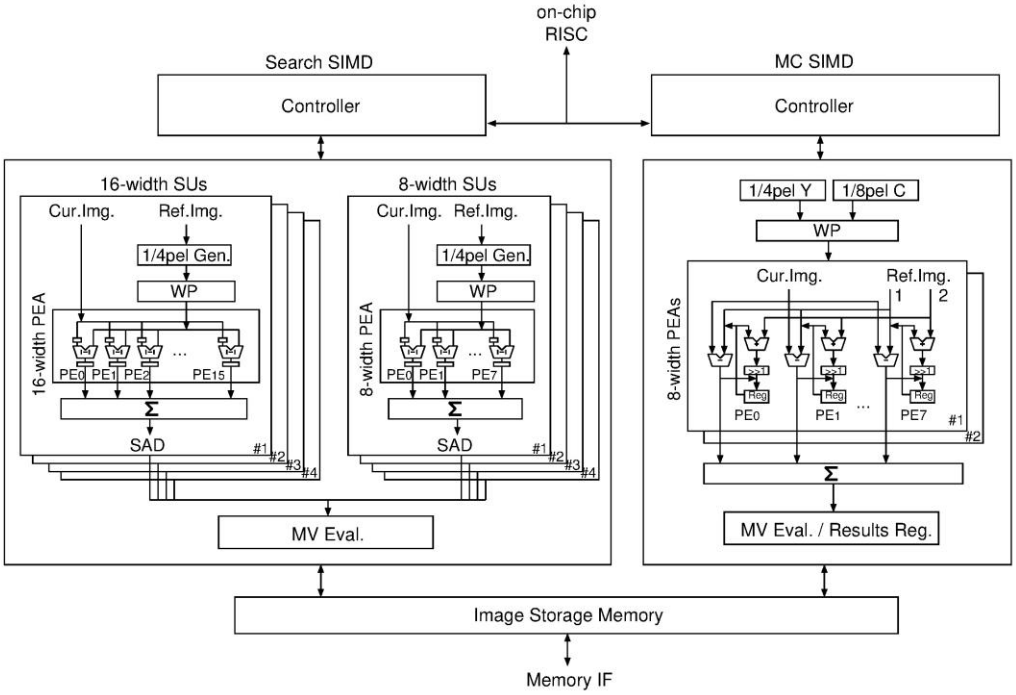


Figure 2.7 Block diagram of fractional motion estimation module.

2.2.4 Configuration of fractional motion estimation and mode decision module

Figure 2.7 shows the block configuration of the fractional motion estimation unit which is configured to do the inclusive motion estimation. This block is designed to search and evaluate block sizes of 8×8 and above, since block sizes less than 8×8 do not contribute much for coding efficiency of SDTV and larger pictures.

In this block configuration, it is necessary to repeat the fractional motion estimation and the prediction mode decision (reference image, MV, prediction direction) for each block size from 8×8 to 16×16. In order to pipeline these tasks in parallel, a single instruction multiple data (SIMD) processor in charge of search and the MC is provided respectively to control the operation of the related functional blocks.

When integer MV candidates are ready, reference images of the secondary search area are half-pel filtered and stored in the image memory. A total of 8 search units (SU) continuously execute neighboring 9-point searches with 1/2- and 1/4-pixel precision according to the

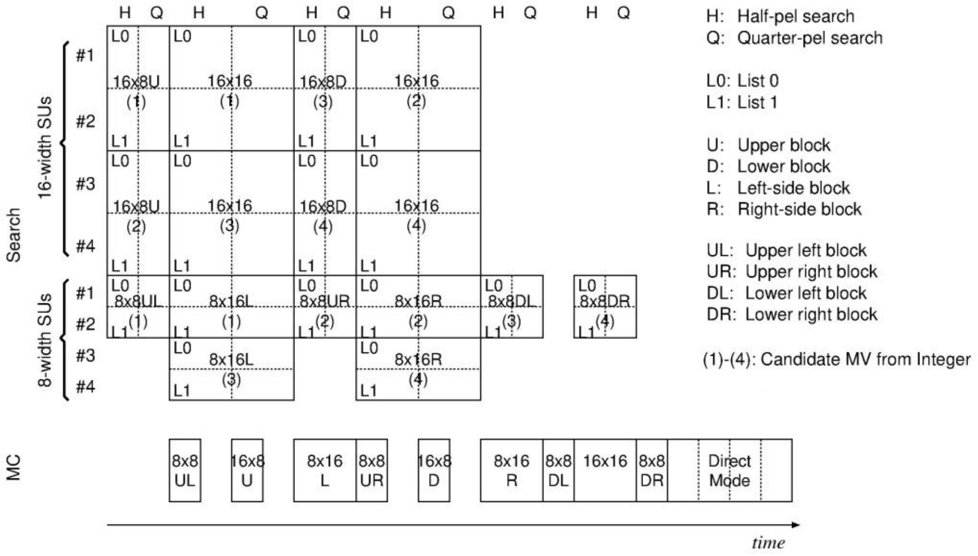


Figure 2.8 Pipeline for B-picture encoding.

instruction of a search control SIMD processor. The SU with 16 pixels wide is responsible for processing 16×16 and 16×8 blocks, and the SU with 8 pixels wide is responsible for processing 8×16 and 8×8 blocks. As directed by a search control SIMD processor, each PE array (PEA) does the neighboring nine-point half-pel search, and then the nine-point quarter-pel search.

In each SU, the reference image is supplied to the PEA through 1⁄4 precision pixel generation and weighted prediction (WP[6]) processing. The absolute difference value between the encoded image and predicted image is calculated in units of one horizontal pixel row (8 to 16 pixels), and the sum is output as a SAD value. Based on the search results of each SU, the motion vector (MV) evaluation block compares the SAD value with the coding cost (the cost required for coding the motion vector and the reference image number) and selects the best motion vector.

[6] In H.264/AVC, the concept of weighted prediction is introduced, which transforms the predicted pixel value p(x, y) with

$$\mathrm{p}'(\mathrm{x},y) = scale\ \times p(x,y) + offset$$

in order to overcome the problem that motion estimation does not work well with scenes with changing light conditions or fading scenes. The proposed configuration does not support *scale* factor but supports *offset* factor when explicitly designated by outside of fractional motion estimation block.

In parallel with the search, MC operation is performed based on the reference image and motion vectors selected in each block. A predicted image is generated by two 8-pixel-wide PEAs according to the instruction of the MC control SIMD processor. The PE that composes this PEA includes an adder, a shifter, and a register. In B-picture, a bidirectional prediction image is also generated while switching the data path and the best prediction direction is determined. The search results are successively accumulated in the register to prepare for final block mode decision.

Figure 2.8 shows the pipeline operation when performing bi-directional prediction of B pictures. For each block size, fractional search and decision of prediction direction by MC operation are sequentially processed from upper left to lower right. All processing ends in one macroblock cycle, including direct mode[7] evaluation and final block mode determination processing.

Thus, by performing parallel pipeline control of search and MC processing by cooperation of two SIMD processors, inclusive fractional-pel precision search can be realized. Operation control program on each SIMD processor is flexibly configured, changing the operation of PAs and data paths so as to adapt to various coding modes such as picture adaptive frame/field (PAFF), macroblock adaptive frame/field (MBAFF), WP and 4:2:2, as well as to prior standards of MPEG-2/4.

Furthermore, the two SIMD processors compare the coding costs of all block modes based on mode decision programs and select the best coding mode. Coding costs are provided as ME/MC results and also from an intra prediction module as intra coding costs. Mode decision offsets and thresholds are externally modified from a top-level RISC CPU to control scene-adaptive parameters.

In this manner, co-operative control and mode selection operation by the two SIMD processors facilitates adaptation to a wide variety of operating modes and upgrading of mode decision algorithms.

[7] Direct mode is an inter-frame prediction mode where motion vectors are derived from motion vectors of adjacent previously-encoded blocks. Direct mode does not need to describe motion vectors in the encoded stream and can save bits. However, in order to evaluate direct mode prediction images, motion vectors of adjacent previously-encoded blocks must be fixed and pipelined mode decision is now allowed here.

2.2.5 Features in the proposed H.264/AVC prediction core

The H.264/AVC prediction core architecture which is described above has the following features for each requirement described in Section 2.1:

(1) Support for motion vectors with large motion: The “telescopic” primary search for multiple reference pictures, capable of tracking the motion of objects in farther reference pictures in time.

(2) Support for motion vectors of multiple block sizes: The “inclusive” secondary fractional motion estimation from primary search results of 8×8 blocks fully utilizes the primary motion estimation results in each block size.

(3) Support for precise encoding mode decision: Motion estimation and mode decision scheduling in the secondary fractional motion estimation achieve sequential decision of motion vectors and coding modes in each block size, realizing precise coding cost evaluation with motion vectors and coding modes of adjacent blocks all fixed. This avoids ambiguous estimation of coding costs with unfixed adjacent blocks’ motion vectors and coding modes, which results in coding cost degradation.

(4) Support for flexible mode decision adjustment: Operation of the two SIMD processors can be changed and upgraded for mode decision improvement and different encoding modes.

Implementation and encoding quality performance of the proposed techniques are described later in Section 2.6.1.

The above-mentioned techniques are well suitable for H.264/AVC encoding, however, in the next latest coding standard H.265/HEVC, major change in reference picture structures and significantly increased prediction modes and their combination make the proposed techniques hard to adapt to H.265/HEVC by themselves. In the next section, a set of new techniques which are proposed to adapt to the new characteristics in H.265/HEVC are described, in order to achieve real-time encoding of super-high-resolution H.265/HEVC video.

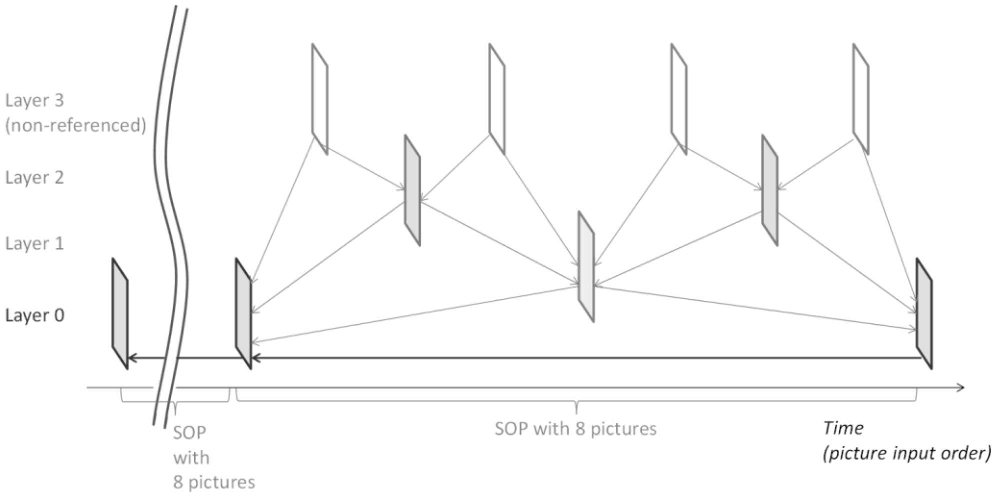


Figure 2.9 Example of “pyramid” style reference picture structure in H.265/HEVC.

2.3 H.265/HEVC prediction core architecture

2.3.1 Motion estimation characteristics in H.265/HEVC

In order to achieve higher coding efficiency, substantial changes have been done for prediction techniques of H.265/HEVC. One of the major changes is the “pyramid” style reference picture structure, which is partly adopted in H.264/AVC instead of conventional P/B-picture structure and is fully adopted in H.265/HEVC encoding. Figure 2.9 shows the example of “pyramid” style reference picture structure, where pictures are classified into multiple layers and each picture in a certain layer refers to (i.e. perform motion estimation to) nearest pictures of one layer below. A set of pictures constructing one “pyramid” is called a structure of pictures (SOP) and an SOP usually comprises 8 pictures[8]. Compared to the conventional MPEG-2 and H.264/AVC reference structure as depicted in Figure 2.2, this pyramid style structure has an advantage in coding efficiency, because pictures over layer 0

[8] SOP size can be set to 2^n, smaller SOP leads to less latency and bigger SOP yields more coding efficiency. For high frame rate encoding such as 100 frames per second (in Europe) and 120 frames per second (in Japan/USA), SOP size of 16 is often selected.

is always placed at equal distance between two reference pictures and well-predicted even in scenes with fades and lighting changes, as long as the motions are uniform.

To think about motion estimation of layer 0, motion estimation has to be performed with 8 picture distance in time, and for multiple reference 16 picture distance. Storing all intermediate 15 pictures for telescopic search consumes too much buffer memory and the telescopic search therefore is not appropriate.

Instead of telescopic search, inter-frame prediction for H.265/HEVC therefore adopts direct search between encoding picture and reference pictures. To achieve a request of larger search range without the telescopic search technique, and at the same time to realize a precise motion estimation, direct primary search with adaptive downscaling is proposed.

Sharply increased block size combination of 8×8/16×16/32×32/64×64 pixel blocks with rectangle partitions is another problem to be solved. Proposed "inclusive" motion estimation for H.264/AVC is not realistic to adapt to H.265/HEVC due to this huge combination, therefore a secondary search for H.265/HEVC works in double-pel precision and heavily duplicated evaluation of different sized blocks are performed with aggregated multi-block-size SAD calculation. Integer and fractional-pel motion estimation are placed as tertiary search. In addition, in order to meet the high demand of reference picture load from these search modules, high-speed reference image cache architecture is proposed.

For mode decision, huge combination of motion vectors, coding modes and block sizes is a substantial issue. A mode decision module that allows deeply centered mode decision procedure with multi-block-size combination is thus proposed.

In the next subsection, block diagram of overall H.265/HEVC video encoder VLSI is first presented for illustrating general features of the VLSI design, and then proposed techniques for each motion estimation module block are individually described.

2.3.2 Block diagram of H.265/HEVC video encoder VLSI

A block diagram of proposed H.265/HEVC video encoder VLSI is shown in Figure 2.10. Input images obtained from video interface (VIF) are passed through image feature extraction (IFE) in order to gather image characteristic data for picture quality control, stored into external DDR-SDRAMs, and supplied to a prediction core with the encoding picture order. The prediction core consists of a multi-block-size edge detector (MED) to

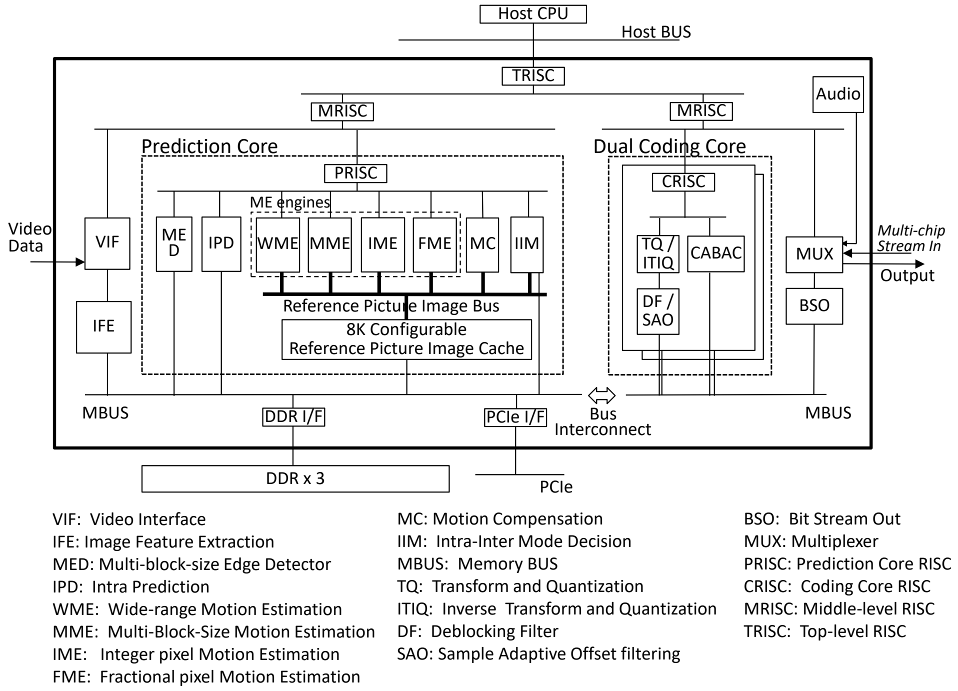

Host CPU
Host BUS
TRISC
Audio
MRISC
MRISC
Prediction Core
Dual Coding Core
PRISC
CRISC
ME engines
Video Data
VIF
ME D
IPD
WME
MME
IME
FME
MC
IIM
TQ / ITIQ
CABAC
MUX
Multi-chip Stream In
Output
DF / SAO
BSO
IFE
Reference Picture Image Bus
8K Configurable Reference Picture Image Cache
MBUS
DDR I/F
PCIe I/F
Bus Interconnect
MBUS
DDR x 3
PCIe
VIF: Video Interface
IFE: Image Feature Extraction
MED: Multi-block-size Edge Detector
IPD: Intra Prediction
WME: Wide-range Motion Estimation
MME: Multi-Block-Size Motion Estimation
IME: Integer pixel Motion Estimation
FME: Fractional pixel Motion Estimation
MC: Motion Compensation
IIM: Intra-Inter Mode Decision
MBUS: Memory BUS
TQ: Transform and Quantization
ITIQ: Inverse Transform and Quantization
DF: Deblocking Filter
SAO: Sample Adaptive Offset filtering
BSO: Bit Stream Out
MUX: Multiplexer
PRISC: Prediction Core RISC
CRISC: Coding Core RISC
MRISC: Middle-level RISC
TRISC: Top-level RISC

Figure 2.10 Block diagram of the H.265/HEVC video encoder VLSI.

perform efficient intra prediction (IPD) and 768 GOPS ME engines for performing wide-range variable block-size motion vector (MV) search. The prediction core also has an 8K configurable 210 Mbit reference picture image cache with a 5120-bit image bus connected to the ME engines to meet reference picture demands. Detailed structures and processing algorithms of the prediction core are illustrated later in this section. After prediction, coding cores perform transform and quantization (TQ), filtering (DF/SAO), and entropy coding (CABAC) operations. The coding core’s dual configuration allows CABAC to adopt wavefront parallel processing (WPP) and also picture/slice level parallelism, which achieves a bit stream output up to 600 Mbps. The coded bit stream is multiplexed (MUX) with audio streams, with multichip input for 8K stream generation.

2.3.3 Primary motion estimation with adaptive down sampling in WME

For 4K and 8K ultra-high definition encoding, motions of picture objects become larger in proportion to pixel density, making a wide motion search range essential. Motion estimation

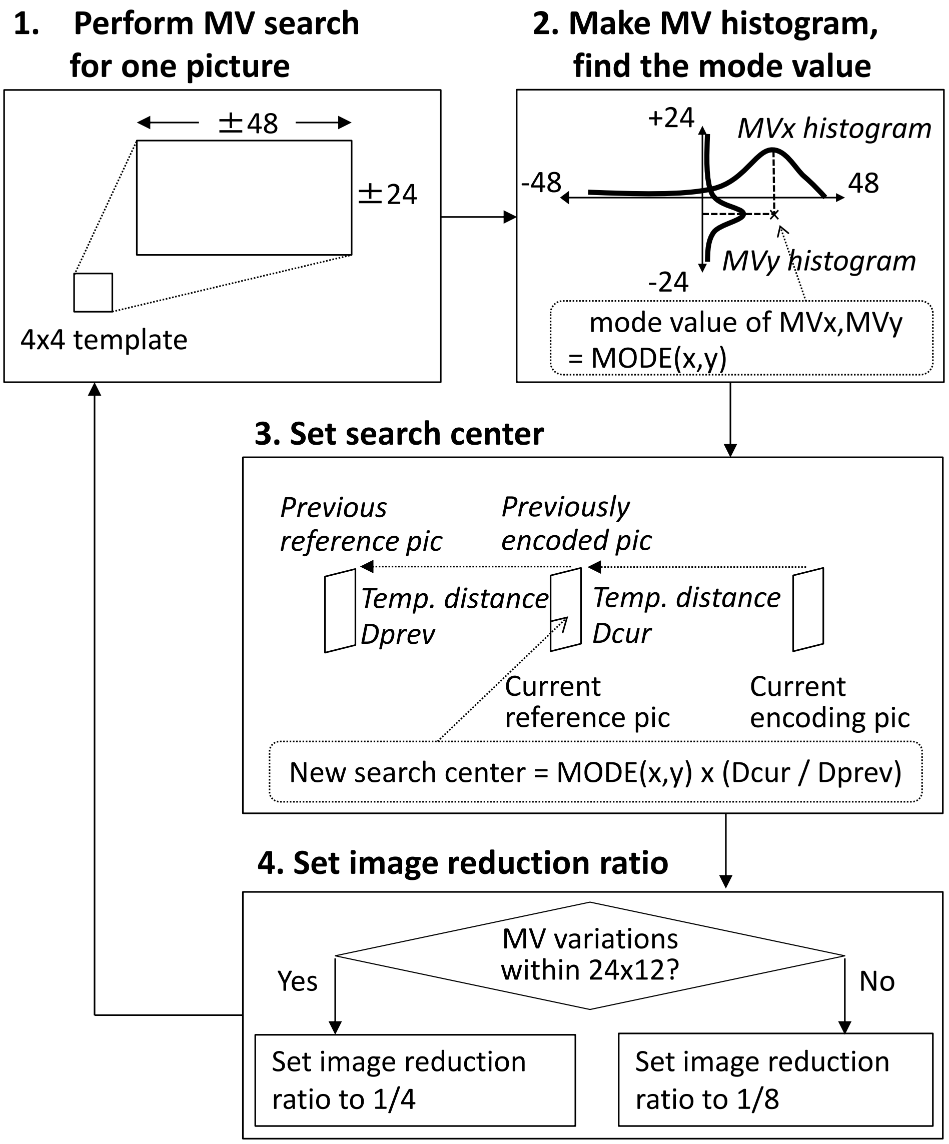

1. Perform MV search for one picture
±48
±24
4x4 template
2. Make MV histogram, find the mode value
+24
MVx histogram
-48
48
-24
MVy histogram
mode value of MVx,MVy = MODE(x,y)
3. Set search center
Previous reference pic
Previously encoded pic
Temp. distance Dprev
Temp. distance Dcur
Current reference pic
Current encoding pic
New search center = MODE(x,y) x (Dcur / Dprev)
4. Set image reduction ratio
MV variations within 24x12?
Yes
No
Set image reduction ratio to 1/4
Set image reduction ratio to 1/8

Figure 2.11 Wide motion estimation with statistically adaptive approach.

with subsampled and downscaled pictures is a common technique, but template matching with deeply downscaled pictures results in poor matching accuracy. The technique therefore can be improved by adaptively changing the downscale ratio and motion search centers with statistical motion vector analysis as illustrated in Figure 2.11.

At the beginning of the inter prediction, wide-range motion estimation (WME) performs 4×4 template matching with a ±48×±24 search range (equivalent to ±384×±192 motion vector in images downscaled to 1/8). According to our previous findings, when the search rage between frames that are 1/60 second apart is ±48, sufficient search performance is obtained for most broadcast program materials with horizontal pans. Since H.265/HEVC's reference structure generally requires motion estimation of an eight-frame distance at maximum, horizontal ±384 search capability therefore is given to the initial wide motion estimation.

When WME is performed for one picture, search results are statistically stored to form a two-dimensional WME motion vector (MV) histogram as illustrated in Figure 2.11. The top of the histogram is chosen as a major motion vector of this picture, and when the next picture is encoded, this major motion vector is stretched in proportion to the temporal distance of pictures with the equation of:

$$\text{New search center} = \text{MODE}(x, y) \times (\text{Dcur} \ / \ \text{Dprev})$$

where MODE (x, y) is a mode value of motion vector histogram in x-axis and y-axis each, Dcur and Dprev are distances in time between "current encoding picture and current reference picture" and "previously encoded picture and previous reference picture." The results of this equation are interpreted as the most probable motion vector of the current picture and set as the current picture's WME search center. The motion vector distribution of the histogram is also stretched, and if it is within the search range of 1/4 downscaling, a 1/4 downscaled search is applied. Otherwise, 1/8 downscaling is applied. In this way, search centers and down sampling ratio are adaptively changed statistically due to previously encoded picture's WME results. This functionality achieves both wide search range and motion vector accuracy according to motion characteristics of encoding videos.

2.3.4 Aggregated multi-block-size secondary motion estimation in MME

The adaptively down sampled primary search described in the previous subsection uses 4×4 template in 1/4 or 1/8 reduced pictures, equivalent to 16×16 or 32×32 blocks. Since H.265/HEVC's maximum (and most common) CTU is 64×64, as described in Section 1.4, one 64×64 comprises sixteen 16×16s. It is therefore unrealistic to adapt the "inclusive" neighboring search which was proposed for H.264/AVC's secondary motion estimation in Section 2.2.3.

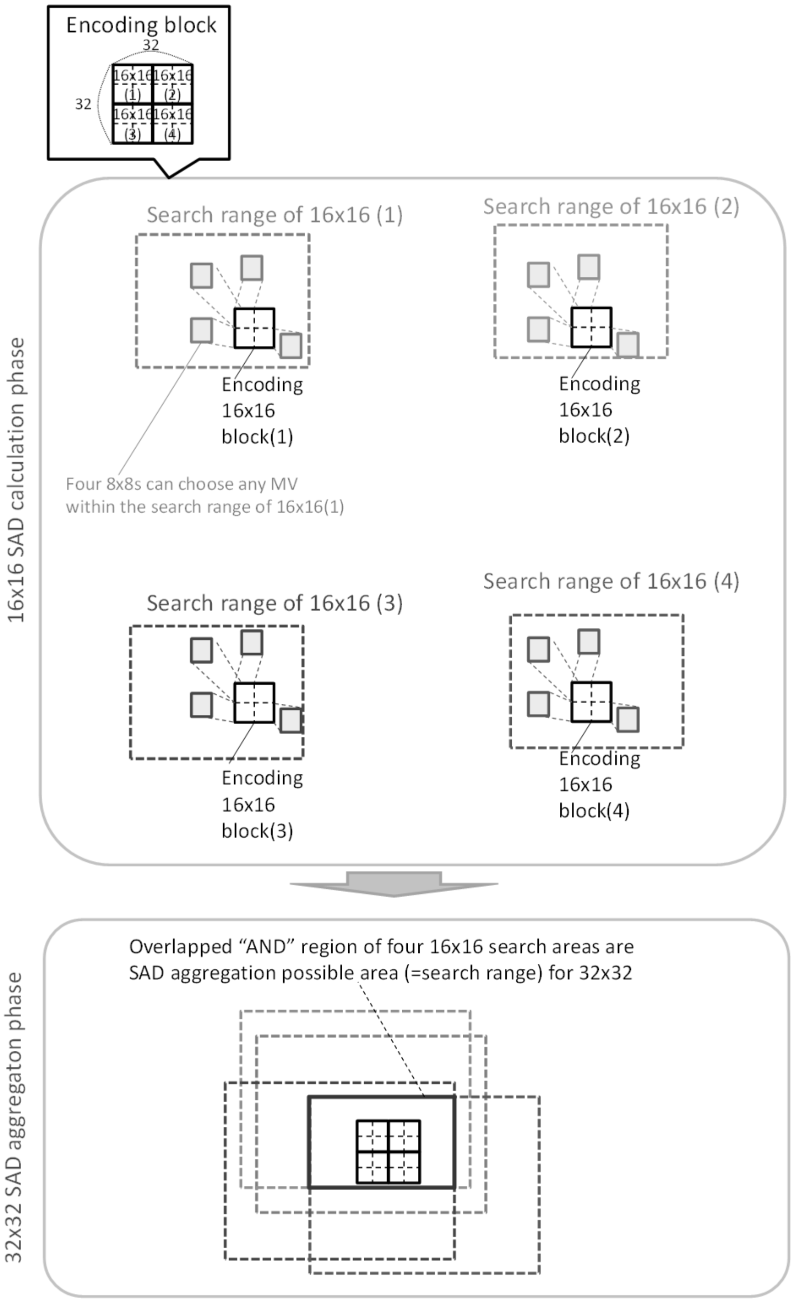

Encoding block
32
32
16x16 (1)
16x16 (2)
16x16 (3)
16x16 (4)
16x16 SAD calculation phase
Search range of 16x16 (1)
Search range of 16x16 (2)
Search range of 16x16 (3)
Search range of 16x16 (4)
Encoding 16x16 block(1)
Encoding 16x16 block(2)
Encoding 16x16 block(3)
Encoding 16x16 block(4)
Four 8x8s can choose any MV within the search range of 16x16(1)
32x32 SAD aggregaton phase
Overlapped "AND" region of four 16x16 search areas are SAD aggregation possible area (=search range) for 32x32

Figure 2.12 Concept of aggregated motion estimation.

Instead, in order to achieve multi-block-size neighboring search in H.265/HEVC, an aggregated SAD calculation is proposed, the concept of which is illustrated in Figure 2.12.

Neighboring motion estimation is performed in every 16×16 block size and SAD values are calculated at each search point. And with one 16×16 block size, four 8×8 blocks can each have any motion vectors within the 16×16 search range. This allows 8×8 blocks to represent fine movement of picture objects as long as they are within the search range of 16×16. In H.264/AVC's secondary motion estimation which is described in Section 2.2.3, neighboring search of fractional-pel precision is limited to ±0.75, however, in the proposed H.265/HEVC secondary search, neighboring search range is extended to ±3 or ±7 in double-pel precision, which is equivalent to ±6 pixels or ±14 pixels in integer-pel precision as described later.

When SAD values in all 16×16 blocks' search range are obtained, subsequently SAD aggregation for motion vector search of 32×32 and larger is performed. As described at the bottom half of Figure 2.12, overlapped "AND" region of four 16×16 search areas is a SAD aggregation possible area (=search range) for 32×32 blocks and motion vector of 32×32 blocks are evaluated in the range of this overlapped "AND" region. This approach is reasonable because when the block size of 32×32 is chosen at a final mode decision, this means that objects within this 32×32 block are still or moving uniformly with the same motion vectors, four 16×16s' motion search area therefore are anticipated to be well overlapped to enable 32×32 block motion search possible. On the contrary, if the motion search range of four 16×16 blocks is scattered and overlap of four search range is very little, this means that objects within this 32×32 block are moving apart and motion vectors of 16×16 blocks are more likely to be chosen than this 32×32 block. In the same way, SAD values of a 64×64 block can be aggregated with aggregated SAD of four 32×32 blocks. With this technique, heavily overlapped SAD calculation for motion search of multiple block size is reduced.

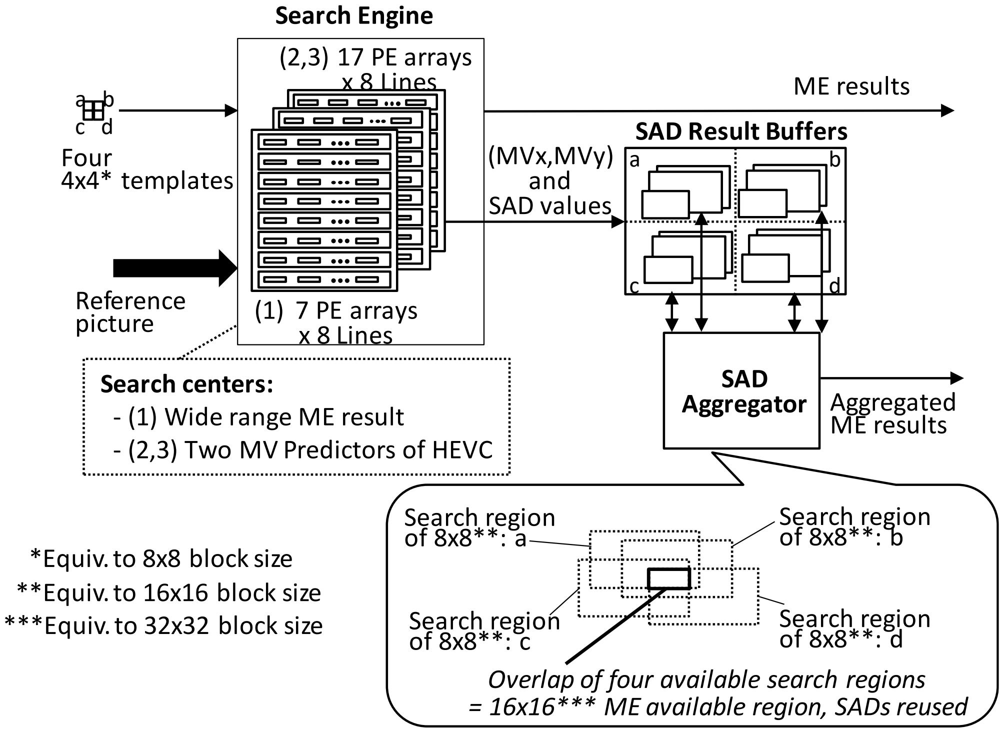

Search Engine
(2,3) 17 PE arrays x 8 Lines
a b
c d
Four 4x4* templates
Reference picture
(1) 7 PE arrays x 8 Lines
Search centers:
- (1) Wide range ME result
- (2,3) Two MV Predictors of HEVC
ME results
SAD Result Buffers
(MVx,MVy) and SAD values
SAD Aggregator
Aggregated ME results
*Equiv. to 8x8 block size
**Equiv. to 16x16 block size
***Equiv. to 32x32 block size
Search region of 8x8**: a
Search region of 8x8**: b
Search region of 8x8**: c
Search region of 8x8**: d
Overlap of four available search regions = 16x16*** ME available region, SADs reused

Figure 2.13 SAD aggregation and reusing scheme in MME.

In the actual secondary multi-block-size ME (MME), SAD is calculated for three motion search areas. The first two is centered on motion vector predictors which are derived from adjacent blocks' motion vectors, where the derivation process of the two is specified in the H.265/HEVC standard. As for motion vectors of adjacent blocks, if the final fixed coding modes and motion vectors are obtained at the time, the fixed motion vectors are used and if not, previously obtained double-pel precision motion vectors within the MME are used for motion predictor calculation.

Figure 2.13 shows the structure of internal PE arrays and a SAD value holding buffer in the MME. The search engine calculates SAD for 4×4 templates with 1/2 down sampled images (equivalent to 8×8 block motion estimation). Three search centers are utilized for motion estimation. One is the result of WME with a 7×7 search range. The other two with 17×17 search ranges are MV predictors from adjacent above and left blocks' MVs, which are defined in the HEVC standard. Calculated SAD results are temporarily stored in the

SAD result buffers, and the SAD aggregator creates SAD values of larger block sizes by using overlapping search regions, as illustrated at the bottom of Figure 2.13. This scheme avoids SAD calculation of blocks larger than 8×8 and reduces the total SAD computational load to 1/4, while still being able to track both distributed small block motion and uniform large block motion.

SAD values from 8×8 through 64×64 are thus obtained, and best motion vector for each block is selected by choosing the motion vector with minimum coding cost. Coding cost is typically estimated with

$$Cost = SAD + \lambda \times BitCost$$

where λ is a lambda value for rate-distortion optimization [36] and BitCost is a bit amount consumed for motion vector description.

2.3.5 Fractional tertiary motion estimation in FME

The FME, which performs the fractional tertiary motion estimation, has three motion estimation engines in parallel, one for 16×16 block sizes, one for 32×32 block sizes and the rest is 64×64/8×8 block size compatible. With these motion estimation engines, a fractional motion estimation combination of

Smaller block size: 8×8, 16×16 and 32×32

or

Larger block size: 16×16, 32×32 and 64×64

can be selected for each CTU. In order to choose the combination, coding cost summation of smaller three and larger three block sizes obtained in the MME is calculated and the combination with less coding costs are selected.

2.3.6 Statistical intra mode prediction in MED and IPD

In real-time H.265/HEVC encoding, intra prediction with a sharply increased combination of edge directions and block sizes is another substantial issue as discussed in Section 1.4. Before final mode decision issues are discussed, complexity reduction technique which is utilized in intra prediction should also be discussed and thus presented here.

In H.265/HEVC’s intra prediction, pixel values are predicted by copying surrounding pixel values with appropriately angled directions. Thirty-three angular directions (and additional DC and planar predictions) are defined in the HEVC standard, and encoders must evaluate and find the optimal directions for all combinations of intra block sizes.

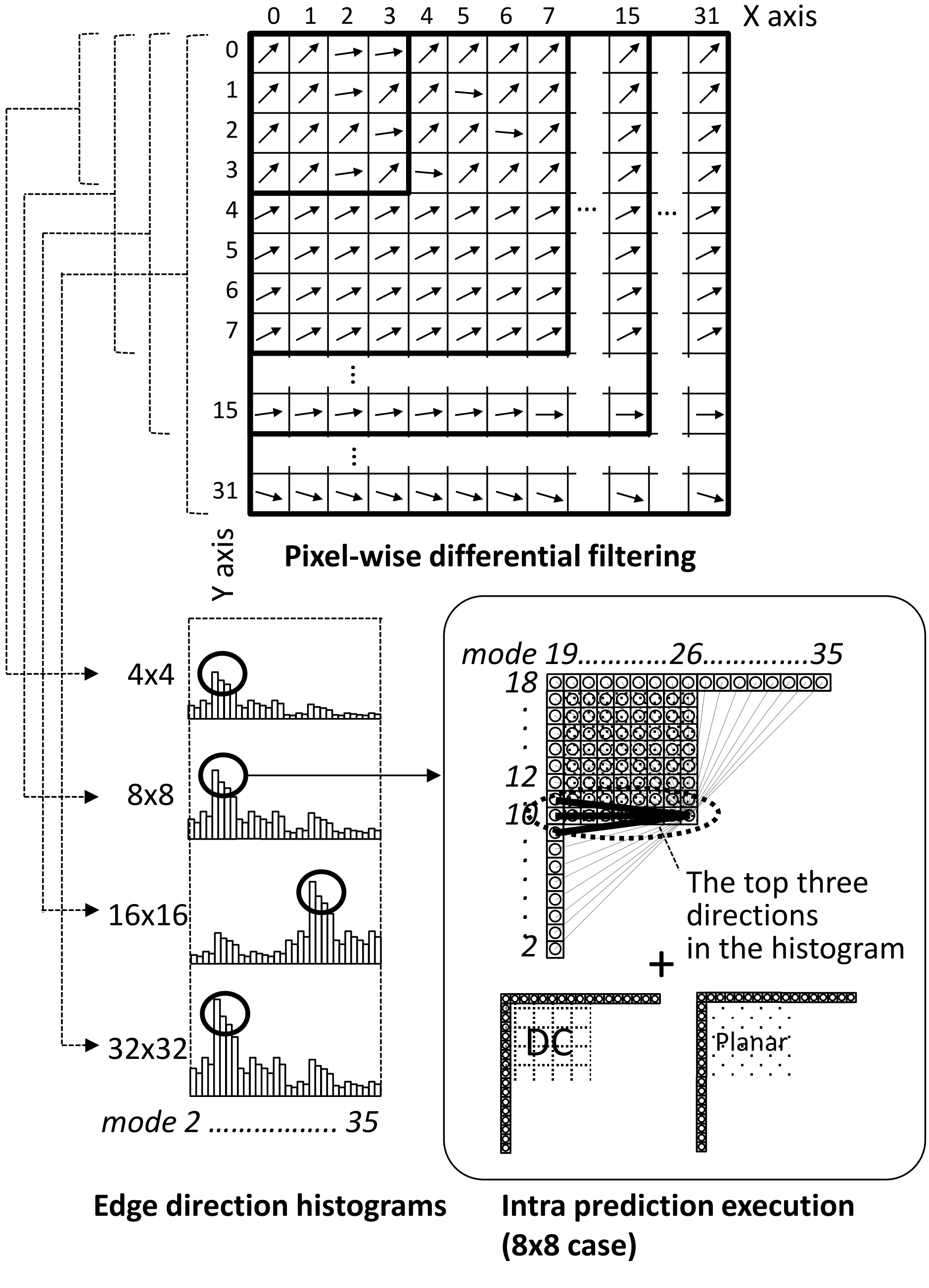

0 1 2 3 4 5 6 7 15 31 X axis
0
1
2
3
4
5
6
7
15
31
Y axis
Pixel-wise differential filtering
4x4
8x8
16x16
32x32
mode 2 35
Edge direction histograms
mode 19...........26.............35
18
12
10
2
The top three directions in the histogram
+
DC
Planar
Intra prediction execution (8x8 case)

Figure 2.14 Statistical intra prediction.

Due to the nature of the HEVC's angular prediction method, intra angular directions and picture objects' edge directions correlate strongly [37] [38]. To efficiently reduce intra prediction candidates, pixel-wise differential filtering and statistical edge direction analysis before the intra prediction are therefore adopted, the procedures of which are as follows.

First, in MED, a five-tap differential filtering is performed throughout the encoding picture, and picture edge directions for each pixel are calculated. These edge directions are then used to form an edge direction histogram for each block size from the smallest (4×4) to the largest (32×32)[9], as illustrated in Figure 2.14. Note that histogram values for the smallest 4×4 blocks can be reused for 8×8 and larger blocks.

Second, in IPD, the histogram for each block size is calculated, and then intra angular mode candidates are pruned such that prediction cost is evaluated for only the top three directions of the histogram, plus DC (prediction with the non-directional average value) and planar (prediction with non-directional curved plane values) prediction modes. Other angular directions are ignored.

This edge-based angular prediction pruning reduces intra prediction computation of IPD to 1/7 (from 35 modes to 5 modes, including DC and planar). Coding efficiency degradation due to this intra prediction pruning is assessed later in Section 2.3.6.

2.3.7 Deeply centered mode decision structure in IIM

H.265/HEVC uses an enhanced motion vector and encoding mode inference techniques from spatially adjacent blocks to maximize the coding efficiency. Parallel and pipelined architectures of conventional high-throughput hardware designs inherently degrade the coding performance, because encoding mode candidates must be pruned and selected despite adjacent motion vectors and modes still not fixed. Therefore, coding costs of selected candidates can fluctuate afterwards and may not be optimal.

To solve this problem, deeply centered mode decision is introduced. On the basis of this scheme, evaluated prediction modes are not pruned during the pipelined motion estimation processes, but maintained as much as possible to be used for a highly sequential mode decision procedure for intra-inter mode decision (IIM), as illustrated in Figure 2.15. From IPD, calculated intra prediction costs for each block size are transferred as an intra cost

[9] In H.265/HEVC, 64×64 intra mode exists but it is considered as four 32×32 modes with the same prediction direction modes. Independent evaluation of the 64×64 intra mode is therefore omitted and it is selected only when four of 32×32 intra modes are selected for final best modes and they happen to have the same prediction direction.

dataset. Also, from fractional pixel motion estimation (FME), inter prediction costs of three different block sizes are transferred as an inter cost dataset. These costs are calculated in parallel, and have tentative values with unfixed adjacent block modes. In IIM, final encoding modes are fully decided sequentially in an order that conforms to the HEVC standard, and costs are re-calculated by referring to the adjacent block MVs and modes that are all fixed. This high-speed mode decision loop enables final mode decision with precise encoding costs, preventing coding efficiency from degrading.

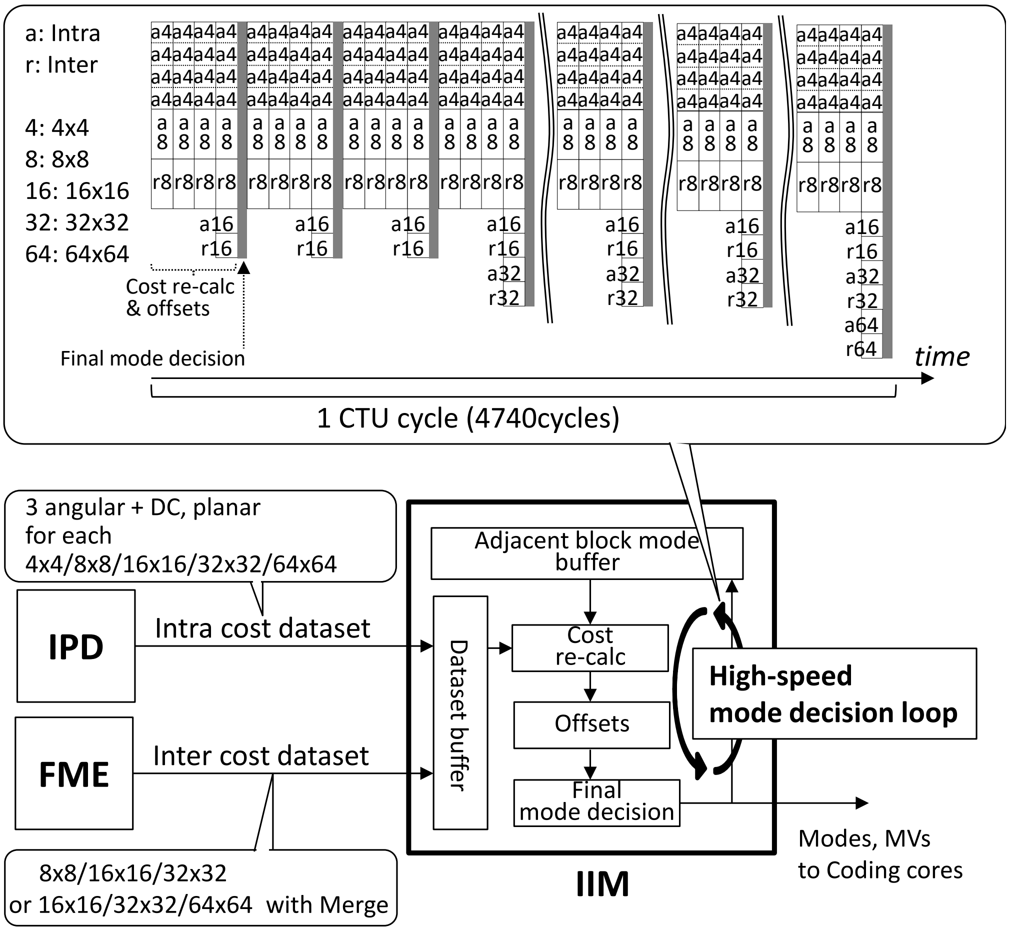


Figure 2.15 Deeply centered mode decision scheme

To guarantee programming flexibility, PRISC software can set cost offsets to each mode to control the final mode judgment, described later in Section 2.3.9.

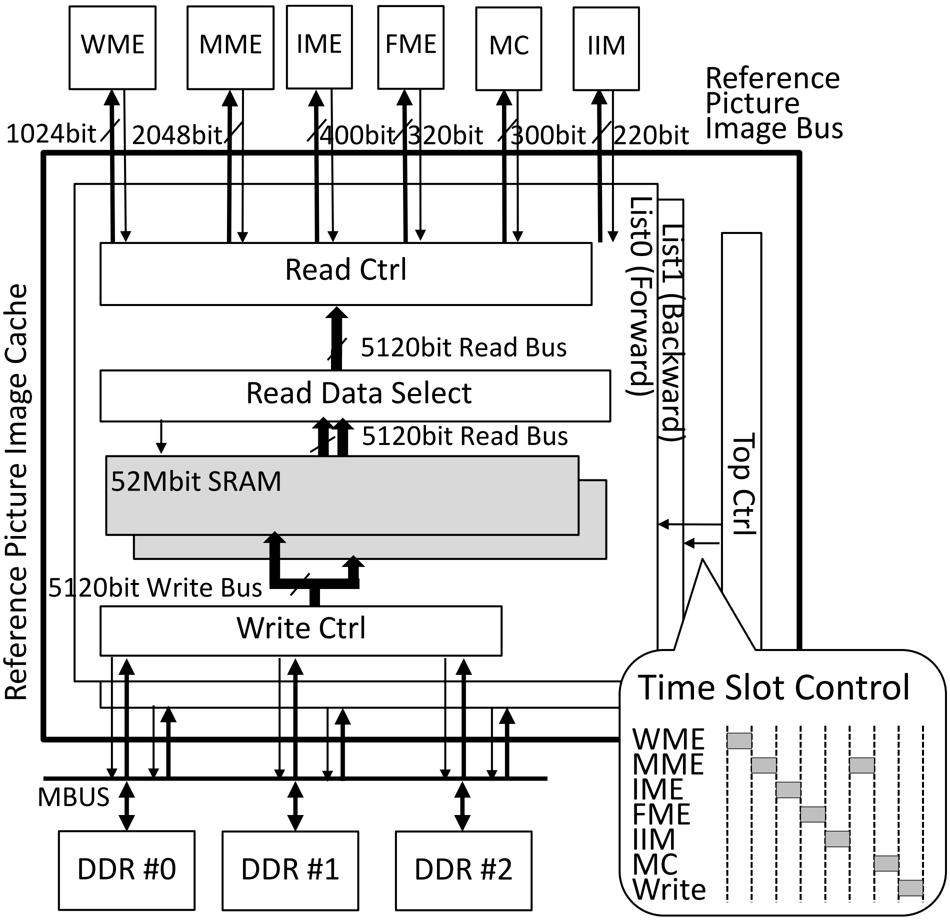


Figure 2.16 High-speed reference image feed.

2.3.8 High-speed reference image feed

In the adaptive and aggregated motion estimation scheme, pipelined ME engines from WME through FME and motion compensation (MC) demand reference image feeds with very high bandwidths in order to retrieve image data with various motion vectors, various block sizes and multiple reduction ratios. Reference image caching [39] is therefore essential for ultra-high definition encoders, and our approach makes maximum use of ME engines with distributed multi-block-size motion vectors. Figure 2.16 shows the reference picture image cache configuration and the bus connection to achieve this requirement. Ten bit/pixel reference picture fragments are read from external DDR-SDRAMs to four 52 Mbit

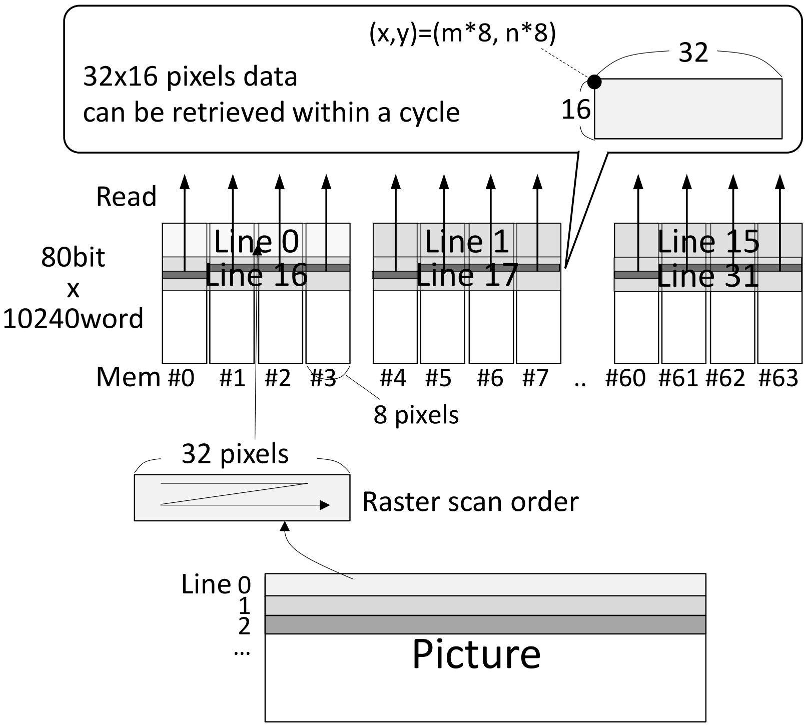


Figure 2.17 Reference picture storage method in the SRAM.

image caches. At the beginning of encoding each picture, write control occupies the cache's R/W ports for rapid cache fulfillment, and afterwards the R/W ports are time slot controlled and reference picture fragments are updated as encoding proceeds. Each image cache comprises 64 single-port SRAMs of 80 bits × 10,240 words. Pixel data are stored so that any 32×16 pixel region (with X and Y coordinates that are multiples of eight) can be retrieved in one cycle from 64 SRAMs with a 5120-bit read bus, as depicted in Figure 2.17. This wideband connection makes it possible to react to 797-Gbps reference image demand from ME engines and MC with distributed motion vectors of variable block sizes without limiting motion vector variations, which is essential for precise motion estimation of H.265/HEVC's variable block size inter prediction.

2.3.9 Operational flexibility

The above-mentioned function blocks that comprise prediction and coding cores are built by using dedicated hardware engines to achieve high throughput for ultra-high definition TV.

However, software-defined function controllability is also indispensable for operational flexibility to gradually improve encoding quality and achieve supplemental functions such as high dynamic range (HDR) and wide color gamut (WCG). For this purpose, PRISC and CRISC processors work closely with dedicated blocks below H.265/HEVC's picture coding unit (CTU) level, while MRISCs run above HEVC's slice level and TRISC runs above its picture level, with coarse connection with the prediction and coding cores. Functions are easily built and changed with C language. This hierarchical software structure contributes to the flexible software controllability of high-throughput hard-wired encoding engines.

In the IIM which is described in Section 2.3.7, deeply centered mode decision structure inevitably limits operation cycles to be allowed for each mode decision (a few clocks for each 4×4 block) and makes intervention by the RISC CPU difficult. Instead, the IIM's hard-wired mode decision calculator has the scale and offset value registers for each coding mode, which allows linear transform of estimated coding costs before final mode decision. These registers are rewritable from RISC CPUs for each CTU cycle (usually in responding to picture characteristic analysis results provided by the IFE, interpreted by RISC CPUs) and this functionality allows the IIM's operational flexibility.

2.3.10 Features in the proposed H.265/HEVC prediction core

The H.265/HEVC prediction core architecture which is described above has the following features for each requirement described in Section 2.1:

(1) Support for motion vectors with large motion: The WME with statistically adaptive search center transition and down scaling ratio switching helps tracking capability of large motion in picture objects with limited search range, while maintaining motion tracking accuracy.

(2) Support for motion vectors of multiple block sizes: The MME with aggregated SAD calculation of over 8×8 blocks substantially reduces overlapped SAD computation between multiple block sizes. Moreover, the reference image feed which responds to the heavy demand of ME engines allows distributed motion vectors of variable block sizes without limiting motion vector variation.

(3) Support for precise encoding mode decision: In addition to the MEs mentioned above, intra prediction in MED and IPD efficiently reduces computational

complexity while effectively creates intra prediction evaluation results from all block sizes from 4×4 through 32×32. Besides, deeply centered mode decision scheme enables highly sequential evaluation in combination of multiple block sizes, modes and motion vectors, resulting in accurate coding cost calculation and precise final mode decision.

(4) Support for flexible mode decision adjustment: RISC CPUs' programmability helps operational flexibility of the proposed prediction core, while scale and offset registers for each coding mode in the IIM helps adjustability of encoding mode decision with RISC CPUs' control.

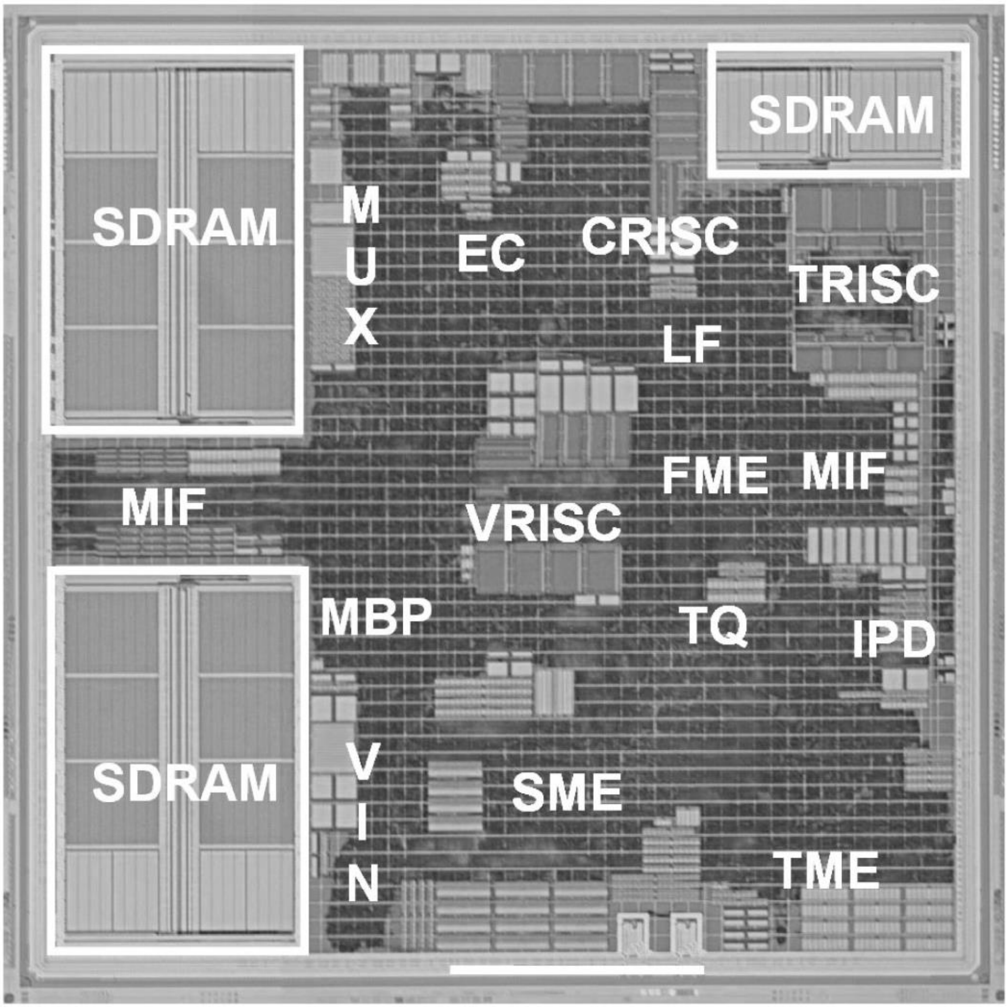


Figure 2.18 H.264/AVC video encoder VLSI "SARA" chip micrograph.

Implementation and encoding quality performance of the proposed techniques are described later in Section 2.6.2.

2.4 Chip configuration and fabrication

2.4.1 H.264/AVC video encoder VLSI "SARA"

We have implemented the search and mode decision modules mentioned in Section 2.2

into our H.264/AVC encoder VLSI that supports High 4:2:2 Level 4.1. The chip micrograph is shown in Figure 2.18 and chip specifications are listed in Table 2.1. The FME/MC and mode decision block is labeled as “SME” in Figure 2.18, with 8.2 million transistors. Together with the telescopic IME engine labeled “TME” (telescopic search) and “FME” (neighboring ±1 search), the chip contains 257 GOPS ME/MC engines with search range -271.75 to +199.75 (H) / -109.75 to +145.75 (V). The chip was fabricated in a 90-nm CMOS process, with 140 million transistors.

Figure 2.19 shows the appearance of this VLSI and an HDTV encoder module mounted with 6 chips. The HDTV encoder module supports 1080 / 60i HDTV video coding with a high 4: 2: 2 profile and level 4.1 by the multi-chip coding configuration, which details are discussed later in Chapter 3. Also, an HDTV encoder device equipped with this module is

Table 2.1 H.264/AVC video encoder VLSI “SARA” chip specifications.

Technology		90-nm CMOS
Transistors		140 million
Clock frequency		max 200 MHz
Supply voltage		Core: 1.2 V, MobileDDR: 1.8 V, eDRAM: 2.5 V, I/O: 3.3 V
Power consumption		3.0 W
Package		625-pin FCBGA (21x21 mm)
eDRAM		72 Mbit
External memory		512 Mbit (32-bit width) Mobile DDR
Video	Profile	H.264: Main / High / High422 (8bit) MPEG-2: Main / 422 P MPEG-4: Simple
	Level	H.264: 3.0 / 4.0 / 4.1 MPEG-2: ML / H14L / HL
	Resolution	Single chip: 720x480 30 fps Multi chip: 1920x1080 30 fps
	Structure	Frame, Field, PAFF, MBAFF
	ME	-271.75 / +199.75 (H) -109.75/+145.75 (V) max 4 reference frames 16x16/16x8/8x16/8x8 block Explicit WP, Spatial/temporal direct
	Transcoding	Available with embedded/external picture/macroblock information
Audio	Input	Linear PCM or encoded AAC
User data	Input	PES format for supplemental data/audio
System	Output	MPEG-2 TS (188/204byte) max 120 Mbps

shown in Figure 2.20. High quality coding is realized by wide motion detection range and continuous improvement of mode decision control algorithm. This module is adopted for a wide range of applications such as satellite relay of broadcast material video using H.264 / AVC and IP transmission.

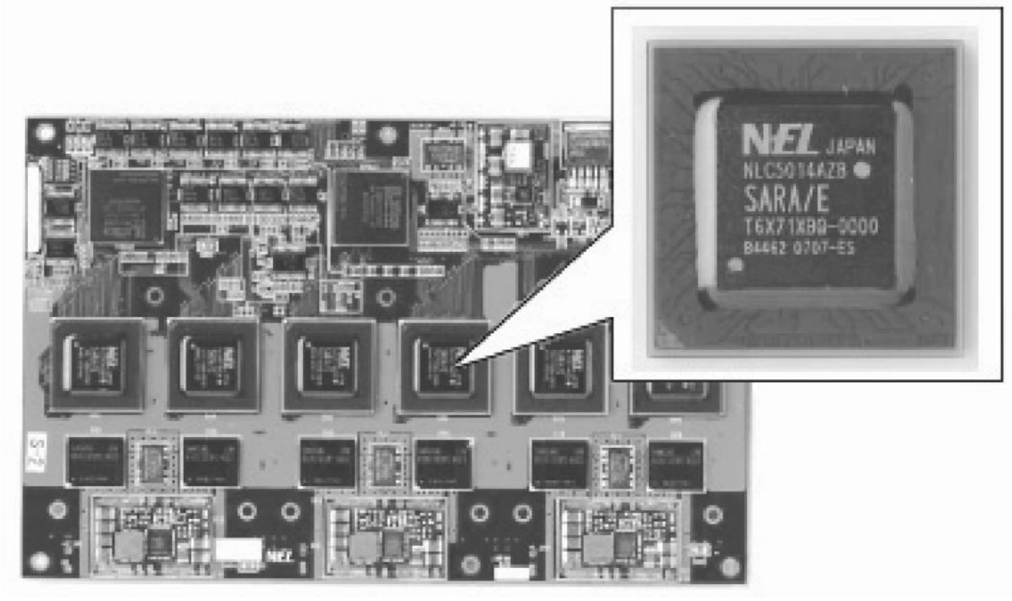


Figure 2.19 A real-time HDTV encoder module with six SARAs.

Figure 2.20 A real-time HDTV encoder device with SARAs.

2.4.2 H.265/HEVC video encoder VLSI “NARA”

The prediction core including the architecture described in Section 2.3 was successfully developed by using SystemC and high-level synthesis. This NARA VLSI’s layout with 28

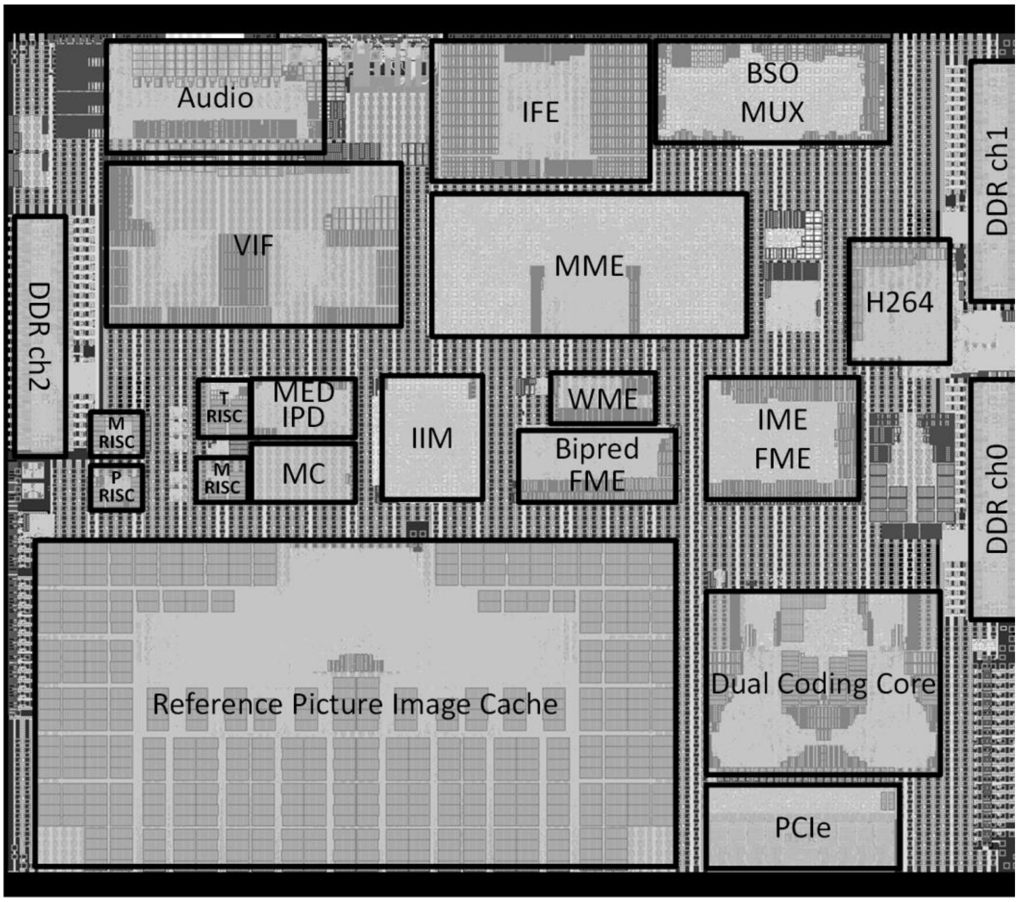


Figure 2.21 H.265/HEVC video encoder VLSI "NARA" chip layout.

nm CMOS technology is shown in Figure 2.21, and a fabricated chip photo is shown in Figure 2.22. Physical features of the NARA VLSI are described in Table 2.2. Power consumption of a video core that comprises the proposed prediction and coding cores is adaptive with respect to encoding picture sizes and frame rates; at 4K 60 fps it is around 7W and at 8K 60 fps, it is around 28W with four NARA VLSIs, which are acceptable for video coding LSIs for professionals.

Various encoder systems beyond the current HDTV have been developed with the NARA VLSI, providing real-time HEVC encoding functionalities to TV broadcasters and distributors. Figure 2.23 depicts a 4K H.265/HEVC encoder with one NARA VLSI and an 8K HEVC encoder that integrates four NARA VLSIs operating in parallel. Both can encode 60 fps ultra-high definition images into H.265/HEVC streams in real time, enabling 4K and

8K TV contribution and broadcasting. Currently, bit rates of 25-35 Mbps are required for 4K encoding, and 85-100 Mbps are used for 8K encoding. However, these can be further reduced by improving picture quality control algorithms at the PRISC and CRISC CPU firmware. The CPU firmware flexibility of the NARA VLSI also supports various encoder configurations, one of which is a high-frame-rate 2K/120 fps HEVC encoder [40].

Table 2.2 H.265/HEVC encoder VLSI “NARA” physical features.

Technology	28 nm CMOS
Num. of transistors	83 Mgates
Clock frequency	Max 600 MHz
Supply voltage	Core: 0.9 V IO: 1.8/3.3 V DDR3: 1.5 V PCIe and 3G-SDI: 0.9/1.8 V
Power consumption	Approx. 15.0W
Package	1152 pin FCBGA (35 x 35mm)
External memories	DDR3 Max 3ch

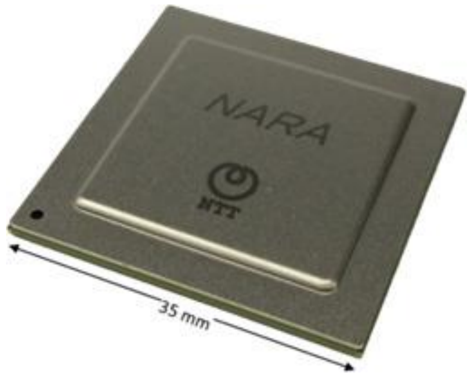


Figure 2.22 H.265/HEVC video encoder VLSI “NARA” chip photograph.

Figure 2.23 4K and 8K H.265/HEVC encoder devices.

2.5 Related work

Intensive work has been done to overcome these requirements recently, especially for H.265/HEVC encoding of very large images beyond HDTV. At first for intra prediction, a H.265/HEVC intra prediction method for 8K images has been proposed [41], but the claimed performance is for only decoding and not encoding. H.265/HEVC intra prediction for encoders has also been proposed [42], however, its picture size is up to HDTV images.

For H.265/HEVC inter-frame prediction, 4K and 8K motion search engines have been studied [43] [44] with the maximum search limited to ±64, which is insufficient for ultra-high definition videos with large motions. Other studies [45] [46] propose an encoder chip and its motion estimation architecture with images up to 8K 30 fps. Their implementation, however, does not support processing of the smallest coding unit (CU) of 8×8 due to high bandwidth demand, and they are hard to accept in professional use because a smaller CU plays a key role in the quality of complex images. Another SoC implementation [47] also limits the maximum search range to 64×32. Techniques to overcome computational complexity, memory bandwidth, and data dependency problems for ultra-high definition codec have been studied [48]. However, they have not yet been applied to an H.265/HEVC encoder SoC.

In addition to SoCs, highly parallelized FPGAs [49] and CPUs [50] [51] have also been utilized. Chassis size and power consumption of their overall encoder systems, however, inevitably becomes higher in exchange for the deep parallelism, which is unacceptable for

Table 2.3 Comparison of H.265/HEVC video encoder VLSIs.

	The NARA VLSI	ICCE2016 [47]	ASPDAC2014 [42]	VLSIC2013 [46]
Tech	28nm CMOS	28nm CMOS	90nm CMOS	28nm CMOS
Supply voltage	0.9V	N/A	0.9V	N/A
Clock frequency	600MHz for HEVC 4K 60fps	600MHz	357MHz	312MHz
Encoding capability	4K 60fps HEVC(up to Main 4:2:2 10 Profile) scalable to 8K 60fps, H.264	4K 30fps HEVC	HDTV 44fps HEVC (Intra)	8K 30fps HEVC
HEVC supported mode	CU Size 8x8-64x64 Frame / Field / Super Low Delay / Multi Stream	CU Size 8x8-32x32 Frame only	CU Size 8x8-32x32 Intra Frame only	CU Size 16x16-64x64 Frame only
Motion search range	{-3847.75,+3847.75}/ {-1926.75,+1926.75}	64x32 around MV predictor	N/A	{-512,+511}/ {-128,+127},
Reference picture feed throughput	797Gbps	N/A	N/A	347Gbps

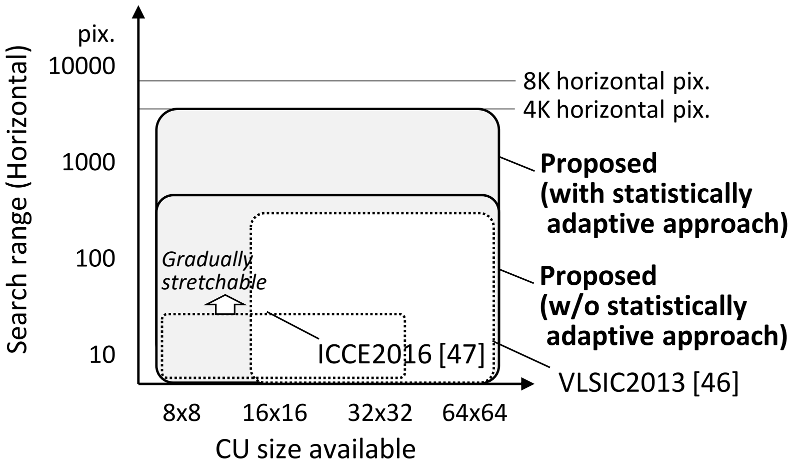


Figure 2.24 Performance evaluation of H.265/HEVC encoder SoCs.

broadcasters aiming at on-site news gathering with mobile devices. Thus, bringing about a complete broadcasting-grade HEVC encoding capability for 4K and 8K ultra-high definition has been still a challenge.

Comparisons with state-of-the-art H.265/HEVC video encoders [42] [46] [47] are summarized in Table 2.3. The NARA VLSI supports the latest HEVC standard and achieves single chip 4K 60 fps 4:2:2 capabilities and 8K scalability which is described in detail in Chapter 3, with higher coding complexity that supports CU sizes from 8×8 to 64×64 and a very wide motion search range. It thus meets the functionality and quality requirements from professional TV broadcasters and content distributors.

Performances of HEVC video encoder SoCs are compared in Figure 2.24, from the viewpoint of search range and supported CU sizes. The proposed motion estimation method covers a widest search range, and the range is further stretchable with the statistically adaptive WME approach, achieving ME suitable to 4K and 8K ultra-high definition videos. The proposed prediction mode pruning techniques help support all CU sizes in the H.265/HEVC standard, which also result in a good coding efficiency suitable for professional broadcasting quality.

2.6 Coding quality evaluation

2.6.1 Coding quality of H.264/AVC prediction core

In order to evaluate the coding efficiency of the proposed “telescopic and inclusive” motion estimation method for H.264/AVC, a software simulation for three different motion estimation and mode decision algorithms was done.

(a) Full search with H.264/AVC Joint model (JM) reference software (developed for standardization activity of H.264/AVC)

(b) Proposed motion estimation (telescopic primary search and inclusive secondary search)

(c) Full search, mode decision with integer-pel search results [35]

The motion search range was ± 24 pixels for the full search in (a) and (c), and for the telescopic search in (b), the search for ± 12 pixels was performed in sequence on the 1⁄2 reduced image.

For mode decision, the method of selecting the smallest "Sum of Absolute Transformed Difference[10] (SATD) + motion vector cost + reference picture number description cost", which is used in JM reference software, was uniformly applied for (a), (b) and (c).

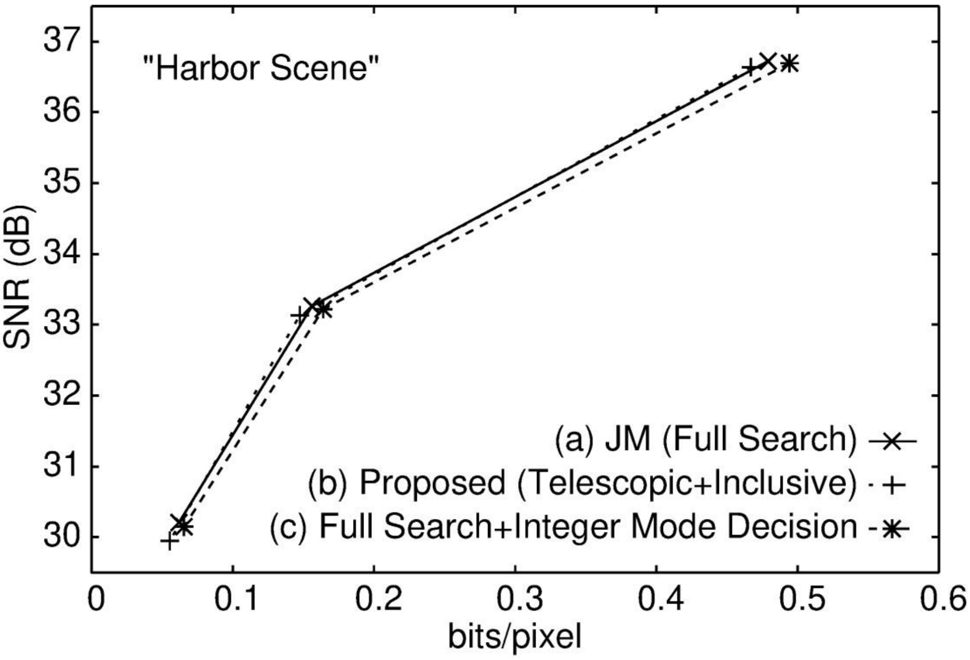
The encoding parameter was: M = 3, N = 30, 28 frames, 1920×1088 pixels, Main profile, Field coding, spatial direct, CABAC, RD optimization off, 8×8/8×16/16×8/16×16 block sizes.

The encoding results for three HDTV sequences “Harbor Scene”, “Bronze with Credits” and “Yachting” are shown in Figure 2.25. The horizontal axis represents coding bits per pixel, and the vertical axis represents the average SNR of luminance samples. When plots are compared to the same bits per pixel (i.e. with one vertical line in the graph) as illustrated in Figure 2.26 (a), the difference with the same bits/pixel is obtained and the upper (the more PSNR with the same bits/pixel), the better. And when plots are compared with the same PSNR (i.e. with one horizontal line in the graph) as illustrated in Figure 2.26 (b), the difference with the same PSNR is obtained and the more left (the little bits with the same

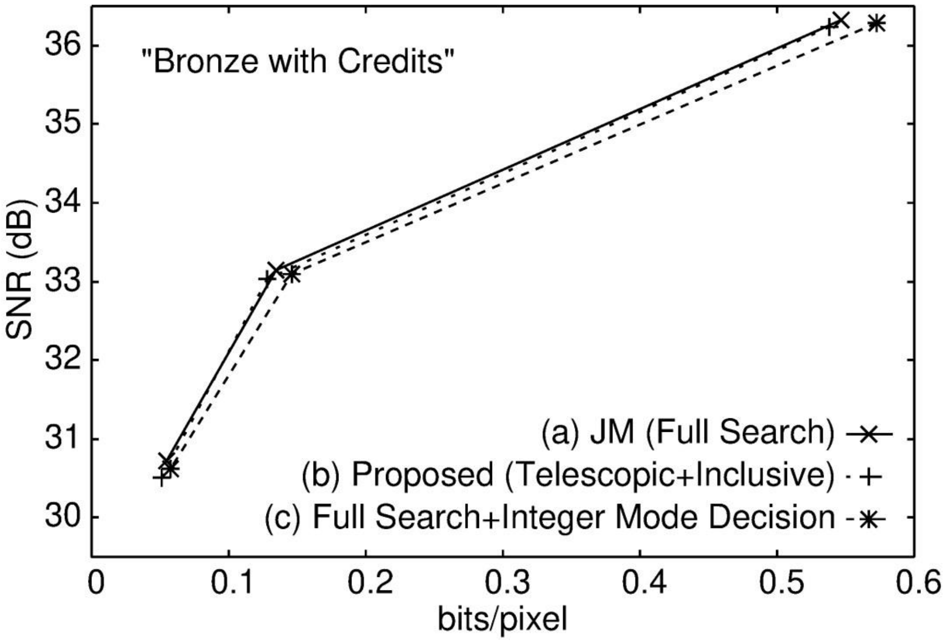
[10] SATD calculates sum of absolute difference after Hadamard transform. Hadamard transform emulates DCT transform results with simple addition and subtraction, which can estimate the coding cost of residual information better than SAD.

PSNR), the better. In general, when plots go upper and more left, the better the coding efficiency.

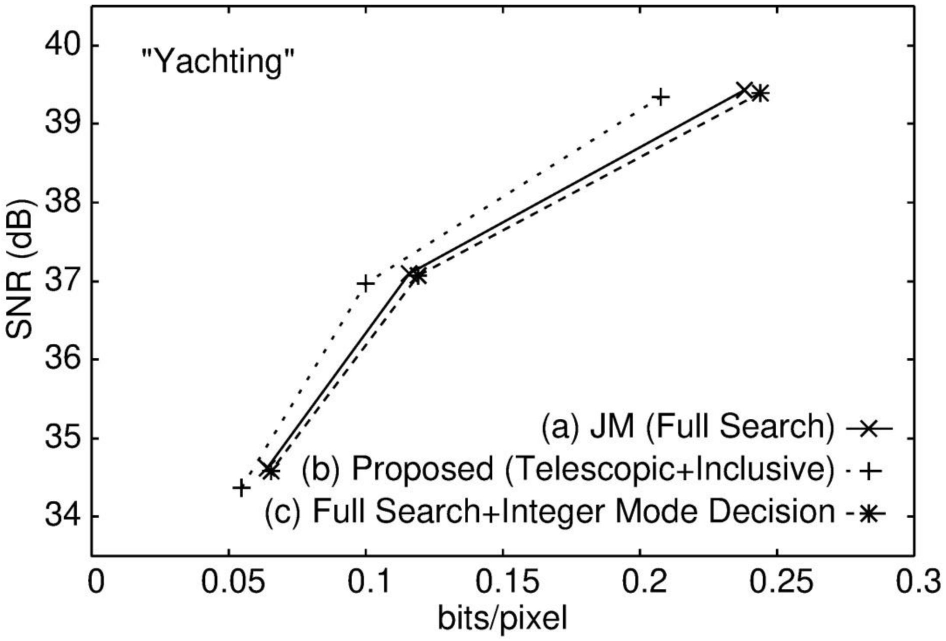
In the sequences with low or medium motion, such as “Harbor Scene” or “Bronze with Credits”, JM's full search method (a), which does exhaustive integer and fractional motion estimation for all block sizes, performed the best. The complexity reduction method (c), which processes block mode decisions at integer-pel precision and does fractional motion estimation for only one block size, degrades coding efficiency by up to 0.5 dB. The proposed search method (b) achieves just 0.1 dB lower coding performance than that of JM, by virtue of the inclusive fractional motion estimation. Also, because of the motion tracking capability telescopic primary search has, our method even outperforms JM with scenes containing faster motion, such as “Yachting” with large horizontal motion in pictures. These results show that the proposed motion estimation in H.264/AVC’s prediction core has sufficient coding quality compared to the full search method with the reference software.



i) Harbor Scene

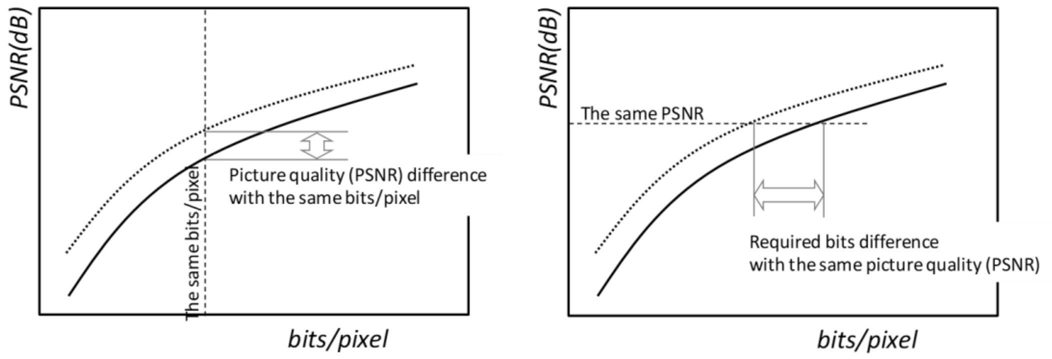


ii) Bronze with Credits



iii) Yachting

Figure 2.25 Performance evaluation of proposed motion estimation for H.264/AVC.



(a) Compare vertically (b) Compare horizontally

Figure 2.26 Meaning of coding quality evaluation results.

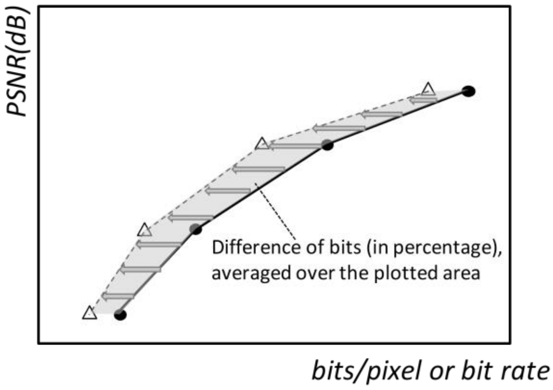


Figure 2.27 Meaning of BD-rate.

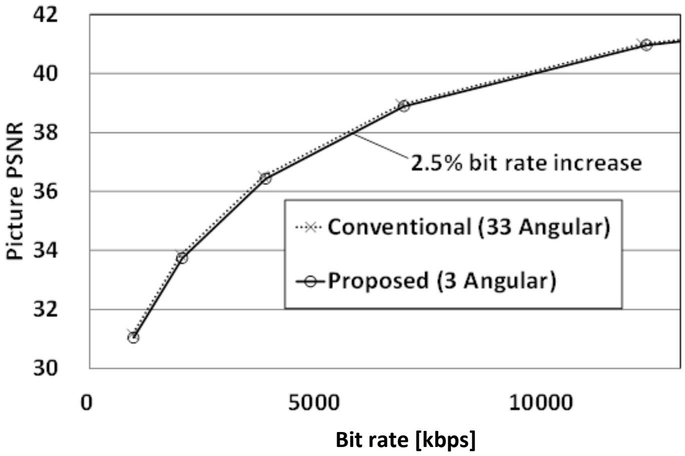


Figure 2.28 Intra prediction performance comparison (Cactus).

Table 2.4 Search performance of statistically adaptive WME.

	From “1/8 fixed” to “Adaptive”	From “1/4 fixed” to “Adaptive”
Bit rate increase/reduction	- 1.0%	- 8.2%

2.6.2 Coding quality of H.265/HEVC prediction core

In this subsection, coding efficiency of the H.265/HEVC prediction core proposed in Section 2.3 is assessed with the three steps listed as follows: 1) coding efficiency of intra prediction technique in MED and IPD, 2) effect of the statistically adaptive WME primary search, 3) relationships between the MME search range and coding efficiency, and 4) overall coding efficiency of the H.265/HEVC encoder VLSI with subjective picture quality evaluation.

First, the proposed coding efficiency of intra prediction technique in MED and IPD was evaluated, in order to clarify the coding efficiency performance of edge-based pruning of angular prediction candidates. Five HDTV sequences (Kimono, ParkScene, Cactus, BasketballDrive, BQTerrance) were tested and a software simulator of H.265/HEVC encoder VLSI was used for the evaluation. The proposed technique with three angular modes calculated in IPD was compared to a full assessment of all 33 angular modes calculated in IPD. Due to the strong correlation between intra angular directions and edge histograms, the simulation results of Cactus show that this method only has a 2.5% higher bit rate than the full 35-mode assessment with the same picture PSNR as depicted in Figure 2.28[11]. Other four sequences have less bit rate increase. These results show that the proposed edge-based intra prediction mode pruning technique works well with real-time H.265/HEVC encoder VLSI design.

Second, in order to evaluate the effect of the proposed statistically adaptive WME primary search, another software simulation was performed to compare statistically adaptive WME versus WME with fixed search center and the same down sampling ratio. Here an 8K video with large motion of nine frames (SL) was used in order to evaluate the motion tracking capability with the highest picture resolution. Coding efficiency is evaluated with BD-rate [52], which denotes the percentage of extra coding bits (positive value) or reduced coding bits (negative value) required to represent the same picture PSNR as illustrated in Figure 2.27. Less BD-rate value (minus value) means less bits with the same PSNR which means more coding efficiency. The results shown in Figure 2.28 indicate that this statistically adaptive WME approach achieves both a wide search range and MV accuracy, and provides 1.0 to 8.2% coding gain for 8K video, compared to the fixed down sampling ratio of 1/4 or 1/8.

The WME's search range of ±48×±24 with 1/8 down sampled pictures (equivalent to ±384×±192 pixel motion search range) originally supports a desirable search range of ±48×8=±384 in horizontal direction[12], however, the proposed statistically adaptive WME functionality helps achieve better coding efficiency compared to a fixed manner.

[11] The horizontal axis here uses "bit rate" (i.e. bits per one second) instead of "bits/pixel" (i.e. bits per one pixel), however, the meaning of the graph is the same as depicted in Figure 2.26.

[12] ±48 is the adequate search range per 1/60 frame distance in 4K, as described in Section 2.1. And eight is the maximum picture distance of pyramid structure described in Figure 2.9.

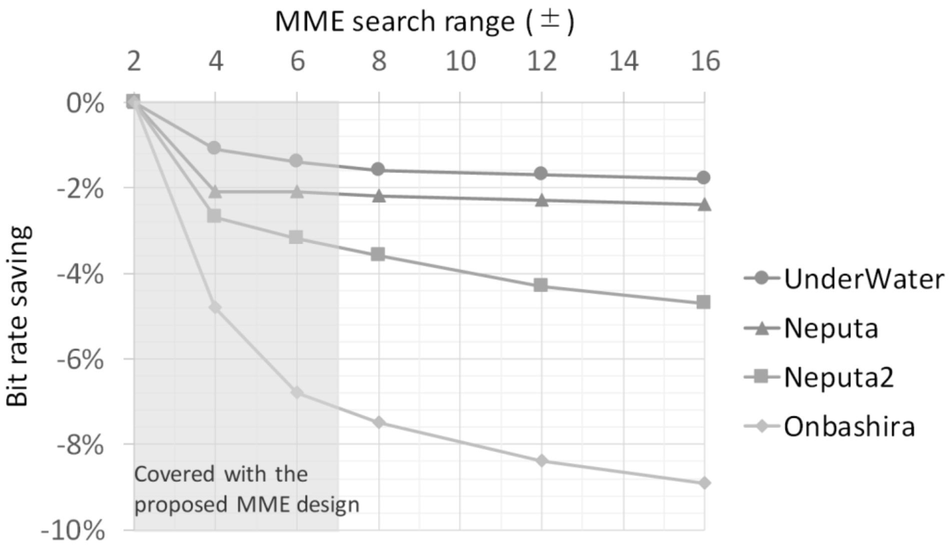


Figure 2.29 Relationships between MME search range and bit rate saving.

Third, in order to evaluate the effectiveness of aggregated multi-block-size MME secondary search, relationships between the MME search range and coding efficiency are evaluated. Four 4K video sequences (UnderWater, Neputa, Neputa2, Onbashira) were selected and encoded with different MME search range from ±2 through ±16.

Figure 2.29 shows the bit rate savings percentages using the BD-rate values, compared to the minimum search range of ±2. In other words, when the MME search range is extended over ±2, bit saving effects depicted in Figure 2.29 are obtained.

The results differ depending on the characteristics of the 4K contents. For example, "Neputa2" has large movements and "Onbashira" is one of the contents which are fulfilled with the highest motion of objects. In these sequences, the wider the search range, the better the bit saving. There seems to be no indication of “sufficient” search range. Even so, there are areas where “bit rate reduction progresses sharply as the search range expands" and where " the bit rate decreases only gently and linearly even if the search area is expanded".

The search performance of “±7” adopted by the proposed MME covers the former area where “bit rate reduction progresses sharply”. These results show that the proposed MME covers the motion search range, which efficiently reduces the bit rate for most 4K contents in general.

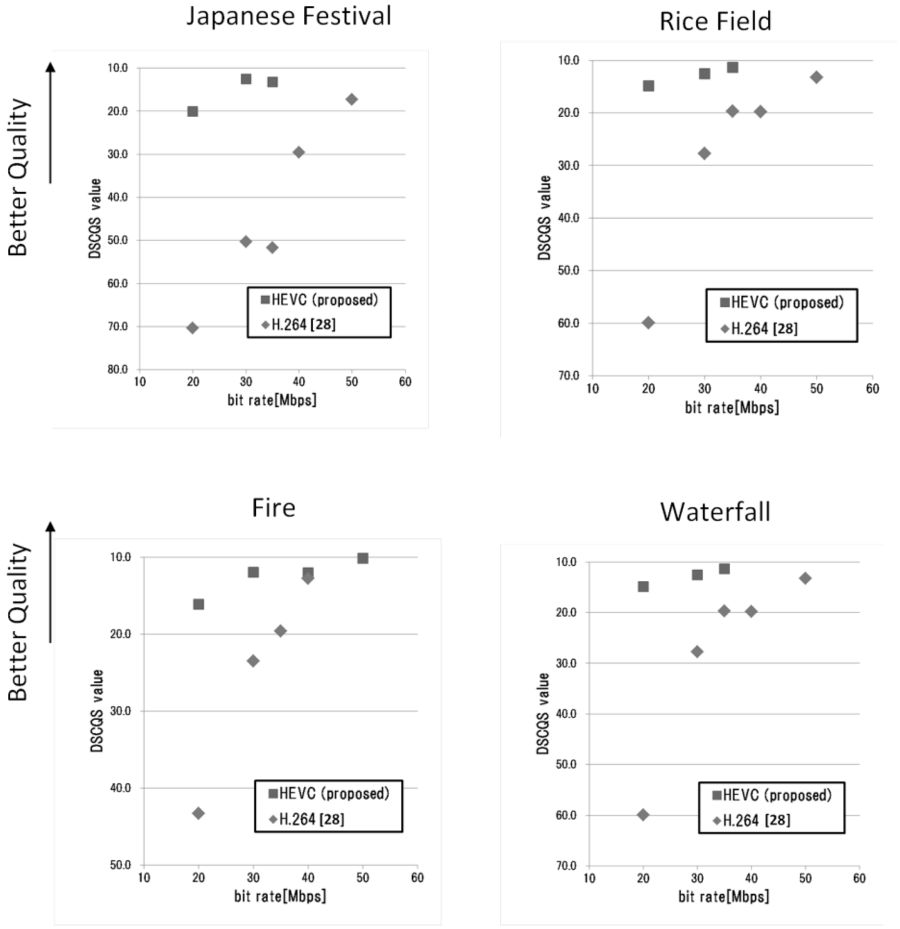


Figure 2.30 Subjective evaluation of H.265/HEVC coding quality (example of individual video).

Finally, a subjective evaluation was conducted to ascertain the H.265/HEVC encoder VLSI’s coding quality performance. For the evaluation, eight different 4K 60 frames per second video sequences lasting 10 seconds were H.265/HEVC encoded. Thirty-two people (non-experts) were then asked to compare them to sequences generated with an H.264/AVC encoder LSI [28] and score the picture quality using the double stimulus continuous quality scale (DSCQS) method [53] in Rec. ITU-R BT.500-13. Conforming to this testing standard, picture quality scores rated by subjective evaluators were statistically analyzed into DSCQS scores, which denotes the consciousness for picture degradation, in which less scores means the less picture degradation and the better quality.

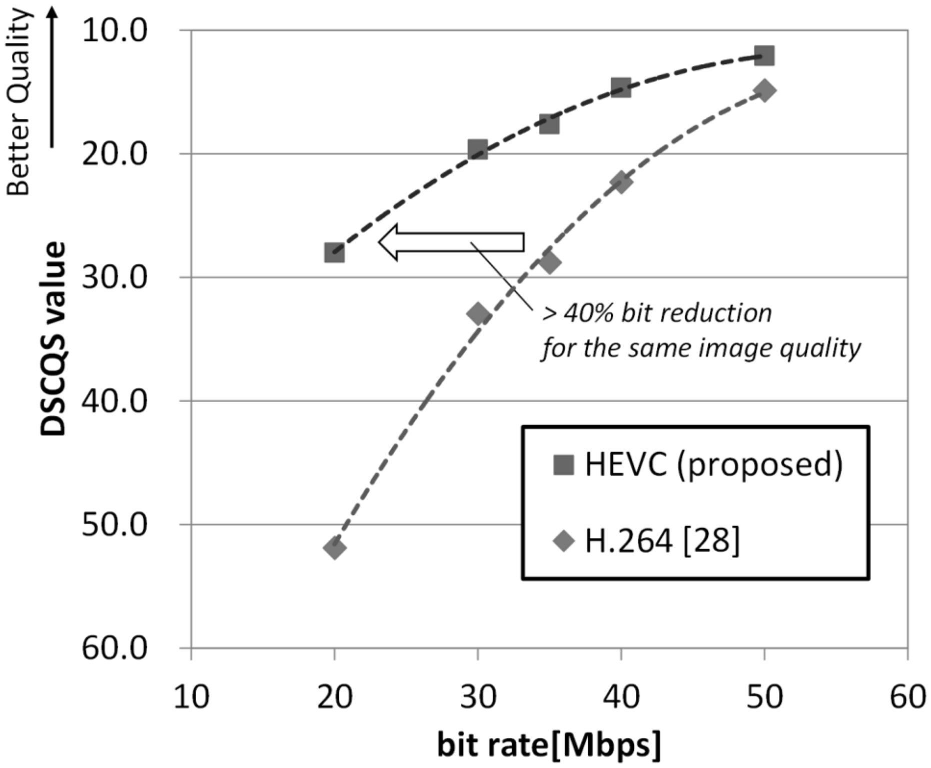


Figure 2.31 Subjective evaluation of H.265/HEVC coding quality (overall average).

Results of characteristic four sequences are shown in Figure 2.30 and the overall average of eight evaluated sequences from 20-50 Mbps are depicted in Figure 2.31. They show that the H.265/HEVC encoder VLSI reduced the bit rate by over 40%, while maintaining the visual quality compared to the former H.264/AVC encoder VLSI. The results demonstrate that the proposed architecture well achieves H.265/HEVC's coding efficiency in real-time operations and is suitable for professional quality H.265/HEVC encoders.

2.7 Chapter summary

In this chapter, video encoding core architecture of real-time video encoder VLSIs aiming at broadcast-level video quality was introduced. Especially, the "prediction core" which handles the intra/inter prediction, motion estimation and determination of prediction modes were intensely described for two standards of H.264/AVC and H.265/HEVC. For major requirements for achieving professional quality real-time video encoders were first presented and techniques to overcome them are described in detail. The “telescopic + inclusive” motion estimation with programmable fractional motion estimation and mode decision with SIMD processors were introduced for H.264/AVC. The statistically adaptive WME and aggregated multi-block-size MME, edge-based intra prediction in MED and IPD, deeply centered high-speed mode decision at IIM while controllable with scale and offset values, and supporting high-speed reference image feed were also presented for H.265/HEVC. The coding quality evaluation results showed that the proposed techniques for real-time video encoders had a sufficient coding efficiency while achieving real-time operation of HDTV and super-high-resolution videos.

Chapter 3
Multi-chip configuration and flexible stream output

3.1 Introduction

The topic of this chapter is a multi-chip configuration of real-time video encoder VLSIs for parallel encoding of super-high-resolution videos or multi-channel videos. For cutting-edge applications which require the highest resolution videos, parallel operation of multiple VLSIs is a very promising solution. This chapter therefore first classifies parallel encoding methods and discusses their pros and cons, after that proposes an inter-chip connection method for super-high-resolution encoding with multiple VLSIs. In addition, an inter-chip flexible stream output technique is proposed, which could accommodate both super-high-resolution and multi-channel encoding and construct MPEG-2 transport streams without external multiplexer devices.

3.2 Classification of parallel encoding methods

Various parallel encoding methods has been continuously studied [54] [55] [16] so far, and Figure 3.1 shows the variation of parallel encoding methods in the viewpoint of how input videos are split and fed into the multiple video encoder VLSIs.

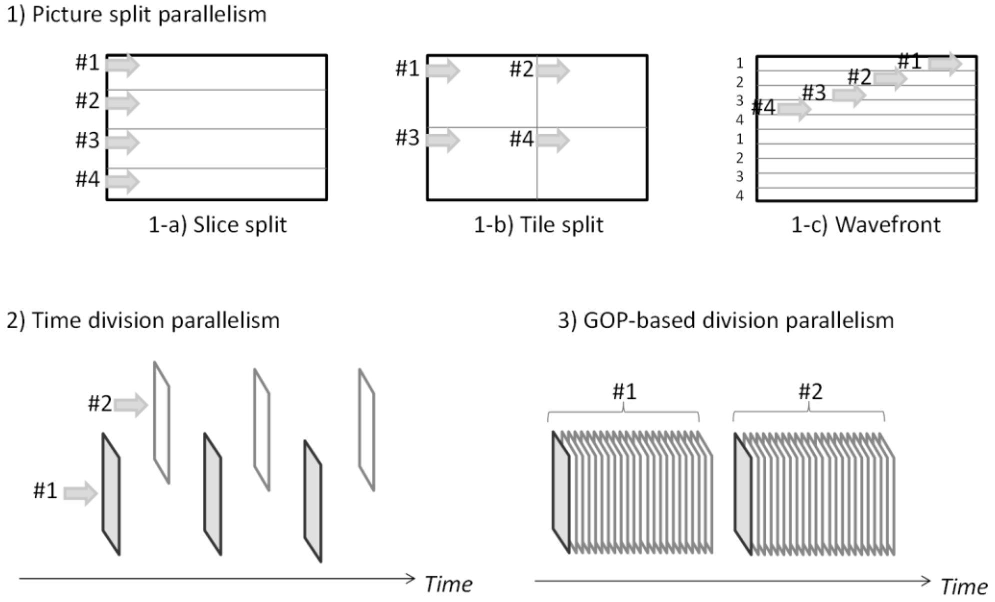


Figure 3.1 Classification of parallel encoding methods.

Table 3.1 Comparison of the parallel encoding methods.

Parallel processing types		**Advantages**	**Disadvantages**
1) Picture split	1-a) Slice split	• Uniform inter-chip data exchange and stream output	• Cause one picture delay for horizontal split
	1-b) Tile split	• Less delay	• Non-uniform inter-chip data exchange and stream output
	1-c) Wavefront	• Least delay	• Tight-coupled CTU-level inter-chip data exchange
2) Time division		• Inter-chip data exchange is required only per picture basis	• Reference structures are limited, hard to increase parallelism
3) GOP-based division		• No inter-chip data exchange is required	• Too much delay (typically over 0.5 x n seconds) n: the number of parallelism

Picture-split parallelism is the most common method as described in Figure 3.1 1) and in which are mainly three types of division. 1-a) Slice split is a horizontal split of each picture into multiple “slices” and encoded individually by multiple encoder VLSIs. The concept of slices was introduced in MPEG-2 and has a long history, therefore this type of division is fundamental in parallel video encoding. 1-b) Tile split [56] is newly introduced in H.265/HEVC standard and it can divide one picture into multiple shapes not limited to horizontally split rectangles, and can be encoded and decoded individually in parallel. 1-c) wavefront [56] is another newly introduced split type in H.265/HEVC, where parallel encoding can be performed per coding tree unit (CTU) row of typically 64 pixels with multiple encoders.

Time division 2) is another way of parallelism, where pictures are assigned to different encoder VLSIs and encoded individually. GOP-based division 3) is an extension of time division, where groups of pictures (GOPs) are assigned to different encoder VLSIs.

Table 3.1 shows the advantages and disadvantages of each split method for parallel video encoding. First, 3) GOP-based division is a technically easiest way of parallelism because when inter-GOP picture reference is not used, no inter-chip data exchange is required and each encoder VLSI can run totally separately. For real-time encoders, however, this approach is unacceptable because of its disadvantage of very long delay which occurs when input pictures are reordered and stored before feeding into parallel encoder VLSIs. (For software-based encoders which are used for offline encoding of video on demand (VoD) contents for distribution, this very long latency is not a problem and therefore this method is often used for parallel encoding.)

Second, 2) time division is the next technically possible solutions, because inter-chip data exchange of reference pictures is required per picture basis and relatively loose inter-chip connection is allowed. However, reference structure is limited due to the encoding timing of each encoder and for the same reason increasing the amount of parallelism is also limited.

In case of 1) picture split, 1-c) wavefront has a definitive advantage of least delay because the input picture buffering required for allocation to multiple encoders is the least amount of approximately “the number of parallel encoder VLSIs × CTU lines”, resulting in parallel processing in lowest latency. However, tight-coupled CTU-level inter-chip data exchange is required for exchanging encoded block information between adjacent VLSIs, which is a burden for inter-chip interface design. Also, special parallel processing mode (adjacent block information is decoupled with each other) must be used for entropy coding, which result in coding efficiency degradation. 1-b) Tile split can achieve less latency, but cross-split pattern requires inter-chip picture exchange between horizontal, vertical and their mixture at the center requires complicated data exchange and it also complicates the decoding process. 1-a) Slice split has a disadvantage of causing one picture delay (33 milliseconds for HDTV and 16 milliseconds for 4K/8K) for buffering and allocation of input pictures, however, this additional delay is acceptable compared to the coding delay (normally 1 second or higher). Uniformity of inter-chip picture exchange between upper and lower VLSIs makes multi-chip control and inter-chip connections easier with common interfaces such as PCI express buses.

In this dissertation, 1-a) slice split is chosen due to the above-mentioned reasons. In the next section, multi-chip parallel encoding method with slice split is introduced with the latest H.265/HEVC encoder VLSIs for super-high-resolution 8K encoding. After that, a distributed multiplexer architecture, which can flexibly generate multi-chip outputs streams for both super-high-resolution and multi-channel applications, is presented.

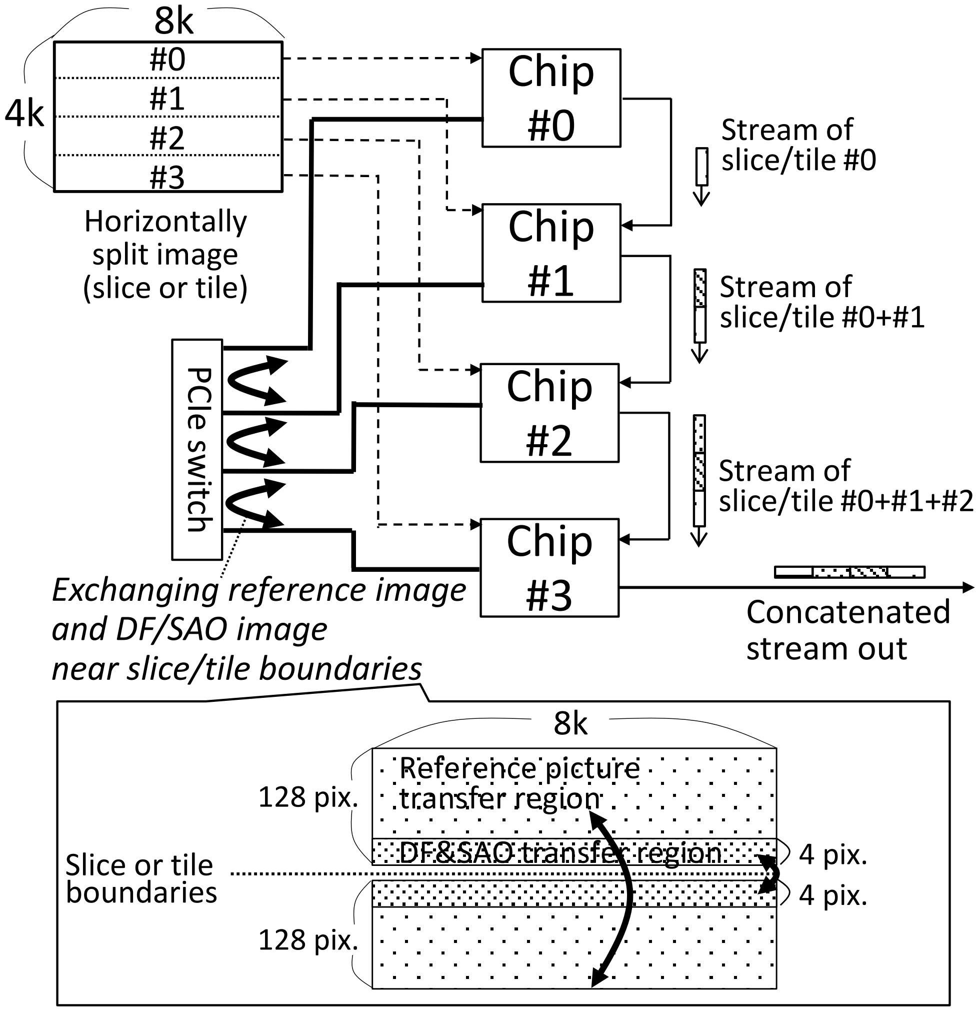

8k
4k
#0
#1
#2
#3
Horizontally split image (slice or tile)
Chip #0
Chip #1
Chip #2
Chip #3
Stream of slice/tile #0
Stream of slice/tile #0+#1
Stream of slice/tile #0+#1+#2
PCIe switch
Exchanging reference image and DF/SAO image near slice/tile boundaries
Concatenated stream out
8k
128 pix.
Reference picture transfer region
DF&SAO transfer region
4 pix.
4 pix.
Slice or tile boundaries
128 pix.

Figure 3.2 8K multichip configuration and interconnect with H.265/HEVC video encoder VLSIs.

3.3 Slice split parallel encoding

Compared to 4K, 8K videos have twice the horizontal and vertical pixels and therefore four 4K real-time encoder VLSIs is equivalent to real-time encoding of one 8K video. 8K H.265/HEVC real-time encoding configuration is thus achieved by parallel operations of four interconnected 4K H.265/HEVC video encoder VLSIs.

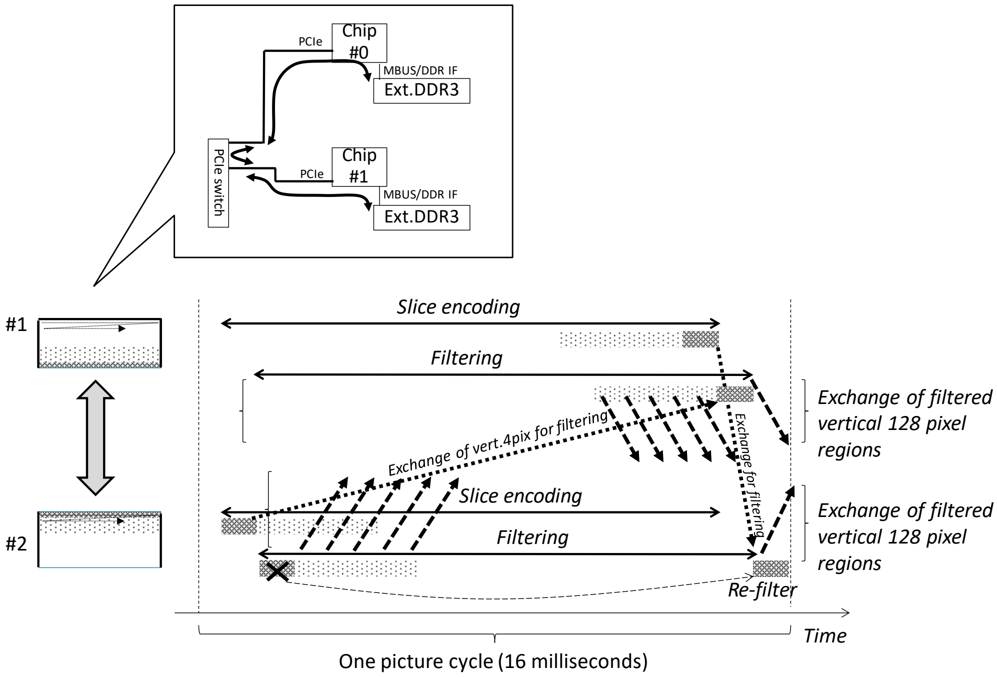

Chip #0
PCIe
MBUS/DDR IF
Ext.DDR3
PCIe switch
Chip #1
PCIe
MBUS/DDR IF
Ext.DDR3
#1
#2
Slice encoding
Filtering
Exchange of filtered vertical 128 pixel regions
Exchange of vert.4pix for filtering
Exchange for filtering
Slice encoding
Filtering
Exchange of filtered vertical 128 pixel regions
Re-filter
Time
One picture cycle (16 milliseconds)

Figure 3.3 Inter-chip data exchange of adjacent chips for parallel encoding.

Horizontally split 8K images are provided to each chip, as illustrated in Figure 3.2. First, the reference picture image cache which is described in Section 2.3.8 can be configured to 8K widths. In order to perform ME across split boundaries, encoded pictures (locally decoded pictures) near the boundaries need to be transferred to the neighboring upward and downward chips as reference pictures of successive pictures. In 8K broadcasting standard [57], motion vectors across these split boundaries are limited within 128 vertical pixels to reduce bandwidth requirements for 8K decoders. This encoded image transfer therefore needs to exchange the rectangular regions of 128 vertical pixels from split boundaries. Other images which need to be transferred is pre-filtered encoding images of vertical 4-pixel rectangle regions, which is required for deblocking filter (DF) and sample adaptive offset (SAO). These filter operations are defined in the H.265/HEVC standard in order to relieve picture degradation artifacts in high compression.

Figure 3.3 shows the inter-chip data exchange in time-wise manner, between adjacent two chips. In each chip, filtering (DF/SAO) of encoded images are processed right after the slice encoding in pipelined scheduling. Vertical 128-pixel regions are transferred after filtering, and transfer of these regions is enough to be finished before the end of one picture cycle, because these regions are used for reference pictures from next encoding picture. Meanwhile, pre-filtered encoding images of vertical 4-pixel rectangle regions must be transferred before filtering is performed in adjacent chips. As illustrated in Figure 3.3, transfer of pre-filtered images from upward to downward is the most critical.

In this way, the amount of data and their criticality varies. Images are scheduled to be transferred via a PCI Express connection, while memory buses (MBUS) perform static (high priority for time critical data) and also dynamic (temporarily high priority in the case of buffer underrun/overrun) quality of service (QoS) management. This QoS management allows each data transfer to be finished in the specified timing to be used, assuring real-time operation of multi-chip encoders.

This inter-chip data exchange functionality for parallel encoding of super-high-resolution videos is included in the H.265/HEVC video encoder VLSI “NARA” which was described in Section 2.4.2 and is utilized for 8K H.265/HEVC real-time encoders.

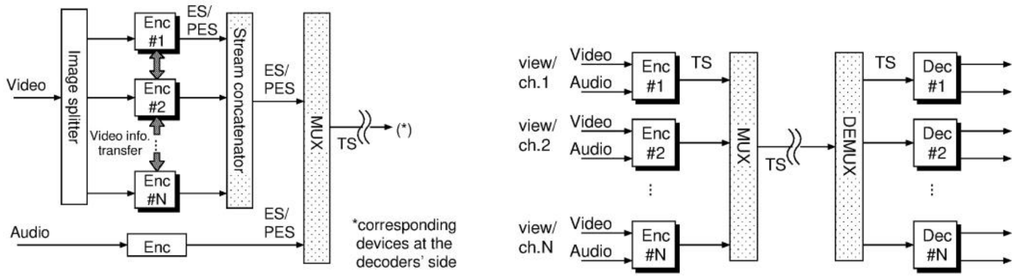
3.4 Distributed and flexible multiplexer architecture

3.4.1 Requirements for inter-chip stream output

Another major inter-chip transfer need is bit stream output, where the encoded bit stream from each chip needs to be transferred and concatenated per slice in the correct order to create a whole stream.

One solution would be a simple concatenation of video streams which is illustrated in the right part of Figure 3.2, which means, encoded video stream (video elementary stream) is transferred per slice from upper chips to lower chips in the correct order via a dedicated stream transfer interface. This solution, however, has a problem at the lowest chip, where stream multiplexing tasks are concentrated on one multiplexer[13] at the lowest chip and maximum output bit rate is limited due to the performance of one multiplexer. This is especially a problem for professional video encoders, because when broadcasters use video

[13] To understand where and how a multiplexer operates, see Figure 1.2, Figure 1.3 and the last paragraph of Section 1.3.



(a) Parallel encoding of large videos

(b) Parallel encoding of multiple channels

Figure 3.4 Two types of multi-chip configuration.

encoders for contribution (transmission of source contents between broadcasting stations), very high bit rate of around several hundred megabits per second is sometimes used for avoiding picture degradation and high bit rate output capability is crucial.

Another factor to be considered is operational flexibility for multiple video encoding applications. Figure 3.4 shows the two types of multi-chip encoding configuration. (a) Parallel encoding of large videos is for super-high-resolution videos, where slice split videos are encoded in parallel and output streams as elementary streams (ESs) or packetized elementary streams (PESs) are gathered and concatenated in the correct order to form a complete ES or PES before the multiplexer (MUX). The other configuration (b) parallel encoding of multiple channels is used for multi-channel or multi-view encoding and MPEG-2 transport stream (TS) output from each encoder is once again multiplexed by external TS multiplexer (TS-MUX) into one multi-channel or multi-view stream.

To solve the issues described above, a distributed TS-MUX architecture which can handle both of the types illustrated in Figure 3.4 with internal multiplexers in video encoder VLSIs are proposed.

3.4.2 Block diagram of a multiplexer

A block diagram of internal MUX is illustrated in Figure 3.5. It is derived from memory-based architecture for MPEG-2 system protocol LSIs [58], with some hardware modification to enhance MPEG-2 TS processing. Dedicated hardware units assist high throughput processing of standard MPEG-2 TS generation, while a RISC CPU dedicates itself to handling protocol extensions and additional requirements. When a codec's encoding and decoding functions operate exclusively, these resources and interfaces of MUX / de-multiplexer (DEMUX) are reconfigured and shared.

To realize the proposed architecture, an external TS input and inter-chip control interfaces are added for multi-chip extension, which are shown in Figure 3.5 with dashed squares. They provide daisy-chained parallel operation of multiple MUXs, which behavior is described in the following sections.

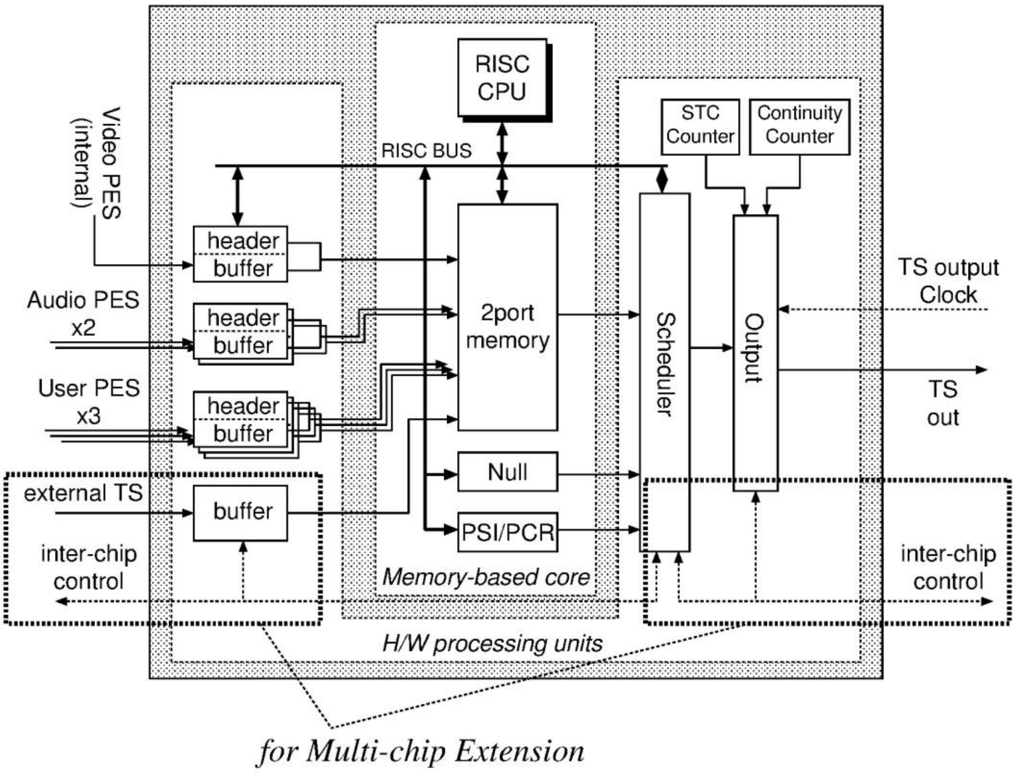
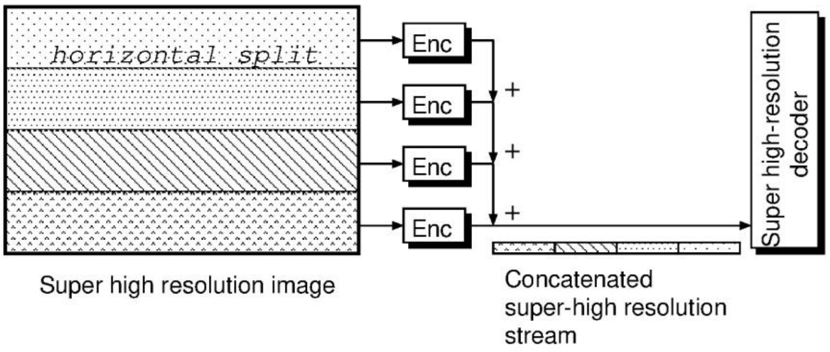


Figure 3.5 A block diagram of the proposed MUX.

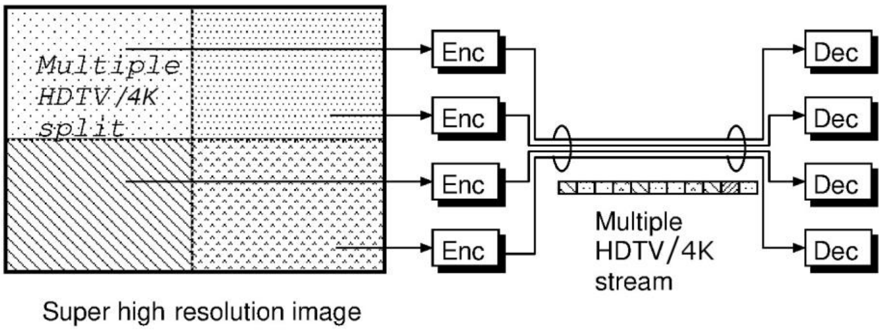
Table 3.2 Operation modes for multi-chip configuration

Application		Configuration
Super-high-resolution video	Horizontally split, sequential output	Concatenation mode
	Multiple HDTV/4K, parallel output	Mixture mode
Multi-program		
Multi-view/-angled		

Two operation modes are arranged to represent each of the configurations in Figure 3.4 for various multichip applications, which are listed in Table 3.2. For encoding of super-high-resolution videos, when images are horizontally split but a complete sequential video stream is required, each chip's output needs to be concatenated per picture, as illustrated in Figure 3.6 (a). In this case, "concatenation mode" is appropriate. In contrast, as depicted in Figure 3.6 (b), when images are cross split into multiple HDTV or 4K videos


horizontal split
Enc
Enc
Enc
Enc
+
+
+
Super high-resolution decoder
Super high resolution image
Concatenated super-high resolution stream

(a) Horizontal split



(b) Multiple HDTV/4K split

Figure 3.6 Two split methods for super-high-resolution video.

and transmitted in parallel to multiple HDTV or 4K decoders, then “mixture mode” can be used to transmit with multiple TS channels. This mode is used when super-high-resolution codec is not available due to technical or procurement problem and multiple HDTV/4K codecs are alternately installed. For multi-program or multi-view/-angled applications, streams are transmitted with multiple TS channels and “mixture mode” operates as a substitute for external TS-MUX.

3.4.3 Concatenation mode

The multi-chip configuration for concatenation mode is illustrated in Figure 3.7. TS I/O interfaces are connected between each neighboring two chips, and inter-chip controls are connected ring-wise to transfer a “token”. Each slave chip encodes a split picture in its charge and its MUX transforms the video PES into TS packets. The MUX in a master chip

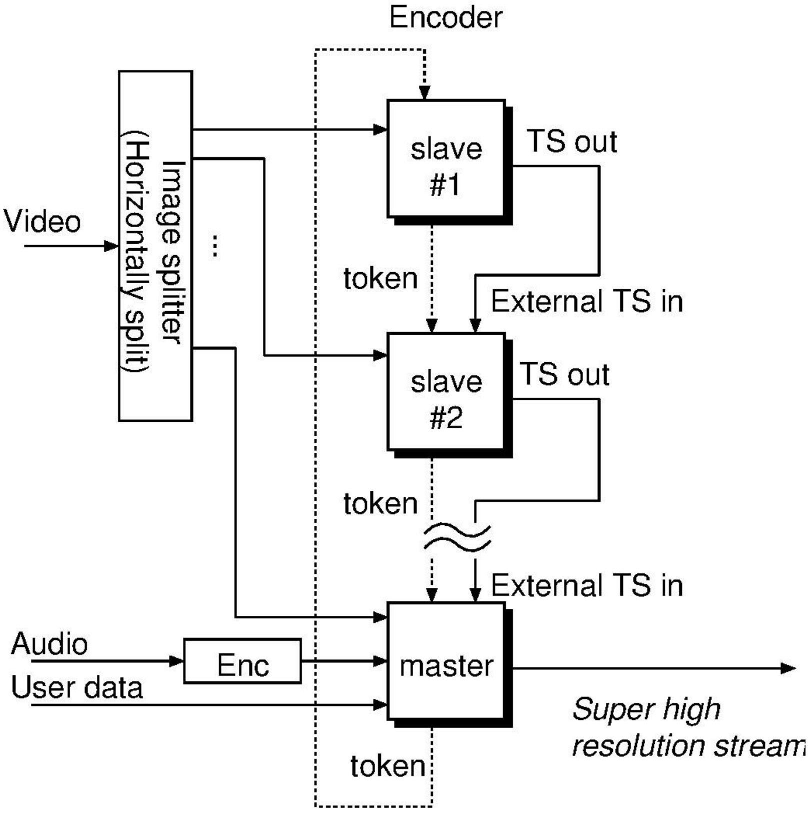
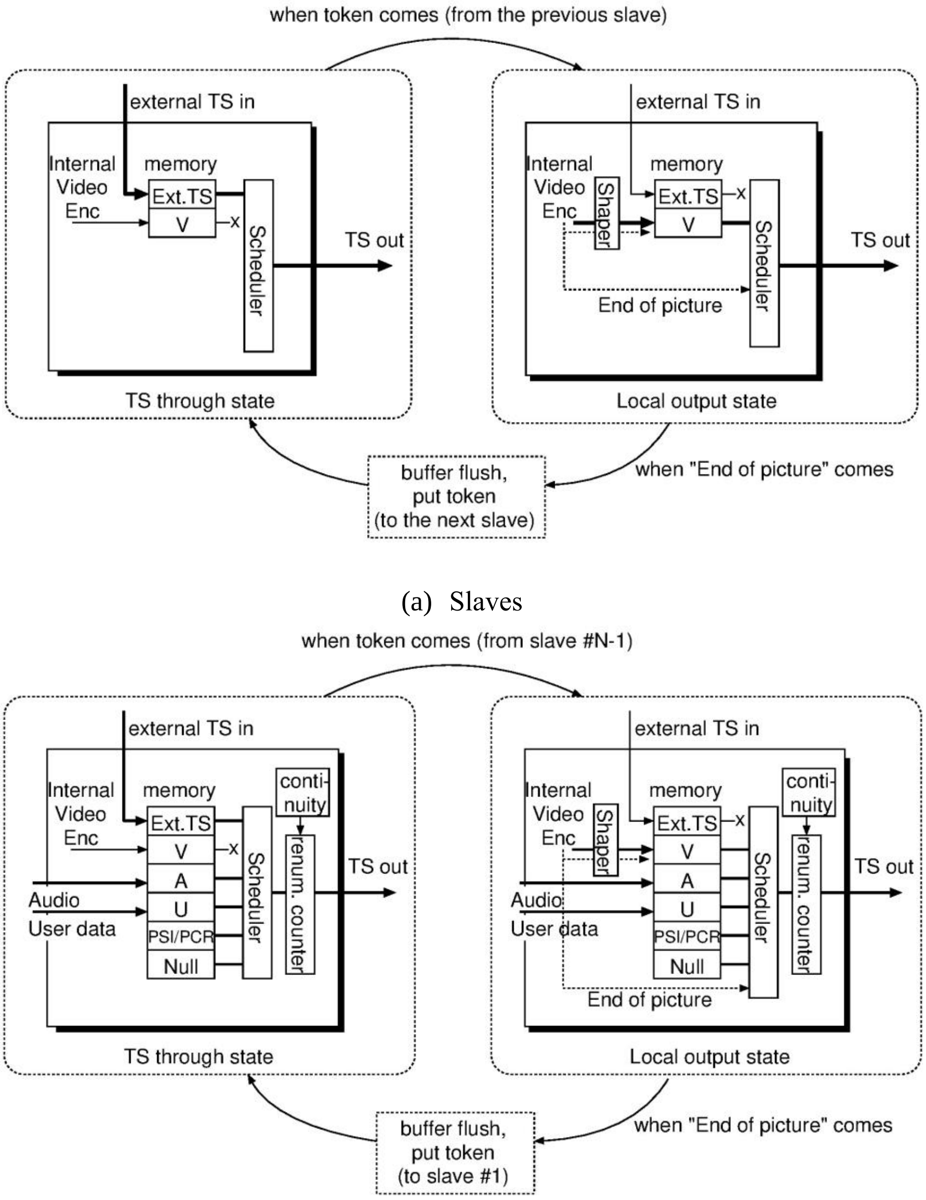


Figure 3.7 Concatenation mode configuration.

additionally handles audio / user data PES inputs, and is also responsible for complete multi-chip stream output.

State transition diagram of each slave MUX is shown in Figure 3.8 (a). Each time, only one MUX which has a token is allowed to output self-generated video TS packets. The other slaves operate in “TS through state”, simply relaying TS packets from the previous MUX to the next. When a certain slave gets a token from the previous chip, its MUX transits its state into “local output state”. Here the encoded video PES is transferred to the MUX in constant bit rate from a shaper, until the end of one split picture. The MUX generates and sends TS packets, and when an “end of picture” signal comes from the internal encoder, the MUX sends the remaining video TS to flush the buffer, puts the token to the next chip, and returns to the TS through state. Padding the last video TS to flush leads to bit loss, the average of which can be calculated as:


when token comes (from the previous slave)
external TS in
Internal Video Enc
memory
Ext.TS
V
x
Scheduler
TS out
TS through state
Shaper
End of picture
Local output state
buffer flush, put token (to the next slave)
when "End of picture" comes
when token comes (from slave #N-1)
A
U
PSI/PCR
Null
Audio
User data
conti-nuity
renum. counter
buffer flush, put token (to slave #1)

(a) Slaves

(b) Master

Figure 3.8 MUX operations in concatenation mode.

$$\frac{\mathit{payload\_size}}{2} \times N \times (\mathit{pictures\_per\_second})$$

where N is the number of chips. For 4-chip 60 frame per second systems, the average loss will be 17.6 kbps, which is almost negligible compared with the high video bit rate.

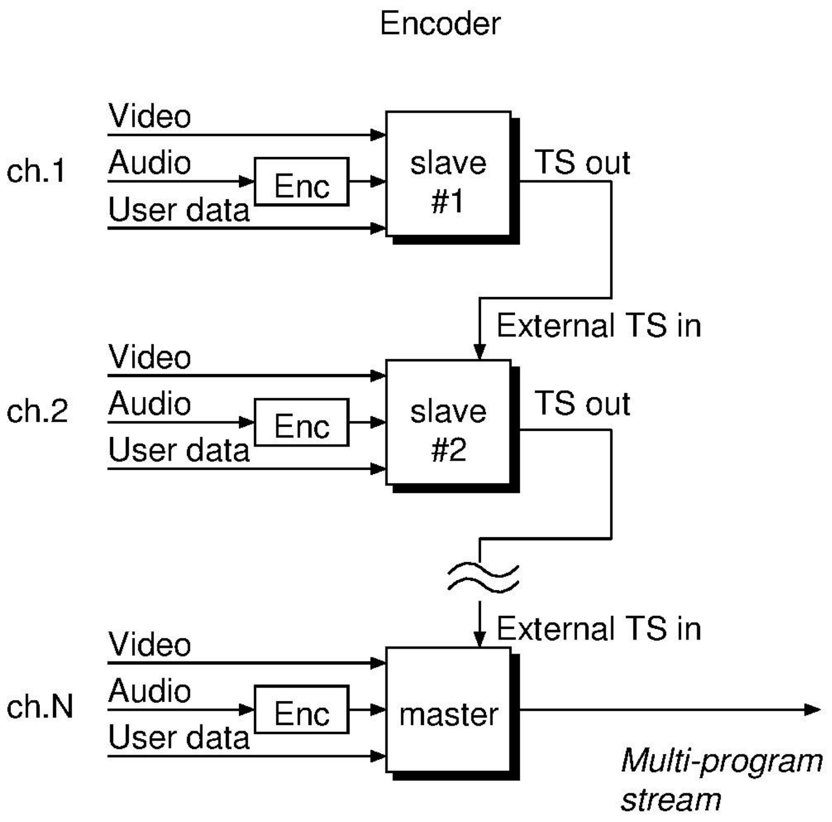


Figure 3.9 Mixture mode configuration.

State transition diagram of the master MUX is illustrated in Figure 3.8 (b). When in TS through state, the scheduler interprets external TS as video TS and does the output scheduling. When the token comes, the scheduler switches the source and chooses the self-generated video TS, until the end of the picture. Other TS types such as audio, user data, program specific information (PSI) / program clock reference (PCR) and null packets are scheduled as per normal. Since video TS packets generated in parallel have inconsistent serial numbers, a continuity counter field in each video TS is renumbered at the final stage.

As mentioned above, split video streams are concatenated at the TS phase, which achieves super-high-resolution stream output without external devices. This method can also apply to other video coding standards, such as H.264/AVC and H.265/HEVC, as long as each split image is encoded as a" slice" and concatenated afterwards. When decoding is also performed in parallel with multiple video decoders, each decoder simply extracts a video

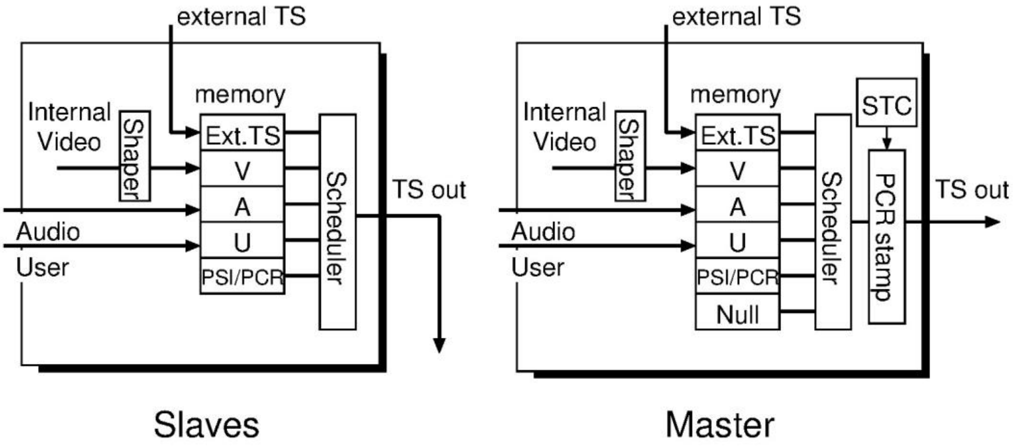


Figure 3.10 MUX operations in mixture mode.

elementary stream (ES) of the picture region in its charge, and no extra mechanism is needed for the DEMUX.

3.4.4 Mixture mode

For mixture mode, the multi-chip configuration and corresponding MUX operation are illustrated in Figure 3.9 and Figure 3.10 respectively. Here a multi-program application is shown, for instance, with each chip handling video, audio and user data of each program. Each slave produces video / audio / user data / PSI / PCR TS packets with individual program IDs (PIDs), but no null packets. The scheduler transmits these TS packets and also external TS packets from the previous chip whenever they arrive, to the next chip. The master is responsible for adding null packets. To avoid timing jitter that occurred during the transfer, shared PCR values are stamped at the final stage.

When encoders operate with super-high-resolution video or multi-view/-angled vision such as MPEG-2 Multi-View Profile, H.264/AVC Multi-View High Profile and H.264/AVC Stereo High Profile, each slave handles only video and external TS. Shapers which control the video PES output rate also need to be controlled according to the multichip rate control; however, other operations remain to be the same as mentioned above. On the decoding side, DEMUX of each decoder simply extracts the video TS in its charge and transfers them to the internal video decoder, thus no extra function is required.

3.5 Related work

Parallel operation of multiple VLSIs has always been a major solution for encoding and decoding large scale data. Parallel encoding methods for video encoding have been studied [54] [16] so far, transferring video information among multiple encoders. However, they focus on the inter-chip exchange of reference pictures for motion estimation and MC and not on the inter-chip DF/SAO filtering which is mandatory for newer standards of H.264/AVC and H.265/HEVC. Furthermore, there have been no other researches about built-in multiplexers of video encoder VLSIs to work in parallel and perform distributed operation of both concatenate mode and mixture mode. The distributed TS-MUX architecture therefore is unique.

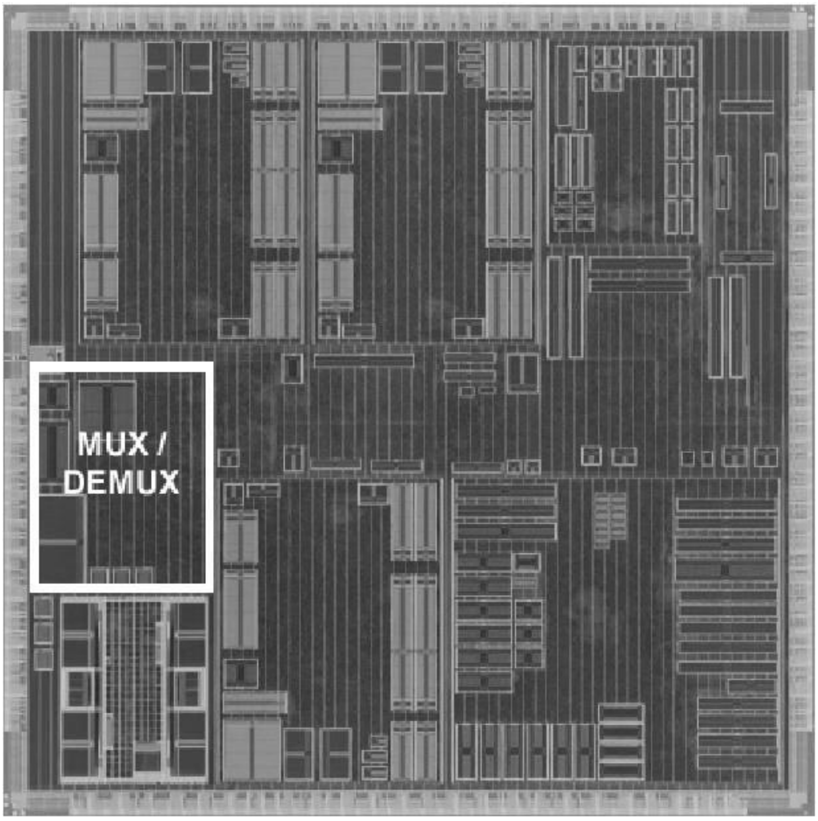
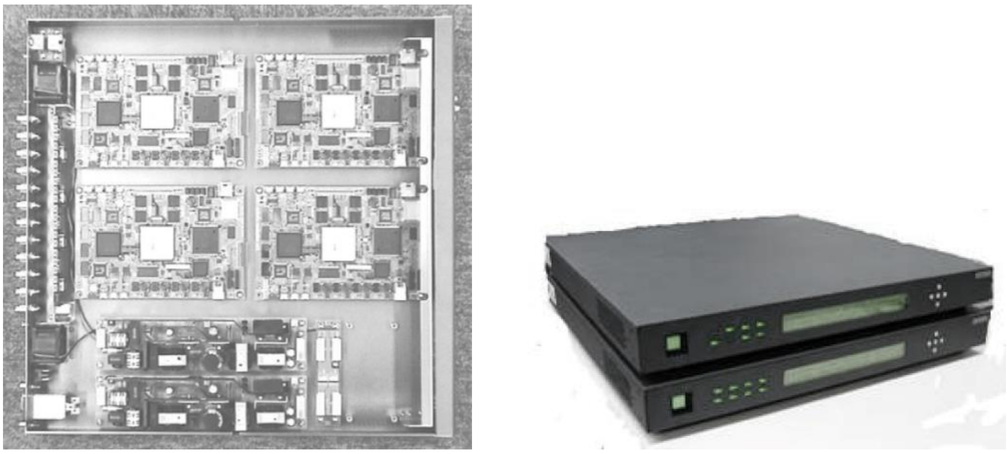


Figure 3.11 A micrograph of the fabricated VLSI “VASA”.

3.6 Implementation and evaluation

The MUX architecture was included in the fabricated single-chip MPEG-2 422P@HL CODEC LSI [23] using 0.13-μm 8-level metal CMOS technology. A micrograph of the LSI is shown in Figure 3.11. In the outlined area, 6% of the floor dimensions are allocated for MUX / DEMUX. The overhead implementing the new architecture is around 2% compared with the conventional MUX / DEMUX, which is almost negligible with a large downsizing advantage of overall systems. The maximum output rate is 270-Mbit/s through an 8-bit parallel TS output interface, and dispensing with low-speed external devices also contributes to adaptability to higher throughput.

We have developed an experimental super-high-resolution encoder system [59] with the proposed architecture and successfully encoded 4Kx2K camera images in real time in a



(a) Inner photograph (b) Outside chassis

Figure 3.12 Photograph of a super-high-resolution encoder.

minimum 60-Mbps transport stream. Four encoder boards are installed inside as shown in Figure 3.12 (a), and each board processes a split image with inter-board bit rate allocation control. Encoder chip interfaces are interconnected with serial cables and MUX is performed in mixture mode as illustrated in Figure 3.6 (b), since a super-high-resolution decoder system is built up with multiple HDTV decoder LSIs [23]. The overall super-high-resolution codec is installed in a 1-U (460 × 440 × 44-mm) chassis as illustrated in Figure 3.12 (b), to which downsizing the proposed architecture significantly contributes.

The output of the super-high-resolution encoder system was analyzed to assess the multiplexing quality of the proposed TS-MUX architecture. Figure 3.13 shows the bit rate of individual video and total TS, observed at the final stream output from the master. Individual video PES output rate is set to 30-Mbps which is multiplexed into the final TS of 145-Mbps. Occasional sags in the bit rate are due to the bit rate control and occur when the actual amount of video PES is less than expected, so that provision of video PES to the MUX is suspended and null TS packets are filled instead.

To check the uniformity of TS output, an evaluation is performed with 6,000 TS packets described by the arrowed line in Figure 3.13. Figure 3.14 shows the consecutiveness and interval of video TS packets that are produced at each encoder chip. Figure 3.14 (a) proves that no TS packets from the same encoder chip are located bumper-to-bumper in the final

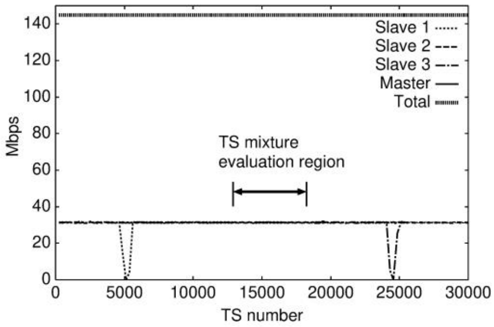
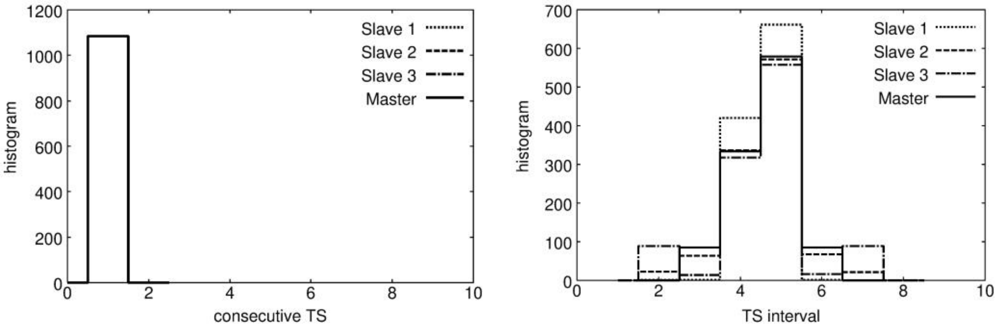


Figure 3.13 Output TS bit rate.



(a) Consecutiveness of TS from each chip

(b) Interval of TS from each chip

Figure 3.14 TS mixture evaluation results.

output. Figure 3.14 (b) illustrates that intervals of TS packets from the same encoder chip are uniformly distributed, proving that there are no unevenly distributed segments and all TS packets are uniformly mixed through the daisy-chained paths.

3.7 Chapter summary

In this chapter, multi-chip configuration of real-time video encoder VLSIs for parallel encoding of super-high-resolution videos or multi-channel videos was presented. An inter-chip connection method for super-high-resolution encoding with multiple VLSIs was described, which has been installed on the latest 4K real-time H.265/HEVC video encoder VLSI "NARA" and utilized for 8K real-time encoders.

In addition, an inter-chip flexible stream output technique was also proposed, which could accommodate both super-high-resolution and multi-channel encoding and construct MPEG-2 transport streams without external multiplexer devices. This has been installed in the HDTV MPEG-2 encoder VLSI "VASA" and multiplexing quality of the output stream was assessed, where TS packets from multiple VLSIs were successfully multiplexed into one TS outputs with daisy-chained paths.

Chapter 4
Power reduction of motion estimation engines

4.1 Introduction

This chapter focuses on power reduction of video encoder VLSI's motion estimation engines in order to reduce the circuit scale and power consumption for mobile application VLSIs and further to develop video encoder VLSIs conforming to next generation standards in the future.

For example, H.265/HEVC video encoder VLSI described in Section 2.4.2, based on the techniques and the architecture proposed in Chapter 2, has made real-time transmission, broadcast and distribution of ultra-high definition programs possible. Development of this VLSI led to Japanese 8K commercial satellite broadcasts which started in December 2018.

However, increased circuit scale and power consumption makes the spread of H.265/HEVC ultra-high definition mobile applications (e.g. ultra-high definition camcorders and transmitters, smartphones with 5G) more difficult and reduced circuit scale and power are vital for further popularization of ultra-high definition videos. Among the H.265/HEVC encoding processes, motion estimation occupies a large portion as discussed in Section 1.4, because larger motion search range and finer multiple-block-size ME are essential for coding efficiency. Reduction in ME processing therefore plays a key role in overall scale and power reduction of H.265/HEVC ultra-high definition encoders.

In this chapter, a bit-reduced ME engine that focuses on the local flatness of encoding images are proposed. When reducing ME operation bit widths, usually the upper bits of the luminance values are utilized and the lower bits are omitted. In the proposed ME engine, however, when a picture has a flat region where the upper bits of the luminance values are uniform, the uniform upper bits are ignored and the lower bits are utilized for sum of absolute difference (SAD) calculation. Adaptively selecting these bit extraction positions enables ME to reduce circuit scale and power consumption without lowering the block matching precision even in a flat luminance region. The proposed techniques are installed in wide ME (WME) and multi-block-size ME (MME) blocks on the H.265/HEVC encoder VLSI design in Section 2.4.2 and circuit scales are reduced by 18-34% with SAD calculation bits shortened by half (i.e. 4 bits). Power simulations show that power consumptions are also reduced by 18-39%. Coding efficiency loss is suppressed by up to 62% in 4K and to 55% in 8K with adaptively bit-shifted ME techniques, indicating that the proposed ME engine effectively maintains ME precision with shorter bit calculation with ultra-high definition videos including High Dynamic Range (HDR).

4.2 ME energy consumption in H.265/HEVC encoder VLSI

In the prediction core which is illustrated in Figure 2.10, four ME operations are performed: wide ME (WME) for wide-area motion estimation of 32×32 pixel blocks, multi-block-size ME (MME) for neighboring search from 8×8 to 64×64 pixel blocks, and integer-pel ME (IME) and fractional-pel ME (FME) for obtaining finer quarter-pel motion vectors (MVs). These MEs account for more than 60% of the energy consumption (average value of power) in the prediction core (excluding the reference picture image cache), as shown in Table 4.1. The majority of ME operations are block matching operations based on

Table 4.1 Energy consumption percentages for ME blocks.

ME block	Processing content	Power consumption ratio in prediction core
WME	Pre-ME with 1/8 or 1/4 (adaptively) reduced picture	4.4%
MME	Multi-block-size ME with 1/2 reduced picture	32.5%
IME	Integer-pel neighboring ME	32.5%
FME	Fractional-pel neighboring ME	32.5%
	Total	62.7%

sum of absolute difference (SAD) calculations. Therefore, reducing the SAD calculation process helps to reduce power consumption in video encoders in general.

However, 4K / 8K ultra-high definition videos with larger pixel sizes inevitably require a wider search range to enable MEs to follow picture motions, and reducing the number of SAD operations is impractical. This paper focuses on bit width reduction when performing SAD operations and proposes adaptive bit-reduced ME techniques that prevent degraded block matching precision with a limited bit width.

4.3 Related work

Methods for reducing the ME computation in hardware-based real-time video encoders have been intensively studied and are mainly classified into two categories. For the first category, which avoids full search and approaches the minimum cost point step by step with limited search points, several techniques have been proposed [60] [61] [62] to employ methods that are familiar with software encoders [63] [64] [65]. However, degraded efficiency in processing elements and memory accesses is inevitable due to the conditional branch control required for step-by-step searching. For the second category, which reduces operational precision of the block matching calculation, techniques have been developed that use only the upper bits of the luminance value in block matching operations [33] [66], or that use one or two bits per pixel image after the edge extraction filter [67] [68]. These techniques, however, inherently ignore gradual changes in pixel values and may cause coding efficiency degradation. In particular, degraded precision in block matching can be visually conspicuous due to expanded luminance bit widths (10, 12 and 16 bits per pixel) and compressed luminance values that occur with the use of HDR technology, which has become popular in ultra-high definition videos.

4.4 Adaptive bit-reduced ME

4.4.1 Adaptive bit-reduced ME concept

The proposed ME techniques focus on picture regions where the luminance values change

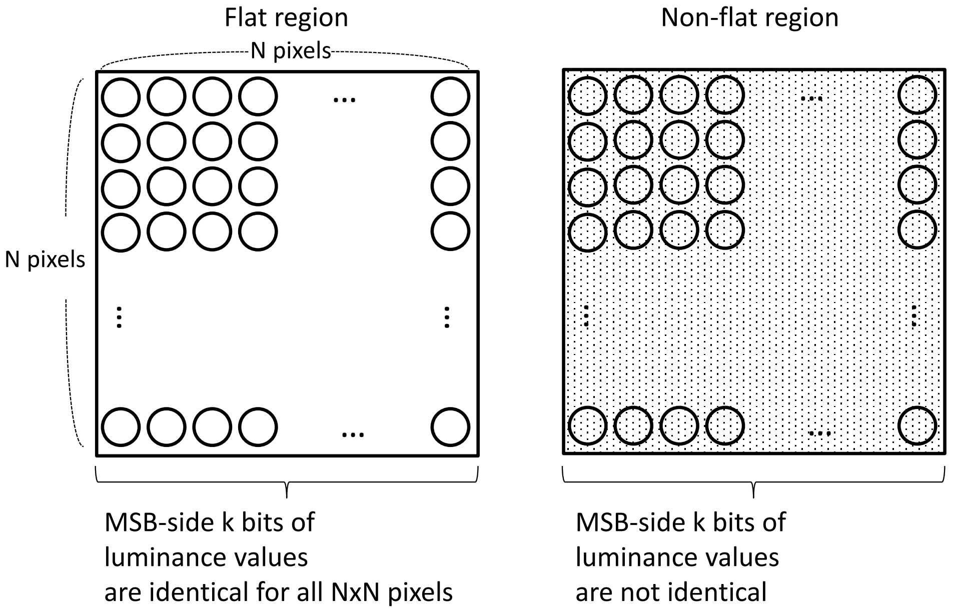

Flat region
Non-flat region
N pixels
N pixels
MSB-side k bits of luminance values are identical for all NxN pixels
MSB-side k bits of luminance values are not identical

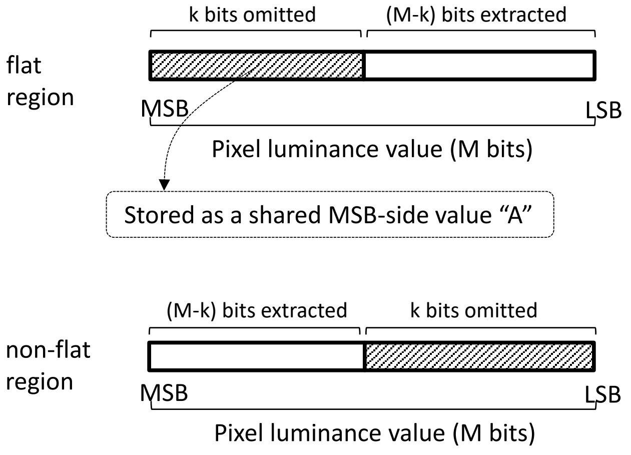
Figure 4.1 Criteria for flat and non-flat regions.

gently, in other words, flat regions. In such regions, the SAD values tend to be small and may be truncated with short bit width. To avoid this and to maintain calculation accuracy, luminance values are extracted with lower bit positions when performing SAD calculations. By adaptively changing the bit extraction positions in accordance with the flatness of luminance values, the proposed ME techniques aim to maintain coding efficiency with shorter calculation bit widths.

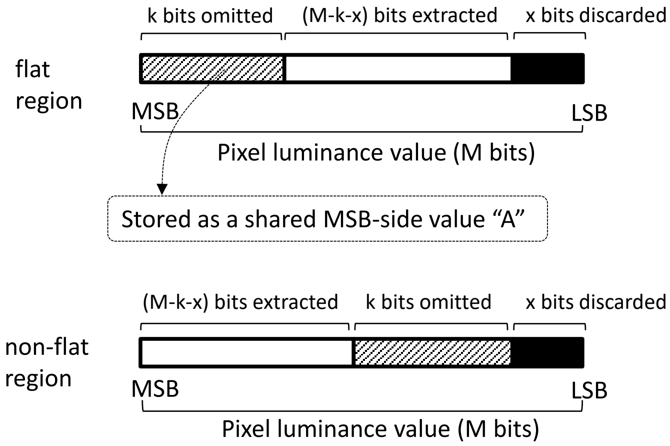
The proposed ME techniques comprise the following procedures: 1) detection of flat regions 2) adaptive picture loading from picture memories to ME engines, and 3) adaptive SAD and cost calculation in ME engines. In the following, the definition of flat/non-flat regions and their bit extraction patterns are first illustrated, after which the adaptive operations are described in detail.

4.4.2 Definition of flat regions and bit extraction patterns

Each N×N pixel region in an input image is defined as "flat" when the upper k bits of all the luminance values are the same, as shown in Figure 4.1. If not, the region is regarded as "non-flat". In H.265/HEVC encoding, the value of N is set to 64 in accordance with the maximum size of the H/265/HEVC prediction unit (PU), to prevent a mixture of flat and



(a) Adaptive extraction from M bits



(b) Adaptive extraction from MSB-side (M-x) bits

Figure 4.2 Bit extraction of luminance values

non-flat regions in the same PU.

Consider that the luminance information is provided by M bits per pixel, k bits are reduced and the remaining (M-k) bits are extracted for bit-shortened SAD calculation. Bit extraction patterns in the proposed techniques are illustrated in Figure 4.2 (a). In the flat region where the upper k bits of all the luminance values are the same, the LSB-side (M-k) bits of luminance values are extracted for each pixel. The most significant bit (MSB)-side k bits in the flat region are identical and therefore omitted from the SAD computation but are stored

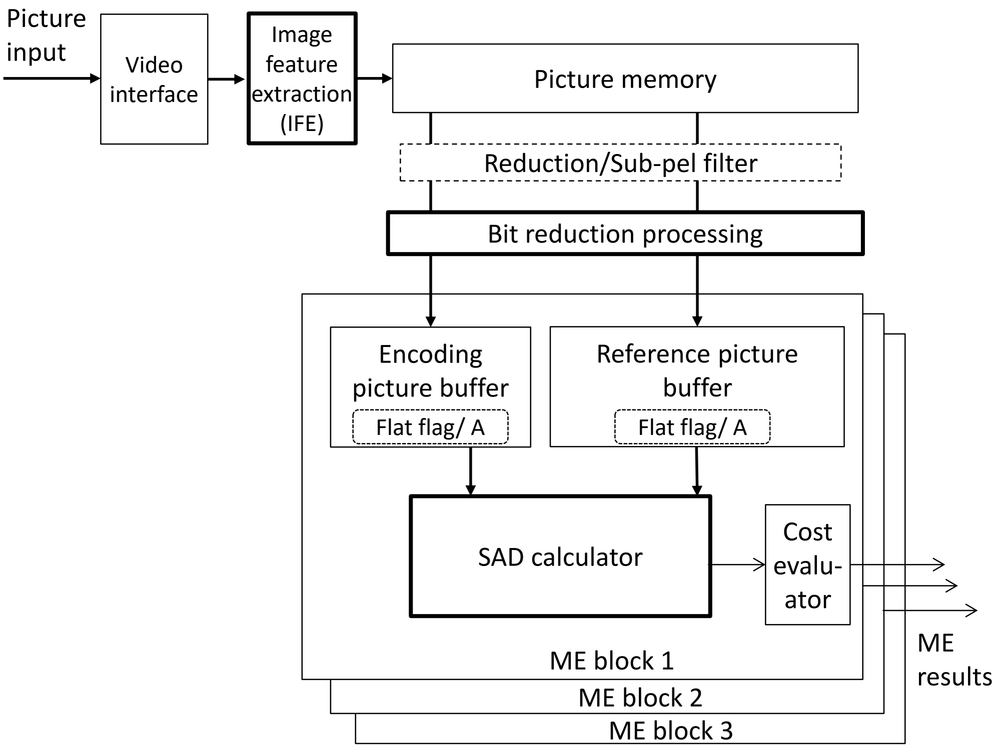


Figure 4.3 Diagram of the proposed ME blocks.

as one shared value “A” for each flat region. On the other hand, in the non-flat region the MSB-side (M-k) bits of each pixel are extracted for the SAD computation and the least significant bit (LSB)-side k bits are omitted.

This method can also be configured so as to permanently discard the lowest x bits as shown in Figure 4.2 (b) and to divide the remaining (M-x) bits to omit k bits and to extract (M-k-x) bits.

4.4.3 Adaptive bit-reduced ME diagram

A diagram of the proposed bit-reduced ME blocks is depicted in Figure 4.3. Input images, after the format conversion and de-noising filtering in the video interface, go through the image feature extraction (IFE) block. Compared to the IFE in Figure 2.10, this IFE has an additional flat detection function to judge the flatness of each region by comparing the upper k bits of pixel luminance values. Input images are then stored in the picture memory with M bits per pixel, with flat/non-flat information produced in the IFE.

When PUs to be encoded and reference pictures of the motion search range are read from the picture memory to the internal picture buffer, bit reduction operation to (M-k) or (M-k-x) bits is applied. Bit extraction positions are adaptively switched for each N×N region by flat/non-flat information. Flat/non-flat flags, with the shared MSB-side k-bit values "A" for flat regions, are also stored in the internal buffer.

The SAD calculator and the cost evaluator do the actual ME operation, performing the SAD calculation and finding motion vectors with minimum coding costs in the processing flow described in the following subsection.

4.4.4 Adaptive bit-reduced ME processing flow

The adaptive bit-reduced ME processing flow is shown in Figure 4.4. For each PU to be encoded, the whole ME process branches in S2, depending on whether the PU belongs to flat or non-flat regions. In the non-flat case, the encoding PU load S3 and the reference picture load S4 to the SAD calculator both comprise MSB-side (M-k-x) bits of each pixel. In the flat case, the picture load S5 and S6 comprise near-LSB (M-k-x) bits. However, in S6 the reference picture load is only for flat regions that have the same shared MSB-side value "A" as the PU to be encoded. Flat regions with different "A" or non-flat regions are ignored.

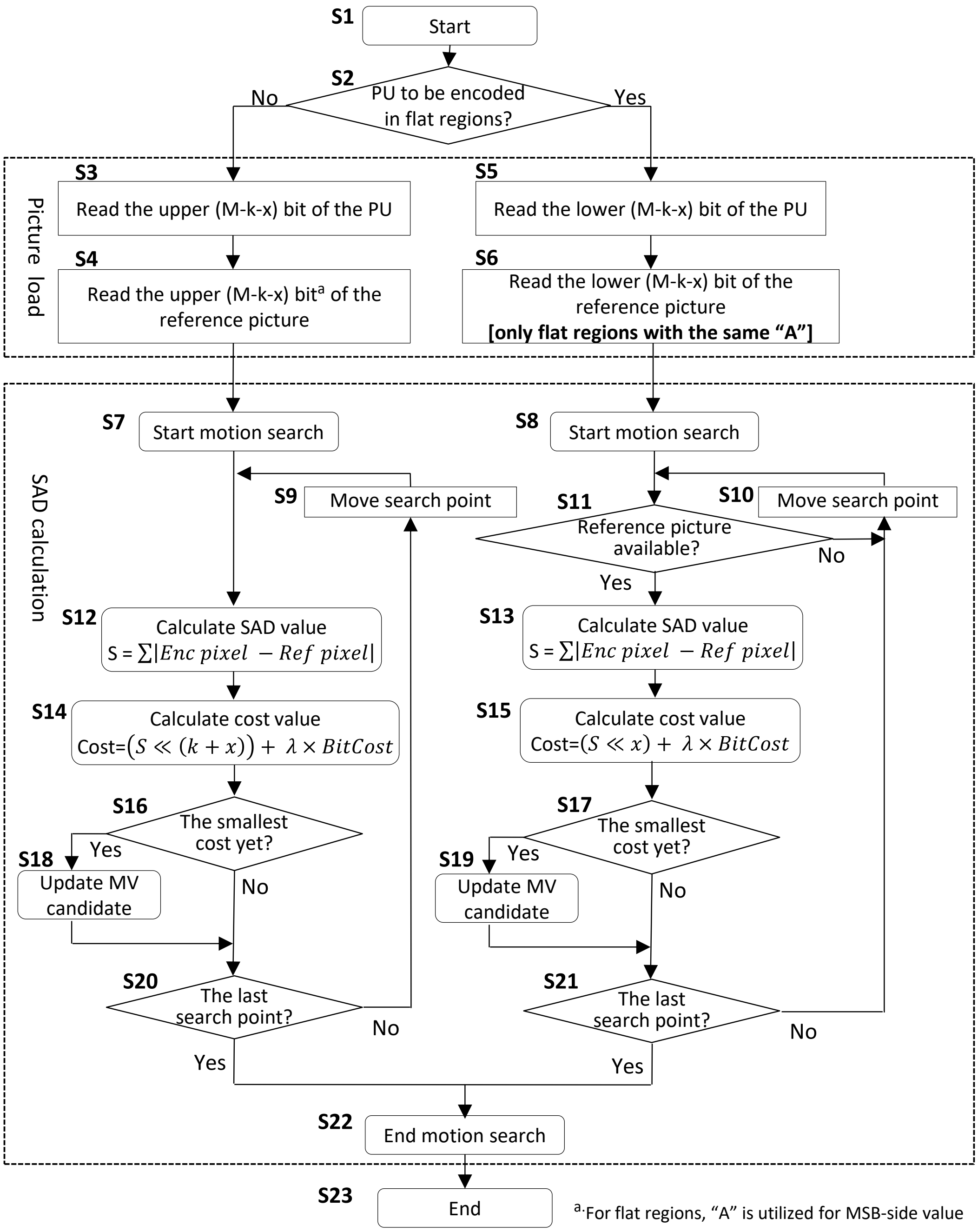
After the picture load operation is completed, a SAD calculation loop is performed throughout the motion search region. It should be noted that in the flat region loop, if there are motion search points where no reference picture is available due to different "A" or non-flat regions, such search points are skipped in S11.

Motion estimation for the best motion vector (MV) is typically done by finding the minimum cost of

$$\mathrm{Cost} = \mathrm{SAD} + \lambda \times \mathrm{BitCost}$$

where λ is a lambda value for rate-distortion optimization [36] and BitCost is a bit amount consumed for MV description. In S14 and S15, SAD values are left-shifted for alignment, and the MV with the minimum cost is stored to be the final ME results.

With the ME operations described above, internal picture buffers and ME operations are reduced to (M-k) or (M-k-x) bits per pixel, except for the cost calculation and comparison in S14-S17. The proposed ME techniques have the advantage of maintaining calculation accuracy in flat regions by using simple bit-shortened SAD engines without introducing complex floating-point operations.


S1
Start
S2
PU to be encoded in flat regions?
No
Yes
Picture load
S3
Read the upper (M-k-x) bit of the PU
S4
Read the upper (M-k-x) bit^a of the reference picture
S5
Read the lower (M-k-x) bit of the PU
S6
Read the lower (M-k-x) bit of the reference picture
[only flat regions with the same "A"]
SAD calculation
S7
Start motion search
S8
Start motion search
S9
Move search point
S10
Move search point
S11
Reference picture available?
No
Yes
S12
Calculate SAD value
$S = \sum|Enc\ pixel - Ref\ pixel|$
S13
Calculate SAD value
$S = \sum|Enc\ pixel - Ref\ pixel|$
S14
Calculate cost value
$Cost = (S \ll (k + x)) + \lambda \times BitCost$
S15
Calculate cost value
$Cost = (S \ll x) + \lambda \times BitCost$
S16
The smallest cost yet?
Yes
No
S17
The smallest cost yet?
Yes
No
S18
Update MV candidate
S19
Update MV candidate
S20
The last search point?
No
Yes
S21
The last search point?
No
Yes
S22
End motion search
S23
End

[a] For flat regions, "A" is utilized for MSB-side value

Figure 4.4 Processing flow of the proposed ME engines.

4.4.5 Selection of bit reduction parameters (k, x)

The amount of reduction in circuit scale and power depends on the parameter (k+x) and must be defined prior to ME engine design, since SAD calculation bit widths in ME engines are specified by the extracted bit width (M-k-x). Larger (k+x) yields more impact on reduction but less calculation precision, and must be $(\mathrm{k}+\mathrm{x}) \leq \lfloor M/2 \rfloor$ to avoid bit positions that cannot be evaluated. In the H.265/HEVC encoder LSI design in Section 2.4.2, WME and MME have M equal to 8, so the maximum (k+x) should be 4.

Once ME engines are designed with the fixed (k+x), k and x can be selected in accordance with the characteristics of encoding videos. Larger k (and smaller x) yields finer SAD calculation precision in flat regions, however, and so it causes fewer flat regions. As shown in step S10-S11 in Figure 4.4, when motion estimation of flat regions is performed, non-flat regions in reference pictures are omitted from the search area. Less flat regions cause smaller motion search areas and the result is limited coding efficiency. To prevent this, the largest k that maintains a certain flat region percentage should therefore be selected. This will be discussed in the evaluation section 4.5.2 and 4.5.3.

4.5 Evaluation of adaptive bit-reduced ME

4.5.1 Circuit scale and power reduction

To clarify how the proposed technique affects the ME blocks in terms of power reduction, a design change was made for two of the ME blocks (WME and MME) in Table 4.1. Since the SAD operation for these blocks was originally designed with 8 bits, the SystemC design was modified so as to reduce it up to by half, i.e. to 4 bits. The modification was synthesized and gate simulation of ME operation with image inputs was performed using the tools listed in Table 4.2. Estimated power consumption values were then obtained.

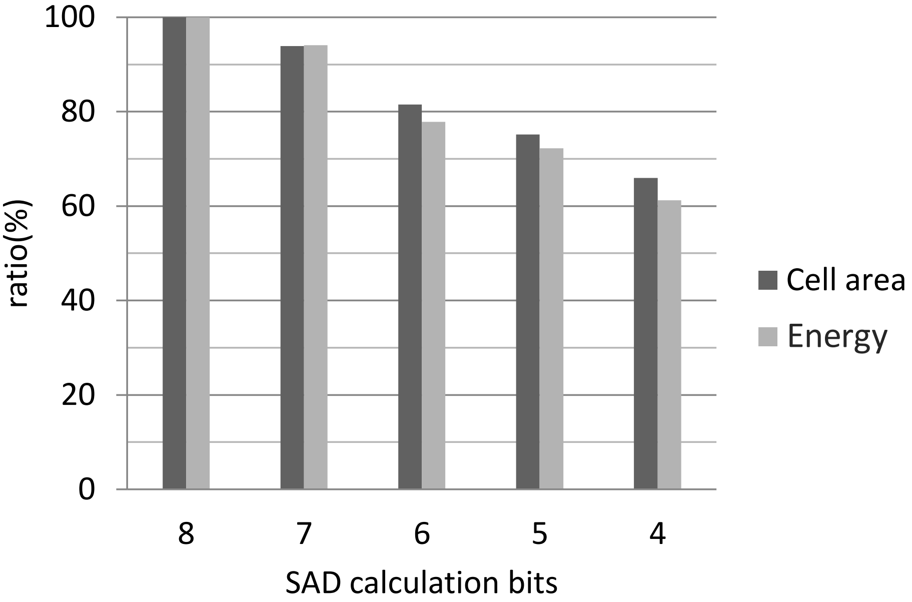


Figure 4.5 Comparison of cell area and energy consumption in WME.

Figure 4.5 shows the transition in cell area and energy consumption (average of power consumption) of WME from the original 8 bits to 4 bits. Circuit scale and energy

Table 4.2 Synthesis and power simulation tools

HLS compiler	Cadence Cynthesizer
Logic synthesis compiler	Synopsys Design Compiler
Power simulator	Synopsys PrimeTime PX
Cell library	TSMC 28HPM P:TT/V:0.9V/T:25°C

Table 4.3 Area and power simulation results (4 bits).

	Cell area	**Energy consumption**
WME	65.9%	61.2%
MME	81.9%	82.5%

consumption proportionally decrease as SAD calculation bits shorten and eventually 39% energy reduction is achieved with 4 bits. Changes to 4 bits in MME were also made and the overall reduction effects in 4 bits are shown in Table 4.3. MME's reduction effect of 18% falls lower than WME because it has an additional multi-block-size (from 8×8 to 32×32 pixels) cost evaluation function that was described in Section 2.3.4 and stays outside of the bit reduction scheme.

4.5.2 Coding efficiency pre-assessment with HDTV video

We compared the adaptive 4-bit ME blocks (WME and MME) mentioned above with fixed MSB-side 4-bit ME blocks in order to pre-assess the characteristics of suppressing coding efficiency degradation. For this, five HDTV video sequences are selected: two for dark night scenes (Twilight, Nebuta) and three with the HDR-compressed luminance values (LucoreHDR, Hawaii1, Hawaii2).

Flat region percentages with different k values, which should be assessed in IFE before encoding, are shown in Table 4.4. The video sequences were then encoded with H.265/HEVC 4:2:0 10bit (Main 10 Profile) and calculated the BD-rate [52], which denotes the extra coding bits required for the same PSNR, compared to the original 8-bit WME and MME blocks. The results are shown in Table 4.5. SAD calculation with fixed MSB-side 4 bits results in a BD-rate increase because of insufficient precision. The increase, however, is suppressed with the adaptive bit-reduced ME techniques as a result of the increased precision in block matching in the flat regions. The underlined figure in each video sequence achieves the best BD-rate increase suppression. When these best BD-rate results are compared with Table 4.4, it is observed that the largest k that maintains a certain flat region percentage achieves the best BD-rate results as discussed in Section 4.4.5, and flat region percentage threshold of 40% is derived experimentally for HDTV video sequences.

There may be better characteristics and parameters to represent thresholds for best BD-rate results (such as variances of flat region percentage and local consecutiveness of flat regions). However, complex characteristics result in more computational complexity of the MRISC CPU software in Figure 2.10, where image feature analysis results from the IFE are processed. Generally, MRISC processing in the beginning and the end of one picture is limited due to other heavy calculations such as picture bit rate control and picture encoding start/stop control for various processing blocks, therefore simpler parameters which can

Table 4.4 Percentage of flat regions in HDTV videos.

k	Video sequences				
	Twilight	**Nebuta**	**Lucore**	**Hawaii1**	**Hawaii2**
4	0.101%	0%	4.6%	21.0%	0.5%
3	9.03%	0%	28.5%	34.3%	12.6%
2	42.2%	22.4%	35.9%	51.4%	37.7%
1	54.9%	95.9%	97.5%	97.5%	83.8%

Table 4.5 BD-rate of fixed and adaptive ME method in HDTV videos.

(k,x)		Video sequences				
		Twilight	**Nebuta**	**Lucore**	**Hawaii1**	**Hawaii2**
Fixed *MSB-side* *4 bits*		*+2.2%*	*+3.7%*	*+6.2%*	*+3.1%*	*+2.6%*
Proposed Adaptive	(4, 0)	+2.2%	+3.7%	+6.1%	+3.2%	+2.7%
	(3, 1)	+2.0%	+3.7%	+6.1%	+3.2%	+2.6%
	(2, 2)	**+1.9%**	+4.4%	+6.6%	**+2.8%**	+2.6%
	(1, 3)	+2.4%	**+1.7%**	**+4.6%**	+3.4%	**+1.2%**

represent the threshold are desirable. For this reason, flat region percentage threshold is chosen.

4.5.3 Flat region percentage threshold in 4K/8K

In prior experiments, a flat region percentage threshold of more than 40% was derived in order to represent the maximum coding efficiency with HDTV video sequences. This

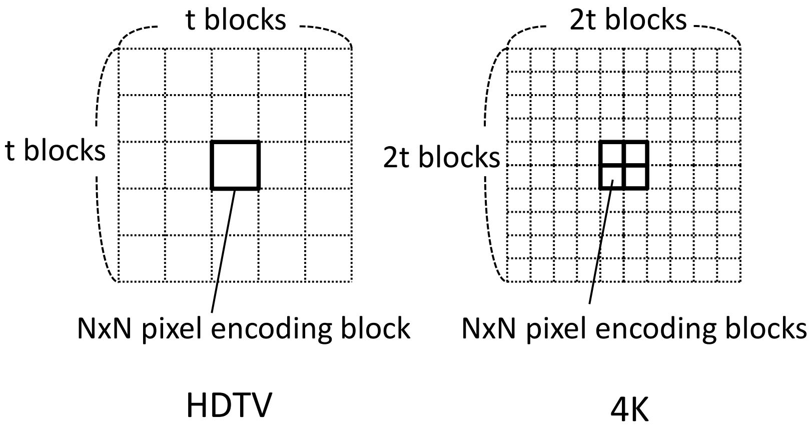


Figure 4.6 Search area modeling with the same view angle.

threshold should be modified for 4K and 8K, according to the search area model described below.

Figure 4.6 shows a search area model of HDTV and 4K with the same view angle. In HDTV, assume that an N×N pixel encoding block has a motion search area containing t^2 blocks. Since 4K has double density pixel resolutions, a motion search area containing $4t^2$ blocks are required to follow the same motion of objects.

Let the flat region percentage of HDTV and 4K be $\mathrm{f_{HD}}$ and $\mathrm{f_{4K}}$. Now when an N×N block is flat, the probability that an adjacent block is also flat is described as $\mathrm{s_{HD}} \cdot \mathrm{f_{HD}}$ and $\mathrm{s_{4K}} \cdot \mathrm{f_{4K}}$ (where $\mathrm{s_{HD}}$ and $\mathrm{s_{4K}}$ are "flat probability enhancement factors" when an adjacent block is flat). In Fig. 7, when an encoding block and a reference picture block in the same position are flat, the probability that the entire search area is flat is $(\mathrm{s_{HD}} \cdot \mathrm{f_{HD}})^{\mathrm{t}^2}$ and$(\mathrm{s_{4K}} \cdot \mathrm{f_{4K}})^{4\mathrm{t}^2}$. To make these probabilities equal for HDTV and 4K, the equation

$$\mathrm{f_{4K}} = \frac{\sqrt[4]{\mathrm{s_{HD}} \cdot \mathrm{f_{HD}}}}{\mathrm{s_{4K}}}$$

is derived and by applying the same relationship between 4K and 8K,

$$\mathrm{f_{8K}} = \frac{\sqrt[4]{\mathrm{s_{4K}} \cdot \mathrm{f_{4K}}}}{\mathrm{s_{8K}}}$$

is derived.

With the four video sequences listed in Table 4.6, the averages of s_{HD}, $\mathrm{s}_{4\mathrm{K}}$ and $\mathrm{s}_{8\mathrm{K}}$ are calculated as 1.96, 1.69, 1.61 and by applying the above equations and $\mathrm{f}_{\mathrm{HD}} = 40\%$,

$$\mathrm{f}_{4\mathrm{K}} = 56\%,\ \mathrm{f}_{8\mathrm{K}} = 61\%$$

are obtained. These values are used for the threshold percentage of 4K and 8K.

4.5.4 Coding efficiency assessment with 4K/8K video

As same as Section 4.5.2, with the power reduction results obtained in Table 4.3 with 4 bits, coding efficiency improvement with the proposed adaptive method was assessed, but this time with 4K and 8K videos and with threshold percentages obtained in Section 4.5.3.

Table 4.6 Video sequences used for evaluation (33 frames each).

Name	Content	Resolution
Lucore	A colorful parrot, HDR	4K
Nebuta	Very dark crowds, SDR	8K (4K down sampled)
Hawaii1	Bright beach at noon, HDR	8K (4K down sampled)
Hawaii2	Dark beach at dawn, HDR	8K (4K down sampled)

Table 4.7 Percentage of flat regions.

k	Lucore			Nebuta		
	HD	4K	8K	HD	4K	8K
4	4.6%	4.4%	-	0.0%	0.0%	0.0%
3	28.5%	37.9%	-	0.0%	0.0%	0.0%
2	35.9%	50.5%	-	22.4%	39.9%	59.8%
1	**97.5%**	**98.8%**	-	**95.9%**	**97.7%**	**99.1%**

k	Hawaii1			Hawaii2		
	HD	4K	8K	HD	4K	8K
4	21.0%	30.9%	43.5%	0.5%	3.6%	12.4%
3	34.3%	48.8%	**62.1%**	12.6%	25.1%	38.5%
2	**51.4%**	**63.7%**	73.5%	37.7%	47.3%	54.8%
1	51.5%	64.3%	74.6%	**83.8%**	**90.3%**	**94.1%**

The video sequences listed in Table 4.6 are used for comparison, one 4K content (Lucore) and three 8K contents (Nebuta, Hawaii1, Hawaii2). Twilight was omitted this time, because it only has an HDTV resolution video. Only Nebuta has standard dynamic range (SDR) signals and the other three sequences comply with HDR. In order to reduce the influence of Gaussian noise included in the video contents, noise filtering (3-tap for 4K and 5-tap for 8K contents) is applied beforehand.

Flat region percentages with different k values, which should be assessed in IFE before encoding, are shown in Table 4.7. The values for HDTV is as same as in Table 4.4 in Section 4.5.2 and the values for 4K and 8K is new here. The underlined figure in each content deserves the largest k value that has a flat region percentage larger than the

Table 4.8 BD-rate of fixed and adaptive ME method.

		Lucore			Nebuta		
(k, x)		**HD**	**4K**	**8K**	**HD**	**4K**	**8K**
Fixed *MSB-side* *4 bits*		*+6.2%*	*+8.6%*	-	*+3.7%*	*+3.4%*	*+3.8%*
Proposed adaptive	(4, 0)	+6.1%	+8.4%	-	+3.7%	+3.4%	+3.8%
	(3, 1)	+6.1%	+8.1%	-	+3.7%	+3.4%	+3.8%
	(2, 2)	+6.6%	+8.3%	-	+4.4%	+3.2%	+2.7%
	(1, 3)	**+4.6%**	**+6.1%**	-	**+1.7%**	**+1.3%**	**+1.7%**

		Hawaii1			Hawaii2		
(k, x)		**HD**	**4K**	**8K**	**HD**	**4K**	**8K**
Fixed *MSB-side* *4 bits*		*+3.1%*	*+4.3%*	*+4.4%*	*+2.6%*	*+3.5%*	*+4.8%*
Proposed adaptive	(4, 0)	+3.2%	+4.0%	+3.2%	+2.7%	+3.4%	+4.8%
	(3, 1)	+3.2%	+3.7%	**+3.0%**	+2.6%	+3.5%	+4.7%
	(2, 2)	**+2.8%**	**+3.4%**	+3.8%	+2.6%	+3.5%	+4.8%
	(1, 3)	+3.4%	+4.5%	+5.3%	**+1.2%**	**+1.4%**	**+2.6%**

threshold discussed in Section 4.5.3.

We then encoded the video sequences with H.265/HEVC 4:2:0 10bit (Main 10 Profile) and calculated the BD-rate, which denotes the extra coding bits required for the same PSNR, compared to the original 8-bit WME and MME blocks. The results are shown in Table 4.8.

Again the values for HDTV is as same as in Table 4.5 in Section 4.5.2 and the values for 4K and 8K is new here. SAD calculation with fixed MSB-side 4 bits results in a BD-rate increase because of insufficient precision. The increase, however, is suppressed with the adaptive bit-reduced ME techniques as a result of the increased precision in block matching in the flat regions. The underlined figure in each content achieves the best BD-rate increase suppression and well matches the judgment of k values with the modified flat region threshold in Table 4.7. BD-rate increases are best suppressed in Nebuta 4K (from +3.4% to +1.3%, 62% suppression) and Nebuta 8K (from +3.8% to +1.7%, 55% suppression), showing the proposed ME techniques achieve good coding efficiency even with halved bit width.

4.6 Chapter summary

In this chapter, the adaptive bit-shortened motion estimation engine was proposed for real-time high efficiency video coding encoders and evaluated its usefulness. It enables power consumption to be reduced by 18-39% and coding loss to be suppressed by up to 62% in 4K and to 55% in 8K. In future work, the author intends to further apply this to other ME blocks and assess its effectiveness as a solution to enhance the development of the next generation ultra-high definition video encoders.

Chapter 5
Concluding remarks

This dissertation has discussed video encoder VLSI architecture and supporting techniques to realize super-high-resolution real-time video encoder VLSIs with high image quality. In order to achieve high quality video encoder VLSIs for now and to keep them prevailing not only for broadcasting and distribution use but also for mobile applications in the future, three major issues have to be solved: 1) Real time processing yet maintaining high image quality, 2) Support for higher resolution over single-chip capability, and 3) Deeper reduction in circuit scale and power consumption.

At first, to solve the issue 1), the video encoding core architecture of real time video encoder VLSIs aiming at broadcast level video quality was introduced. Especially, the “prediction core which handles the intra/inter prediction, motion estimation and determination of prediction modes were intensely described for two standards of H.264/AVC and H.265/HEVC. For major requirements for achieving professional quality real-time video encoders were first presented and techniques to overcome them are described in detail. The “telescopic + inclusive” motion estimation with programmable fractional motion estimation and mode decision with SIMD processors were introduced for H.264/AVC. The statistically adaptive WME and aggregated multi-block-size MME, edge-based intra prediction in MED and IPD, deeply centered high-speed mode decision at IIM while controllable with scale and offset values, and supporting high-speed reference image feed were also presented for H.265/HEVC. The coding quality evaluation results showed that the proposed techniques for real-time video encoders had a sufficient coding efficiency while achieving real-time operation of HDTV and super-high-resolution videos.

Secondly, to address the issue 2), multi-chip configuration of real-time video encoder VLSIs for parallel encoding of super-high-resolution videos or multi-channel videos was presented. An inter-chip connection method for super-high-resolution encoding with multiple VLSIs was described, which has been installed on the latest 4K real-time H.265/HEVC video encoder VLSI "NARA" and utilized for 8K real-time encoders. In addition, an inter-chip flexible stream output technique was also proposed, which could

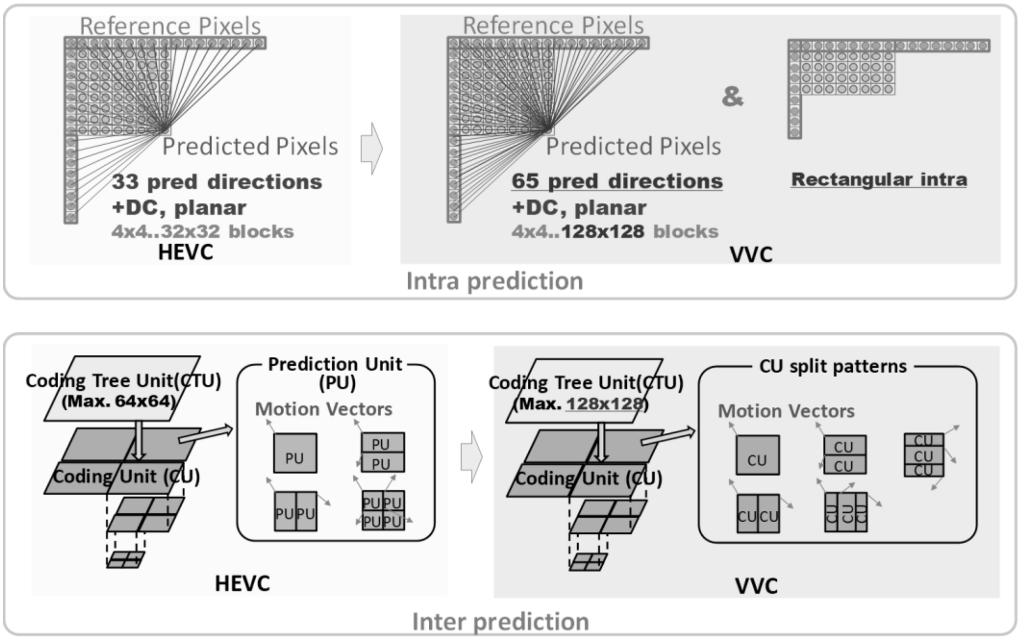


Figure 5.1 Comparison of coding modes between H.265/HEVC and VVC.

accommodate both super-high-resolution and multi-channel encoding and construct MPEG-2 transport streams without external multiplexer devices. This has been installed in the HDTV MPEG-2 encoder VLSI "VASA" and multiplexing quality of the output stream was assessed, where TS packets from multiple VLSIs were successfully multiplexed into one TS outputs with daisy-chained paths.

Finally, in order to solve 3), an adaptive bit-shifted motion estimation engine was proposed for real-time high efficiency video coding encoders and evaluated its usefulness. It enables power consumption to be reduced by 18-39% and coding loss to be suppressed by up to 62% in 4K and to 55% in 8K. In future work, the author intends to further apply this to other ME blocks and assess its effectiveness as a solution to enhance the development of the next generation ultra-high definition video encoders.

Toward the next generation video encoding, the above-mentioned three issues will become more crucial because newer video coding standards tend to have more combination of motion vectors, prediction candidates and encoding modes that used to be omitted due to the heavy computational complexity, but the core encoding structure of “pixel value prediction plus DCT-based transformed residuals, with entropy coding” still remains to be the same. For example, Figure 5.1 shows the comparison of coding modes between the H.265/HEVC and the next coding standard VVC[14] which is now under standardization process and will be effective in the year 2020. Intra prediction modes are increased to 65 with block sizes from 4×4 through 128×128 with rectangular intra blocks allowed, and inter prediction modes are also extended to the maximum of 128×128 pixels CTU with the new ternary tree split patterns. Encoding time comparison of 4K contents with H.265/HEVC and VVC reference software shows that the VVC requires about eight times longer encoding time compared to H.265/HEVC.

It is expected, however, that the proposed intra and inter prediction techniques described in Section 2.3 are easily applicable to the new standard because the basic structure of intra and inter predictions are maintained, and furthermore the proposed technique will be more effective for complexity reduction of increased coding modes. The proposed architecture and techniques, therefore, will still be applicable and further be more effective in the future.

In the meantime, rapid progress of deep neural network (DNN) technologies and their applications for image creation using autoencoders and generative adversarial networks (GANs), so called “imagery coding” [69] is emerging. This coding approach completely differs from the current video coding standards and at the decoding side images of what resembles the original images in human perception is generated with limited signals. In this approach, picture PSNR becomes completely meaningless and decoding picture will be “different from original images, but the meaning and visual perception for human eyes are almost the same.” And also, perceptual picture quality will be quite better than conventional video codecs. When these technologies go into practical use, real-time video coding VLSI architecture will become quite different from current ones, and partly resembles real-time DNN inference architectures. Even if video codecs for video contents consumed by humans become autoencoder-based or GAN-based structures, there still will be vaster spaces for video codecs for machine vision, which mean, video big data consumed and analyzed by artificial intelligence (AI) cloud computers. In these applications, less distortion (i.e. high PSNR) is much more important than human perceptual quality, therefore the proposed techniques for better PSNR with fewer bits continue to work well with the video codec world for a long time.

[14] The name of the new standard will be H.266/VVC (Versatile Video Coding), however, it is now under standardization process and called VVC instead.

From now on, the author will seek for techniques for more resource-efficient and better picture quality video codecs based on the research in the dissertation, in both human perceptual and better PSNR ways.

Bibliography

[1] ITU-T, "ITU-T Recommendation H.261: Line transmission of nontelephone signals: Video codec for audiovisual services at p x 64 kbits," Nov. 1988.

[2] G. J. Sullivan et al, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. CSVT 22,* pp. 1649-1668, Dec. 2012.

[3] ISO/IEC, "ISO/IEC 11172-2: Information technology - Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s - Part2: Video," Aug. 1993.

[4] ISO/IEC and ITU-T, "ITU-T Recommendation H.262 and ISO/IEC 13818-2: Information technology - Generic coding of moving pictures and associated audio information: Video," Nov. 1994.

[5] ITU-T, "ITU-T Recommendation H.263: Video coding for low bit rate communication," Mar. 1996.

[6] ISO/IEC, "ISO/IEC 14496-2: Information technology ? Coding of Audio-Visual Objects - Part 2: Visual," Dec 2000.

[7] ITU-T and ISO/IEC, "ITU-T Recommendation H.264 and ISO/IEC 14496-10: Information technology - Coding of Audio-Visual Objects?Part 10: Advanced Video Coding," Mar. 2003.

[8] ITU-T and ISO/IEC, "ITU-T Recommendation H.265 and ISO/IEC 23008-2 MPEG-H part 2: High efficiency video coding," Apr. 2013.

[9] ITU-T and ISO/IEC, "ITU-T Recommendation H.222.0 and ISO/IEC 13818-1:2018 Generic coding of moving pictures and associated audio information - Part 1: Systems,"

Mar. 2018.

[10] K. Suguri, T. Minami, H. Matsuda, R. Kusaba, T. Kondo, R. Kasai, T.Watanabe, H. Sato, N. Shibata, Y. Tashiro, T. Izuoka, A. Shimizu, and H. Kotera, "A real-time motion estimation and compensation LSI with wide search range for MPEG2 video encoding," *IEEE Journal of Solid-State Circuits, vol. 31, no. 11,* pp. 1733-1741, Nov. 1996.

[11] T. Kondo, K. Suguri, M. Ikeda, T. Abe, H. Matsuda, T. Okubo, K. Ogura, Y. Tashiro, N. Ono, T. Minami, R. Kusaba, T. Ikenaga, N. Shibata, R. Kasai, K. Otsu, F. Nakagawa, and Y. Sato, "Two-chip MPEG-2 video encoder," *IEEE Micro, vol. 16, no. 2,,* pp. 51-58, Apr. 1996.

[12] M. Ikeda, T. Okubo, T. Abe, Y. Itoh, Y. Tashiro, and R. Kasai, "A hardware/software concurrent design for a real-time SP@ML MPEG2 video encoder chip set," *Proceedings of the European Design & Test Conference 1996 (ED&TC). IEEE,* pp. 320-326, Mar. 1996.

[13] T. Minami, T. Kondo, K. Nitta, K. Suguri, M. Ikeda, T. Yoshitome, H. Watanabe, H. Iwasaki, K. Ochiai, J. Naganuma, M. Endo, E. Yamagishi, T. Takahishi, K. Tadaishi, Y. Tashiro, N. Kobayashi, T. Okubo, T. Ogura, and R. Kasai, "A single-chip MPEG2 MP@ML video encoder with multi-chip configuration for a single-board MP@HL encoder," *Proceedings of HOT Chips X. IEEE,* pp. 123-131, Aug. 1998.

[14] K. Nitta, T. Minami, T. Kondo, and T. Ogura, "Motion estimation/motion compensation hardware architecture for a scene-adaptive algorithm on a single-chip MPEG2 MP@ML video encoder," *Proceedings of IS&T/SPIE Conference on Visual Communications and Image Processing ’99, vol. 3653. IS&T/SPIE,* pp. 874-882, Jan. 1999.

[15] M. Ikeda, T. Kondo, K. Nitta, K. Suguri, T. Yoshitome, T. Minami, H. Iwasaki, K. Ochiai, J. Naganuma, M. Endo, Y. Tashiro, H. Watanabe, N. Kobayashi, T. Okubo, T. Ogura, and R. Kasai, "SuperENC: MPEG-2 video encoder chip," *IEEE Micro, vol. 19, no. 4,* pp. 56-65, July/Aug. 1999.

[16] M. Ikeda, T. Kondo, K. Nitta, K. Suguri, T. Yoshitome, T. Minami, J. Naganuma, and T. Ogura, "An MPEG-2 video encoder LSI with scalability for HDTV based on three-layer cooperative architecture," *Design Automation and Test in Europe Conference 1999.*

IEEE, pp. 44-50, Mar. 1999.

[17] M. Ikeda, T. Kondo, K. Nitta, K. Suguri, T. Yoshitome, T. Minami, J. Naganuma, and T. Ogura, "Three-layer cooperative architecture for MPEG-2 video encoder LSI," *IEICE Transactions on Electronics, vol. E80-C, no. 2,* pp. 170-178, Feb. 2000.

[18] K. Nitta, T. Minami, T. Kondo and T. Ogura, "Motion estimation and compensation hardware architecture for a scene-adaptive algorithm on a single-chip MPEG2 video encoder," *IEICE Transactions on Information and Systems, vol. E84-D, no. 3,* pp. 317-325, Mar. 2001.

[19] K. Nitta, T. Yoshitome, T. Kondo, H. Iwasaki, and J. Naganuma, "Improvements on SIMD macroblock processor in MPEG-2 video encoder LSI," *IEICE Transactions on Electronics, vol. J87-C, no. 4 (in Japanese),* pp. 377-385, Apr. 2004.

[20] M. Endo, J. Naganuma, Y. Nakajima, and T. Ogura, "An MPEG-2 encoding PC card system for real-time mobile applications," *Proceedings of International Conference on Consumer Electronics 2001 (ICCE2001), IEEE,* pp. 160-161, Jan. 2001.

[21] K. Suguri, T. Yoshitome, M. Ikeda, T. Kondo, and T. Ogura, "A scalable architecture of real-time MP@HL MPEG-2 video encoder for multi-resolution video," *Proceedings of IS&T/SPIE Conference on Visual Communications and Image Processing '99, vol. 3653. IS&T/SPIE,* pp. 895-904, Jan. 1999.

[22] T. Yoshitome, T. Minami, M. Ikeda, K. Nitta, and K. Suguri, "A 4:2:2P@ML MPEG-2 video encoder board using and enhanced MP@ML video encoder LSI," *IEEE Transactions on Consumer Electronics, vol. 45, no. 4,* pp. 1130-1133, Nov. 1999.

[23] H. Iwasaki, J. Naganuma, K. Nitta, K. Nakamura, T. Yoshitome, M. Ogura, Y. Nakajima, Y. Tashiro, T. Onishi, M. Ikeda, and M. Endo, "Single-chip MPEG-2 422P@HL CODEC LSI with multi-chip configuration for large scale processing beyond HDTV level," *Proceedings of Design, Automation and Test in Europe Conference and Exhibition,* pp. 2-7, Mar. 2003.

[24] H. Iwasaki, J. Naganuma, K. Nitta, K. Nakamura, T. Yoshitome, M. Ogura, Y. Nakajima, Y. Tashiro, T. Onishi, M. Ikeda, T. Minami, M. Endo, and Y. Yashima, "Single-chip MPEG-2 422P@HL CODEC LSI with multichip configuration for large scale processing

beyond HDTV level," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, vol. 15, no. 9,* pp. 1055-1059, Sep. 2007.

[25] Takayuki Onishi, Mitsuo Ikeda, Jiro Naganuma, Makoto Endo and Yoshiyuki Yashima, "A Distributed TS-MUX Architecture for Multi-chip Extension Beyond the HDTV Level," *IEEE 2004 International Symposium Circuits And Systems (ISCAS2004) II,* pp. 261-264, May 2004.

[26] Takayuki Onishi, Ken Nakamura, Takeshi Yoshitome, and Jiro Naganuma, "A Distributed Stream Multiplexing Architecture for Multi-Chip Configuration beyond HDTV," *IEICE Transactions on Information and Systems Vol.E91-D No.12,* pp. 2862-2867, Dec. 2008.

[27] M. Ikeda, H. Iwasaki, K. Nitta, T. Onishi, T. Sano, A. Sagata, Y. Nakajima, M. Inamori, T. Yoshitome, H. Matsuda, R. Tanida, A. Shimizu, and J. Naganuma, "A professional H.264/AVC CODEC chip-set for HDTV broadcast infrastructure and high-end flexible CODEC systems," *Symposium on High Performance Chips (HOT CHIPS 19),* Aug. 2007.

[28] K. Nitta, M. Ikeda, H. Iwasaki, T. Onishi, T. Sano, A. Sagata, Y. Nakajima, M. Inamori, T. Yoshitome, H. Matsuda, R. Tanida, A. Shimizu, K. Nakamura, and J. Naganuma, "An H.264/AVC High422 profile and MPEG-2 422 profile encoder LSI for HDTV broadcasting infrastructures," *VLSI Circuits, 2008 IEEE Symposium on,* pp. 106-107, June 2008.

[29] K. Nitta, H. Iwasaki, T. Onishi, T. Sano, A. Sagata, Y. Nakajima, M. Inamori, R. Tanida, A. Shimizu, K. Nakamura, M. Ikeda, and J. Naganuma, "An H.264/AVC High422 profile and MPEG-2 422 profile encoder LSI for HDTV broadcasting infrastructures," *IEICE Transactions on Electronics, vol. E95-C, no. 4,* pp. 432-440, Apr. 2012.

[30] Takayuki Onishi, Takashi Sano, Koyo Nitta, Mitsuo Ikeda, and Jiro Naganuma, "Multi-Reference and Multi-Block-Size Motion," *IEEE International Symposium on Circuits and Systems (ISCAS 2008),* pp. 800-803, May 2008.

[31] Takayuki Onishi, Koyo Nitta, Takashi Sano, Hiroe Iwasaki, Mitsuo Ikeda, Jiro Naganuma and Kazuto Kamikura, "A Motion Estimation and Motion Compensation

Architecture for Professional H.264/AVC Encoder LSI," *The IEICE transactions on information and systems , vol. 93, no. 10 (in Japanese),* pp. 2148-2155, Oct. 2010.

[32] Takayuki Onishi, Takashi Sano, Yukikuni Nishida, Kazuya Yokohari, Jia Su, Ken Nakamura, Koyo Nitta, Kimiko Kawashima, Jun Okamoto, Naoki Ono, Ritsu Kusaba, Atsushi Sagata, Hiroe Iwasaki, Mitsuo Ikeda and Atsushi Shimizu, "Single-chip 4K 60fps 4:2:2 HEVC video encoder LSI with 8K scalability," *Digest of IEEE Symposia on VLSI Circuits,* pp. C54-C55, June 2015.

[33] Takayuki Onishi, Takashi Sano, Yukikuni Nishida, Kazuya Yokohari, Ken Nakamura, Koyo Nitta, Kimiko Kawashima, Jun Okamoto, Naoki Ono, Atsushi Sagata, Hiroe Iwasaki, Mitsuo Ikeda and Atsushi Shimizu, "Single-chip 4K 60fps 4:2:2 HEVC video encoder LSI with 8K scalability," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol.26, no. 10,* pp. 1930-1938, Oct. 2018.

[34] (in Japanese), Mar. 2017. [Online]. Available: https://game.watch.impress.co.jp/docs/news/1047807.html.

[35] Tung-Chien Chen, Yu-Wen Huang, and Liang-Gee Chen, "Fully utilized and reusable architecture for fractional motion estimation of H.264/AVC," *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '04), V-9-12, vol. 5,* May 2004.

[36] G.J.Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Process. Mag., vol. 15, no. 7,* pp. 74-90, Nov. 1998.

[37] Wen Shi, et al., "Edge information based fast selection algorithm for intra prediction of HEVC," *2014 IEEE Asia Pacific Conf. on Circuits and Systems (APCCAS),* pp. 17-20, Nov. 2014.

[38] Mohammadreza Jamali, et al., "Fast HEVC Intra Mode Decision Based on Edge Detection and SATD Costs Classification," *2015 Data Compression Conference,* pp. 43-52, July 2015.

[39] Shihao Wang, et al., "VLSI Implementation of HEVC Motion Compensation with Distance Biased Direct Cache Mapping for 8K UHDTV Applications," *IEEE Trans.*

Circuits Syst. Video Technol., vol. 27, no. 2, pp. 380-393, Feb. 2017.

[40] Yuya Omori,, Takayuki Onishi, Hiroe Iwasaki and Atsushi Shimizu, "A 120 fps high frame rate real-time HEVC video encoder with parallel configuration scalable to 4K," *Proc. IEEE Symp. Low-Power and High-Speed Chips (COOL CHIPS),* pp. 1-3, Apr. 2017.

[41] Jianbin Zhou, et al., "VLSI architecture of HEVC intra prediction for 8K UHDTV applications," *Proc. IEEE Int. Conf. Image Processing (ICIP) 2014,* pp. 1273-1277, Dec. 2014.

[42] Jia Zhu, et al., "HDTV1080p HEVC Intra encoder with source texture based CU/PU mode pre-decision," *2014 19th Asia and South Pacific Design Automation Conf. (ASP-DAC),* pp. 367-372, Jan. 2014.

[43] Shiaw-Yu Jou, et al., "Fast Motion Estimation Algorithm and Design for Real Time QFHD High Efficiency Video Coding," *IEEE Trans. Circuits Syst. Video Technol., vol. 25, no. 9,* pp. 1533-1544, Sep. 2015.

[44] Mani Laxman Aiyar, et al., "A 2260 GOPS High-Performance and High-Precision Sub-pixel Motion Estimator-Interpolator for Real-Time 8K UHDTV for HEVC Coding in Next Generation Wireless Multimedia Applications," *Proc. IEEE 6th Int. Conf. Advanced Computing (IACC),* pp. 695-700, Feb. 2016.

[45] Gang He, et al., "High-throughput power-efficient VLSI architecture of fractional motion estimation for Ultra-HD HEVC video encoding," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 23, no. 12,* pp. 3138-3142, Dec. 2015.

[46] Sung-Fang Tsai, et al., "A 1062Mpixels/s 8192x4320p High Efficiency Video Coding (H.265) Encoder Chip," *Dig. Symp. VLSI Circuits,* pp. 188-189, June 2013.

[47] Sukho Lee, et al., "Reduced complexity single core based HEVC video codec processor for mobile 4K-UHD applications," *Proc. IEEE Int. Conf. Consum. Electron. (ICCE) 2016,* pp. 94-95, Sep. 2016.

[48] Jinjia Zhou, et al., "100x Evolution of Video Codec Chips," *Proc. IEEE/ACM Int. Symp. Physical Design (ISPD),* pp. 121-122, Mar. 2017.

[49] Kazuhisa Iguchi, et al., "HEVC encoder for Super Hi-Vision," *Proc. IEEE Int. Conf. Consum. Electron. (ICCE) 2014,* pp. 57-58, Sep. 2014.

[50] Tse Kai Heng, et al., "A highly parallelized H.265/HEVC real-time UHD software encoder," *Proc. IEEE Int. Conf. Image Processing (ICIP) 2014,* pp. 1213-1217, Sep. 2014.

[51] Ronan Parois, et al., "Real-time UHD scalable multi-layer HEVC encoder architecture," *24th European Signal Processing Conference (EUSIPCO),* pp. 1298-1302, Aug. 2016.

[52] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," *ITU-T Q.6/SG16 VCEG 13th meeting, VCEG-M33,* Apr. 2001.

[53] ITU-R, "Methodology for the subjective assessment of the quality of television pictures," *Recommendation BT.500,* Jan. 2012.

[54] S. Kumaki, H. Takata, Y. Ajioka, T. Ooishi, K. Ishihara, A. Hanami, T. Tsuji, T. Watanabe, C. Morishima, T. Yoshizawa, H. Sato, S. Hattori, A. Koshio, K. Tsukamoto, and T. Matsumura, "A 99-mm2 0.7-W Single-Chip MPEG-2 422P@ML Video, Audio, and System Encoder With a 64-Mb Embedded DRAM for Portable 422P@HL Encoder System," *IEEE J. Solid State Circuits}, vol. 37, no. 3,* pp. 450-454, Mar. 2002.

[55] T. Yoshitome, K. Nakamura, Y. Yashima, and M. Endo, "A scalable architecture for use in an over-HDTV real-time codec system for multiresolution video," *SPIE Visual Communication and Image Processing (VCIP2003),* pp. 1752-1759, July 2003.

[56] Takahiro Nishi, "Bit stream structures and its functionaliies," *ITE Transactions Special Issue vol.67 no.7,* pp. 549-552, Jul. 2013 (In Japanese).

[57] Association of Radio Industries and Businesses (ARIB), "Video Coding, Audio Coding and Multiplexing Specifications for Digital Broadcasting," *ARIB STD-B32 version 3.9,* Dec. 2016.

[58] M. Inamori, J. Naganuma and M. Endo, "A memory-based architecture for MPEG2 system protocol LSIs," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Volume: 7 , Issue: 3,* pp. 339-344, Sep. 1999.

[59] K. Nakamura, T. Yoshitome and Y. Yashima, "Super High Resolution Video CODEC System with Multiple MPEG-2 HDTV CODEC LSIs," *IEEE International Symposium on Circuits and Systems (ISCAS 2004), Vol. III,* pp. 793-796, May 2004.

[60] Obianuju Ndili, Tokunbo Ogunfunmi, "Hardware-oriented Modified Diamond Search for Motion Estimation in H.246/AVC," *Proc. IEEE ICIP,* pp. 749-752, Sep. 2010.

[61] Pablo Montero, Javier Taibo, "Fast GPU approximation of EPZS motion estimation," *Proc. IEEE MMSP,* pp. 356-361, Oct. 2013.

[62] Gustavo Sanchez, Marcelo Porto, Luciano Agostini, "A hardware friendly motion estimation algorithm for the emergent HEVC standard and its low power hardware design," *Proc. IEEE ICIP,* pp. 1991-1994, Sep. 2013.

[63] Shan Zhu and Kai-Kuang Ma, "A New Diamond Search Algorithm for Fast Block-Matching Motion Estimation," *IEEE Trans. Image Processing, Vol. 9, No. 2,* pp. 287-290, Feb. 2000.

[64] Zhibo Chen, Peng Zhou, Yun He, "Fast Integer Pel and Fractional Pel Motion Estimation for JVT," *JVT-F017rl.doc, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG,* Dec. 2002.

[65] Alexis M. Tourapis, "Enhanced Predictive Zonal Search for Single and Multiple Frame Motion Estimation," *Proc. IEEE VCIP,* pp. 1069-1079, Jan. 2002.

[66] Arnab Raha, Hrishikesh Jayakumar, Vijay Raghunathan, "A Power Efficient Video Encoder using Reconfigurable Approximate Arithmetic Units," *Proc. IEEE VLSI Design,* pp. 324-329, Jan. 2014.

[67] Alp Erturk, Sarp Erturk, "Two-Bit Transform for Binary Block Motion Estimation," *IEEE Trans. CSVT, Vol. 15, No. 7,* pp. 938-946, Jul. 2005.

[68] Abdulkadir Akin, Yigit Dogan, Ilker Hamzaoglu, "High performance hardware architectures for one bit transform based motion estimation," *IEEE Trans. Consumer Electronics 2019,* pp. 941-949, Apr. 2009.

[69] Tetsuo Nozawa, "Already surpassing rule-based code generation, ultra-high efficiency according to the person's "image" standard," *Nikkei Electronics Magazine July 2019*

Issue, pp. 55-59, Jul. 2019 (In Japanese).

[70] Koyo Nitta, "Motion Estimation and Compensation Hardware Architecture with Hierarchy of Flexibility in Video Encoder LSIs," *Doctoral dissertation of Kyoto University,* Mar. 2015.

Appendix A
List of publications

Journal articles related to this dissertation

1. Takayuki Onishi, Ken Nakamura, Takeshi Yoshitome, and Jiro Naganuma, “A Distributed Stream Multiplexing Architecture for Multi-Chip Configuration beyond HDTV,” IEICE Transactions on Information and Systems Vol.E91-D, No.12, pp. 2862-2867, Dec. 2008. **[Chapters covered: Chapter 3]**

2. Takayuki Onishi, Koyo Nitta, Takashi Sano, Hiroe Iwasaki, Mitsuo Ikeda, Jiro Naganuma and Kazuto Kamikura, “A Motion Estimation and Motion Compensation Architecture for Professional H.264/AVC Encoder LSI,” IEICE Transactions on Information and Systems (Japanese Edition) Vol. J93-D, No. 10, pp. 2148-2155, Oct. 2010 (In Japanese). **[Chapters covered: Chapter 2]**

3. Takayuki Onishi, Takashi Sano, Yukikuni Nishida, Kazuya Yokohari, Ken Nakamura, Koyo Nitta, Kimiko Kawashima, Jun Okamoto, Naoki Ono, Atsushi Sagata, Hiroe Iwasaki, Mitsuo Ikeda, and Atsushi Shimizu, “A Single-chip 4K 60fps 4:2:2 HEVC Video Encoder LSI Employing Efficient Motion Estimation and Mode Decision Framework with Scalability to 8K,” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol.26, no. 10. pp. 1930-1938, Oct. 2018. **[Chapters covered: Chapter 2, 3]**

4. Takayuki Onishi, Yuya Omori, Ken Nakamura, Hiroe Iwasaki, Atsushi Shimizu and Hiroshi Nakamura, “A Low Power Motion Estimation Engine for Real-time 4K/8K Video Encoders,” IEICE Transactions on Information and Systems (Under review). **[Chapters covered: Chapter 4]**

Refereed conference papers related to this dissertation

1. Takayuki Onishi, Mitsuo Ikeda, Jiro Naganuma, Makoto Endo and Yoshiyuki Yashima, "A distributed TS-MUX architecture for multi-chip extension beyond the HDTV level," in *Proceedings of 2004 IEEE International Symposium on Circuits and Systems (ISCAS 2004)*, pp. II-261-264, Sep. 2004. **[Chapters covered: Chapter 3]**

2. Takayuki Onishi, Takashi Sano, Koyo Nitta, Mitsuo Ikeda and Jiro Naganuma, "Multi-reference and multi-block-size motion estimation with flexible mode selection for professional 4:2:2 H.264/AVC encoder LSI," in *Proceedings of 2008 IEEE International Symposium on Circuits and Systems (ISCAS 2008)*, pp. 800-803, Sep. 2008. **[Chapters covered: Chapter 2]**

3. Takayuki Onishi, Takashi Sano, Yukikuni Nishida, Kazuya Yokohari, Jia Su, Ken Nakamura, Koyo Nitta, Kimiko Kawashima, Jun Okamoto, Naoki Ono, Ritsu Kusaba, Atsushi Sagata, Hiroe Iwasaki, Mitsuo Ikeda and Atsushi Shimizu, "Single-chip 4K 60fps 4:2:2 HEVC video encoder LSI with 8K scalability," in *Proceedings of 2015 Symposium on VLSI Circuits (VLSI Circuits 2015)*, DOI: 10.1109/VLSIC.2015.7231325, June 2015. **[Chapters covered: Chapter 2]**

4. Takayuki Onishi, Yuya Omori, Ken Nakamura, Hiroe Iwasaki and Atsushi Shimizu, "A Low Power Motion Estimation Engine with Adaptive Bit-Shifted SAD Calculation," in *Proceedings of 2019 IEEE International Symposium on Circuits and Systems (ISCAS 2019)*, DOI: 10.1109/ISCAS.2019.8702287, May 2019. **[Chapters covered: Chapter 4]**

Reports related to this dissertation

1. Takayuki Onishi, Jiro Naganuma, Hiroe Iwasaki, Koyo Nitta, Ken Nakamura, Takeshi Yoshitome, Mitsuo Ogura, Yasuyuki Nakajima, Yutaka Tashiro, Mitsuo Ikeda, Makoto Endo, and Yoshiyuki Yashima, "Single chip MPEG-2 422P@HL CODEC LSI (VASA) — An extensible MUX part architecture —," in *Proceedings of Forum on Information Technology (FIT2003), J-104*, Sep. 2003 (in Japanese). **[Chapters covered: Chapter**

3]

2. Takayuki Onishi, Mitsuo Ikeda, Hiroe Iwasaki, Koyo Nitta, Takashi Sano, Atsushi Sagata, Yasuyuki Nakajima, Minoru Inamori, Takeshi Yoshitome, Hiroaki Matsuda, Ryuichi Tanida, Atsushi Shimizu, Ken Nakamura, and Jiro Naganuma, "SARA: A Professional H.264/AVC Encoder LSI for HDTV CODEC Systems—Search Module Configuration —," in *Proceedings of IEICE General Conference 2008, D–11–112, Vol 2008. 2*, Mar. 2008 (in Japanese). **[Chapters covered: Chapter 2]**

3. Takayuki Onishi, Takashi Sano, Yukikuni Nishida, Ritsu Kusaba, Atsushi Sagata, Hiroe Iwasaki, Mitsuo Ikeda and Atsushi Shimizu, "Study of Parallel Encoding Framework for UHDTV," in *Proceedings of Forum on Information Technology (FIT2014), I-031,* Sep. 2009 (in Japanese). **[Chapters covered: Chapter 3]**

4. Takayuki Onishi, Yuya Omori, Hiroe Iwasaki and Atsushi Shimizu, "A Power saving Method for Real time HEVC Encoder LSIs," in *IEICE technical report, vol. 116, no. 334, ICD2016-44*, pp. 33-38, Nov. 2016 (in Japanese) **[Chapters covered: Chapter 4]**

Reviews related to this dissertation

1. Takayuki Onishi, Takashi Sano, Kazuya Yokohari, Jia Su, Mitsuo Ikeda, Atsushi Sagata, Hiroe Iwasaki and Atsushi Shimizu, "HEVC hardware encoder technology," *NTT Technical Journal*, Vol. 26, No. 2, pp. 51–54, Feb. 2014 (in Japanese). **[Chapters covered: Chapter 2]**

2. Takayuki Onishi, Takashi Sano, Kazuya Yokohari, Jia Su, Mitsuo Ikeda, Atsushi Sagata, Hiroe Iwasaki and Atsushi Shimizu, "HEVC hardware encoder technology," *NTT Technical Review, Vol. 12, No. 5*, May 2014. **[Chapters covered: Chapter 2]**

Patents related to this dissertation

1. Takayuki Onishi and Jiro Naganuma, JP Patent 3891035, Nippon Telephone and

Telegraph Corporation, Dec. 2006. **[Chapters covered: Chapter 3]**

2. Takayuki Onishi, Takashi Sano, Mitsuo Ikeda and Jiro Naganuma, JP Patent 4516088, Nippon Telephone and Telegraph Corporation, May 2010. **[Chapters covered: Chapter 2]**

3. Takayuki Onishi, Koyo Nitta and Jiro Naganuma, JP Patent 4430690, Nippon Telephone and Telegraph Corporation, May 2010. **[Chapters covered: Chapter 2]**

4. Takayuki Onishi, Takashi Sano, Hiroe Iwasaki and Kazuto Kamikura, JP Patent 5286573, Nippon Telephone and Telegraph Corporation, June 2013. **[Chapters covered: Chapter 2]**

5. Takayuki Onishi, Takashi Sano and Atsushi Shimizu, JP Patent 6053210, Nippon Telephone and Telegraph Corporation, Dec. 2016. **[Chapters covered: Chapter 2]**

6. Takayuki Onishi, Yuya Omori, Hiroe Iwasaki and Atsushi Shimizu, JP Patent Pending 2016-224344, Nippon Telephone and Telegraph Corporation, Nov. 2016. **[Chapters covered: Chapter 4]**

Other journal articles

1. Takayuki Onishi, Mitsuo Ikeda, Jiro Naganuma, Makoto Endo and Yoshiyuki Yashima, “Highly Accurate De-Jittering for Broadcast Quality Video Transmission,” IEICE Transactions on Information and Systems Pt.1 (Japanese Edition) Vol. J88-D-1, No. 2, pp. 353-360, Feb. 2005 (In Japanese).

2. Hiroe Iwasaki, Jiro Naganuma, Koyo Nitta, Ken Nakamura, Takeshi Yoshitome, Mitsuo Ogura, Yasuyuki Nakajima, Yutaka Tashiro, Takayuki Onishi, Mitsuo Ikeda, Toshihiro Minami, Makoto Endo and Yoshiyuki Yashima, “Single-Chip MPEG-2 422P@HL CODEC LSI With Multichip Configuration for Large Scale Processing Beyond HDTV Level,” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 15,

Issue 9, pp. 1055-1059, Sep. 2007.

3. Koyo Nitta, Hiroe Iwasaki, Takayuki Onishi, Takashi Sano, Atsushi Sagata, Yasuyuki Nakajima, Minoru Inamori, Ryuichi Tanida, Atsushi Shimizu, Ken Nakamura, Mitsuo Ikeda, and Jiro Naganuma, "An H.264/AVC High422 Profile and MPEG-2 422 Profile Encoder LSI for HDTV Broadcasting Infrastructures," IEICE Transactions on Electronics, Vol. E95-C, No. 4, pp. 432–440, Apr. 2012.

4. Yukikuni Nishida, Takayuki Onishi, Hiroe Iwasaki, Mitsuo Ikeda and Atsushi Shimizu, "8K Scalable Reference Picture Buffer Memory Architecture for HEVC Encoder LSIs," IEICE Transactions on Information and Systems (Japanese Edition) Vol. J99-D, No. 12, pp.1142-1153, Dec. 2016 (In Japanese).

5. Yuya Omori, Takayuki Onishi, Hiroe Iwasaki and Atsushi Shimizu, "A 120 fps High Frame Rate Real-time HEVC Video Encoder with Parallel Configuration Scalable to 4K", IEEE Transactions on Multi-Scale Computing Systems, Volume 4, Issue 4, pp. 491-499, Oct. 2018.

6. Daisuke Kobayashi, Ken Nakamura, Takayuki Onishi, Hiroe Iwasaki and Atsushi Shimizu, "A 4K/60p HEVC Real-Time Encoding System With High Quality HDR Color Representations", IEEE Transactions on Consumer Electronics, Volume 64, Issue 4, pp. 433-441, Nov. 2018.

Other refereed conference papers

1. Takayuki Onishi, Koyo Nitta, Hiroe Iwasaki, and Kazuto Kamimura, "SystemC-Based High-Level Synthesis of an AVC/H.264 Intra HDTV Encoder," *Design Automation Conference (DAC2010)*, No. 2U.26p, June 2010.

2. Jiro Naganuma, Hiroe Iwasaki, Koyo Nitta, Ken Nakamura, Takeshi Yoshitome, Mitsuo Ogura, Yayusuki Nakajima, Yutaka Tashiro, Takayuki Onishi, Mitsuo Ikeda, and Makoto Endo, "Single-chip MPEG-2 422P@HL CODEC LSI with Multi-chip

Configuration for Large Scale Processing beyond HDTV Level," in *Proceedings of Hot Chips: A Symposium on High Performance Chips (Hot Chips 14)*, Aug. 2002.

3. Hiroe Iwasaki, Jiro Naganuma, Koyo Nitta, Ken Nakamura, Takeshi Yoshitome, Mitsuo Ogura, Yasuyuki Nakajima, Yutaka Tashiro, Takayuki Onishi, Mitsuo Ikeda, and Makoto Endo, "Single-chip MPEG-2 422P@HL CODEC LSI with Multi-chip Configuration for Large Scale Processing beyond HDTV Level," In *Proceedings of Design, Automation and Test in Europe Conference 2003 (DATE'03)*, pp. 2-7 supple., Mar. 2003.

4. Hiroe Iwasaki, Jiro Naganuma, Yasuyuki Nakajima, Yutaka Tashiro, Ken Nakamura, Takeshi Yoshitome, Takayuki Onishi, Mitsuo Ikeda, Takaaki Izuoka and Makoto Endo, "A 1.1 W single-chip MPEG-2 HDTV codec LSI for embedding in consumer-oriented mobile codec systems," in *Proceedings of IEEE 2003 Custom Integrated Circuits Conference*, Sep. 2003.

5. Jiro Naganuma, Hiroe Iwasaki, Mitsuo Ikeda, Koyo Nitta, Ken Nakamura, TakeshiYoshitome, Mitsuo Ogura, Yasuyuki Nakajima, Yutaka Tashiro, Takayuki Onishi, Toshihiro Minami, Takaaki Izuoka, Makoto Endo, and Yoshiyuki Yashima, "VASA/ISIL: Single-chip MPEG-2 HDTV CODEC LSIs for Advanced Professional and Consumer Embedded Systems," in *Proceedings of IEEE Symposium on Low-Power and High-Speed Chips (COOL Chips VII)*, pp. 87-100, Apr., 2004.

6. Minoru Inamori, Hiroe Iwasaki, Takayuki Onishi, Mitsuo Ikeda, Jiro Naganuma and Yoshiyuki Yashima, "New set-top box for interactive visual communication of home entertainment using MPEG-2 full-duplex codec LSI," in *2005 Digest of Technical Papers. International Conference on Consumer Electronics (ICCE2005),* Jan. 2005.

7. Mitsuo Ikeda, Hiroe Iwasaki, Koyo Nitta, Takayuki Onishi, Takashi Sano, Atsushi Sagata, Yasuyuki Nakajima, Minoru Inamori, Takeshi Yoshitome, Hiroaki Matsuda, Ryuichi Tanida, Atsushi Shimizu, and Jiro Naganuma, "A professional H.264/AVC CODEC chip-set for HDTV broadcast infrastructure and high-end flexible CODEC systems," in *Proceedings of Hot Chips: A Symposium on High Performance Chips (Hot*

Chips 19), Aug. 2007.

8. Hiroe Iwasaki, Mitsuo Ikeda, Koyo Nitta, Takayuki Onishi, Takashi Sano, Atsushi Sagata, Yasuyuki Nakajima, Minoru Inamori, Takeshi Yoshitome, Hiroaki Matsuda, and Jiro Naganuma, "Professional H.264/AVC Decoder LSI for High-quality HDTV Broadcast Infrastructure," in *Proceedings of IEEE Symposium on Low-Power and High-Speed Chips (COOL Chips XI)*, pp. 287-293, Apr. 2008.

9. Koyo Nitta, Mitsuo Ikeda, Hiroe Iwasaki, Takayuki Onishi, Takashi Sano, Atsushi Sagata, Yasuyuki Nakajima, Minoru Inamori, Takeshi Yoshitome, Hiroaki Matsuda, Ryuichi Tanida, Atsushi Shimizu, Ken Nakamura, and Jiro Naganuma, "An H.264/AVC High422 Profile and MPEG-2 422 Profile Encoder LSI for HDTV Broadcasting Infrastructures," in *Proceedings of 2008 IEEE Symposium on VLSI Circuits (VLSI Circuits 2008)*, pp. 106-107, June 2008.

10. Mitsuo Ikeda, Takayuki Onishi, Takashi Sano, Atsushi Sagata, Hiroe Iwasaki, Yasuyuki Nakajima, Koyo Nitta, Yasuko Takahashi, Kazuya Yokohari, Daisuke Kobayashi, Kazuto Kamikura, and Hirohisa Jozawa, "MVC real-time video encoder for full-HDTV 3D video," in *Proceedings of IEEE International Conference on Consumer Electronics (ICCE2012)*, pp. 166-167, Jan. 2012.

11. Hiroe Iwasaki, Takayuki Onishi, Ken Nakamura, Koyo Nitta, Takashi Sano, Yukikuni Nishida, Kazuya Yokohari, Jia Su, Naoki Ono, Ritsu Kusaba, Atsushi Sagata, Mitsuo Ikeda and Atsushi Shimizu, "Professional H.265/HEVC encoder LSI toward high-quality 4K/8K broadcast infrastructure," in *Proceedings of Hot Chips: A Symposium on High Performance Chips (Hot Chips 27)*, Aug. 2015.

12. Yuya Omori, Takayuki Onishi, Hiroe Iwasaki and Atsushi Shimizu, "A 120 fps high frame rate real-time HEVC video encoder with parallel configuration scalable to 4K," in *Proceedings of 2017 IEEE Symposium in Low-Power and High-Speed Chips (COOL CHIPS)*, Apr. 2017.

13. Daisuke Kobayashi, Ken Nakamura, Takayuki Onishi, Yasuyuki Nakajima, Hiroe

Iwasaki, Mitsuo Ikeda and Atsushi Shimizu, “An HEVC real-time encoding system with high quality HDR color representations,” in *Proceedings of 2018 IEEE International Conference on Consumer Electronics (ICCE)*, Jan. 2018.

14. Ken Nakamura, Yuya Omori, Daisuke Kobayashi, Tatsuya Osawa, Takayuki Onishi, Koyo Nitta, Hiroe Iwasaki and Atsushi Shimizu, “Low Delay 4K 120fps HEVC Decoder with Parallel Processing Architecture,” in *Proceedings of 2019 IEEE Symposium in Low-Power and High-Speed Chips (COOL CHIPS)*, Apr. 2019.

15. Yasuhiro Mochida, Takayuki Nakachi, Takahiro Yamaguchi, Takayuki Onishi and Ken Nakamura, “An MMT Module for 4K/120fps Temporally Scalable Video,” in *Proceedings of 2019 IEEE International Symposium on Circuits and Systems (ISCAS 2019)*, May 2019.