

Motion Estimation and Compensation
Hardware Architecture
with Hierarchy of Flexibility
in Video Encoder LSIs

Koyo Nitta

Abstract

This dissertation devotes to investigate motion estimation and compensation (ME/MC) hardware architecture with hierarchy of flexibility in video encoder Large Scale Integrated circuits (LSIs) that realizes high image quality and high functionality. Through the development of three video encoder LSIs, the studies presented here discuss and optimize ME/MC hardware architecture from the perspective of flexibility.

Video coding technology has been widespread over various video applications. Most of these applications require real-time encoding. Many real-time video encoder LSIs, therefore, have already been developed. However, almost all of them focus on circuit area and power consumption, rather than image quality and functionality. Therefore, this dissertation aims at realization of video encoder LSIs that pursues image quality and functionality enough to be utilized even for professional use, such as broadcasting applications. This dissertation concentrates on ME/MC, and investigates to realize ME/MC hardware architecture with high image quality and high functional expandability by introducing a concept of hierarchy of flexibility. Hierarchy of flexibility contributes for ME/MC architecture to support a wide range of coding tools, to decide coding modes intelligently, and to expand ME/MC operations.

First, ME/MC hardware architecture with functional block level flexibility is proposed. By the proposed Flexible Communication Architecture, a scene-adaptive algorithm is realized. Experiments show that image quality is enhanced 1.2 dB in peak signal-to-noise ratio (PSNR). The ME/MC hardware architecture is implemented in a single-chip MPEG-2 4:2:2 Profile at Main Level (422P@ML) video encoder LSI.

Secondly, a Single Instruction stream Multiple Data streams (SIMD) macroblock processor is proposed. With instruction level flexibility of the SIMD, many ME/MC coding tools in MPEG-2, such as half-pel precision motion compensation (MC), bi-directional prediction MC, and field/frame adaptive MC, can

be easily and efficiently supported on its software. Owing to the flexibility of the SIMD, 4:2:2 encoding function can also be realized by rewriting SIMD's program. Moreover, improvement on the SIMD by optimizing hardware architecture is also proposed and discussed.

Finally, ME/MC hardware architecture with thread level flexibility is proposed for H.264/AVC High422 Profile video encoder LSI. The motion vector (MV) search operation is decomposed into a unit search, called thread. Then, the arrangement of the threads can realize new ME/MC coding tools, such as multiple reference MC. The thread level flexibility, in cooperation with the functional block level flexibility, also plays important roles on functional expandability, such as transcoding and two-pass encoding, which are required for broadcasting infrastructures. The video encoder LSI implementing the proposed ME/MC architecture realizes almost the same image quality as that of the reference software of H.264/AVC, Joint Model (JM).

Three levels of flexibility provide ME/MC hardware architecture with variety of programmability and scene-adaptivity. As a result, ME/MC hardware architecture with hierarchy of flexibility can realize high image quality and high functionality of video encoder LSIs.

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 Takashi Sato, for withstanding the enduring task of examining the draft, and pointing out countless mistakes. The advices you committed were invaluable and indispensable. I would also like to thank my committee members, Professor Hidetoshi Onodera, and Professor Naofumi Takagi, for serving as my committee members and making a lot of fruitful advices.

I would also like to thank Professor Shuzo Yajima, who decided me to make my career in Informatics. I am also very grateful to Professor Yasuhiko Takenaga now with the University of Electro-Communications, my first supervisor at Kyoto University, Professor Kiyoharu Hamaguchi with Shimane University, Professor Hiroyuki Ochi with Ritsumeikan University, Professor Kazuyoshi Takagi of Kyoto University, and all other members I studied with at Yajima laboratory. I was so lucky to be surrounded with excellent researchers at the very first time of my career.

I would also like to express my appreciation to managers at Nippon Telegraph and Telephone corporation (NTT) LSI laboratories, Dr. Ryota Kasai now at NTT Electronics, Prof. Takeshi Ogura now with Ritsumeikan University, Prof. Jiro Naganuma with Shikoku University, Prof. Toshio Kondo with Mie University, Prof. Takeshi Yoshitome with Tottori University, for guiding my research. I would also like to thank managers, supervisors, and colleagues at NTT Human Interface Labs, NTT Cyber Space Labs, and NTT Media Intelligence Labs, Dr. Masahiko Hase now CEO at NTT-IT, Prof. Yoshiyuki Yashima with Chiba Institute of Technology, Prof. Kazuto Kamikura with Tokyo Polytechnic University, Dr. Hiroto Inagaki, Mr. Mitsuo Ikeda, Dr. Hiroe Iwasaki, for their helpful support and valuable discussion. Your advices on both research as well as on my career were price-

less. A special thanks to my colleagues, Mr. Takuro Takahashi and Mr. Yasuhiko Sato 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 Mizue, more than words, my parents Tadao and Yoshiko, and my son and daughter Shoot and Naho, for sincere and never ending love, patience and support. Without your encouragement, I would not have been to accomplish the researches presented here.

Koyo Nitta
Kanagawa, February 2015.

Contents

1	Introduction	1
1.1	Background	1
1.2	Video encoder LSIs	3
1.3	The purpose of this dissertation	6
1.4	The overview of dissertation	7
2	Motion estimation and compensation	11
2.1	Fundamentals of ME/MC	11
2.2	ME/MC algorithms	12
2.2.1	Full search algorithm	12
2.2.2	Categorization of ME/MC algorithms	13
2.3	Extensions of ME/MC	16
3	Functional block level flexibility for a scene-adaptive algorithm	21
3.1	Introduction	21
3.2	Scene-adaptive algorithm	23
3.2.1	Hierarchical telescopic search	23
3.2.2	Scene-adaptive control	25
3.3	Hardware architecture	29
3.3.1	Flexible communication architecture	29
3.3.2	Search engine	31
3.4	Implementation results	33
3.4.1	Image quality evaluation	33
3.4.2	Chip implementation	35
3.5	Chapter summary	35
4	Instruction level flexibility — SIMD macroblock processor	41
4.1	Introduction	41

4.2	SIMD macroblock processor	43
4.3	Improvement on SIMD	45
4.3.1	Approaches for improving the SIMD performance	45
4.3.2	Addition of specific execution hardware	46
4.3.3	Improvement on I/O throughput of image data	49
4.3.4	Optimization of instruction set architecture	50
4.4	Evaluations	52
4.5	Implementation	54
4.6	Chapter summary	55
5	Thread level flexibility for H.264/AVC High422 Profile encoder LSI	59
5.1	Introduction	59
5.2	System Architecture	62
5.2.1	SARA architecture	62
5.2.2	HDTV configuration	65
5.3	ME/MC Architecture	65
5.3.1	ME/MC algorithm	65
5.3.2	Two-pel motion estimation architecture	67
5.4	Implementation	72
5.5	Evaluations	73
5.6	Chapter summary	76
6	Concluding remarks	79
	Bibliography	83
A	List of publications	91

List of Figures

1.1	Video coding framework.	3
1.2	The number of papers related to video encoder LSIs presented at ISSCC since 1993 to 2013.	4
1.3	The number of transistors in video encoder LSIs and Intel micro-processors.	5
1.4	Structure of this dissertation.	8
2.1	Motion estimation and compensation.	12
3.1	Hierarchical telescopic search.	24
3.2	Area hopping method.	26
3.3	Function of the forward/backward/interpolative decision.	28
3.4	Block diagram of the SuperENC.	29
3.5	Intra-chip communication model with the Flexible Communication Architecture.	30
3.6	Block diagram of the SE.	32
3.7	PSNR of “football” sequence.	33
3.8	The original 23rd frame in the “football” sequence.	34
3.9	The part of the 23rd frame in decoded images of “football” sequence.	37
3.10	Chip photograph of SuperENC.	38
4.1	Block diagram of the SuperENC-II.	43
4.2	SIMD macroblock processor architecture in SuperENC.	45
4.3	Double-issued instruction of the SIMD in the SuperENC.	46
4.4	Pixel interpolator.	47
4.5	Interpolated pixels stored in PE_i	48
4.6	Cycle-stealing architecture.	49
4.7	Example of write access by the SDRAM interface.	50
4.8	LIW instruction format issued by SIMD controller.	51

4.9	SIMD software examples.	52
4.10	Comparison of the numbers of dynamic steps.	53
4.11	Comparison of the numbers of dynamic steps for each operation (B-pictures).	54
4.12	Comparison of the rate of PE operations.	55
4.13	Comparison of the numbers of static steps.	56
4.14	Chip photograph of the SuperENC-II.	57
5.1	Block diagram of the SARA.	63
5.2	Advanced coding control scheme with pre-analysis engines. . . .	64
5.3	Memory mapping to reduce bandwidth when 4:2:2 encoding. . . .	64
5.4	HDTV configuration.	66
5.5	ME/MC algorithm used in SARA.	67
5.6	The TME architecture.	68
5.7	Two types of parallelism introduced in the PE array group. . . .	69
5.8	Approximation of predicted motion vector.	70
5.9	Instruction of the TME sequencer.	71
5.10	Examples of thread level flexibility.	72
5.11	Microphotograph of the SARA.	73
5.12	The SARA HD module.	74
5.13	Image quality comparison between SARA and JM.	74
5.14	Adaptive widening search area scheme.	75
5.15	Fade scene with or without automatic weighted prediction. . . .	76
5.16	HDTV H.264/AVC encoder equipment using the SARA chips. . .	77

List of Tables

1.1	The requirements that each level of flexibility will solve.	8
3.1	Specifications of the SuperENC.	38
3.2	Comparison between MPEG-2 encoder LSIs.	39
4.1	MC operations assigned to SIMD macroblock processor and their complexity (MOPS) in SuperENC-II.	44
4.2	Specifications of the SuperENC-II.	57
5.1	Specifications of the SARA.	78

List of Acronyms

3D	three dimensional
ALU	arithmetic logic unit
ARIB	Association of Radio Industries and Businesses
ASIC	application specific integrated circuit
ATSC-M/H	Advanced Television Systems Committee - Mobile/Handheld
BS	Broadcasting Satellite
BD	Blu-ray Disc
CATV	Community Antenna TV
CABAC	content adaptive binary arithmetic coding
CAVLC	content adaptive variable length coding
CS	Communications Satellite
CPU	Central Processing Unit
DCT	Discrete Cosine Transform
DDR-SDRAM	double data rate SDRAM
DVB-H	Digital Video Broadcasting - Handheld
DVD	Digital Versatile Disc
eDRAM	embedded DRAM
EPZS	Enhanced Predictive Zonal Search

FS	full search
GOP	group of pictures
HD	high definition
HDTV	high definition TV
IPTV	Internet Protocol TV
ISDB-T	Integrated Services Digital Broadcasting - Terrestrial
ISO/IEC	International Organization for Standardization / International Electrotechnical Commission
ISSCC	International Conference on Solid-State Circuits
ITE	the Institute of Image Information and Television Engineers
ITU-T	International Telecommunication Union - Telecommunication Standardization Sector
JM	Joint Model
LIW	long instruction word
LSI	Large Scale Integrated circuit
MB	macroblock
MBAFF	macroblock adaptive field/frame coding
ME	motion estimation
ME/MC	motion estimation and compensation
MC	motion compensation
MPEG	Moving Picture Experts Group
MPSoC	multiple-processor system-on-chip
MV	motion vector
NTT	Nippon Telegraph and Telephone corporation

OCP	Open Core Protocol
PAFF	picture adaptive field/frame coding
PC	personal computer
PE	processing element
PSNR	peak signal-to-noise ratio
RISC	Reduced Instruction Set Computing
SAD	sum of absolute difference
SIMD	Single Instruction stream Multiple Data streams
SDRAM	synchronous DRAM
SD	standard definition
SDTV	standard definition TV
TM	Test Model
TV	television
VCEG	Video Coding Experts Group
VLC	Variable Length Coding
VLIW	very long instruction word
VoD	Video on Demand

Chapter 1

Introduction

1.1 Background

In March 2012, analog terrestrial television (TV) broadcasting was finished in Japan. Since then, the transition to digital terrestrial TV broadcasting has been completed. Besides terrestrial broadcasting, digital video broadcasting, which utilizes video coding technology, has been spread out over satellite broadcasting, such as Broadcasting Satellite (BS) and Communications Satellite (CS) services, Internet Protocol TV (IPTV), Community Antenna TV (CATV), and broadcasting services for mobile devices. In addition to broadcasting, video coding technology is also used in various video applications such as Video on Demand (VoD), Internet video streaming, video communications (video conference and video telephony), video surveillance, telemedicine, and so forth.

Digital video data, in general, are enormous. For example, standard definition TV (SDTV), which is supposed to be 720×480 pixel per frame, 60 field¹ per second, and 4:2:0 chroma format, has about 125 mega bit per second (Mbps) pixel data. In the case of high definition TV (HDTV), assumed that is $1,920 \times 1,080$, 60 frame per second, and 4:2:0 chroma format, they are no less than 1.5 giga bps (Gbps). A two-hour high definition (HD) movie amounts to over 10 tera bit (Tbit). Thus, digital video data are too expensive to transmit through or to store into any medium as they are. Video coding technology, therefore, is indispensable in order to utilize transmission bandwidth or storage capacity efficiently. In digital video transmission, digital video data are encoded and compressed at the transmitter side, and are decoded and decompressed at the receiver side. Digital video stor-

¹In the interlaced video format, two complementary fields comprise one video frame.

age media, such as Digital Versatile Discs (DVDs) or Blu-ray Discs (BDs), store encoded video data, which are decoded by video players. Interoperability between video encoders and decoders encourages the growth of video application markets. Independence from transmission and storage media enlarges a range of video applications. These interoperability and independency are main reasons why video coding technology should be standardized.

Several video coding standards have already been made; H.261[23] in 1988, MPEG-1[16] in 1993, MPEG-2 (a.k.a. H.262)[18] in 1995, H.263[24] in 1996, MPEG-4[17] in 1999, and H.264/MPEG-4 AVC[19] in 2003. The latest video coding standard, HEVC[20], has just been standardized by a joint collaborative team of International Organization for Standardization / International Electrotechnical Commission (ISO/IEC) Moving Picture Experts Group (MPEG) and International Telecommunication Union - Telecommunication Standardization Sector (ITU-T) Video Coding Experts Group (VCEG). All of these standards have a common framework, so called “*MC+DCT*,” that is based on motion compensation (MC) and Discrete Cosine Transform (DCT), as well as quantization and entropy coding. Digital video data have a lot of redundancy. Video encoding can eliminate the redundancy and compress video in a lossy manner. The MC, incorporated with motion estimation (ME), can reduce temporal redundancy, and the DCT and quantization can reduce spatial redundancy. The entropy coding can also reduce data of intermediate coded words. Among them, motion estimation and compensation (ME/MC) is one of the operations with the highest computational complexity in video encoding.

Real-time video encoding is necessary for the spread of video applications. However, the aforementioned video coding standards consist of the state-of-the-art technologies at those times when they were standardized, and require much computational loads so that the cutting edge Central Processing Units (CPUs) in those days could not process video encoding in real time. Hence, video encoding has been implemented on Large Scale Integrated circuits (LSIs) as application specific integrated circuits (ASICs). In a video encoder LSI, due to its computational complexity, the ME/MC has great impact on both circuit area and memory bandwidth. The computational load reduction of the ME/MC, while it can lead to the reduction of the area and the bandwidth, is inevitable to degrade decoded image quality. Thus, in order to realize a video encoder LSI with high image quality, how to implement the ME/MC is one of the major issues.

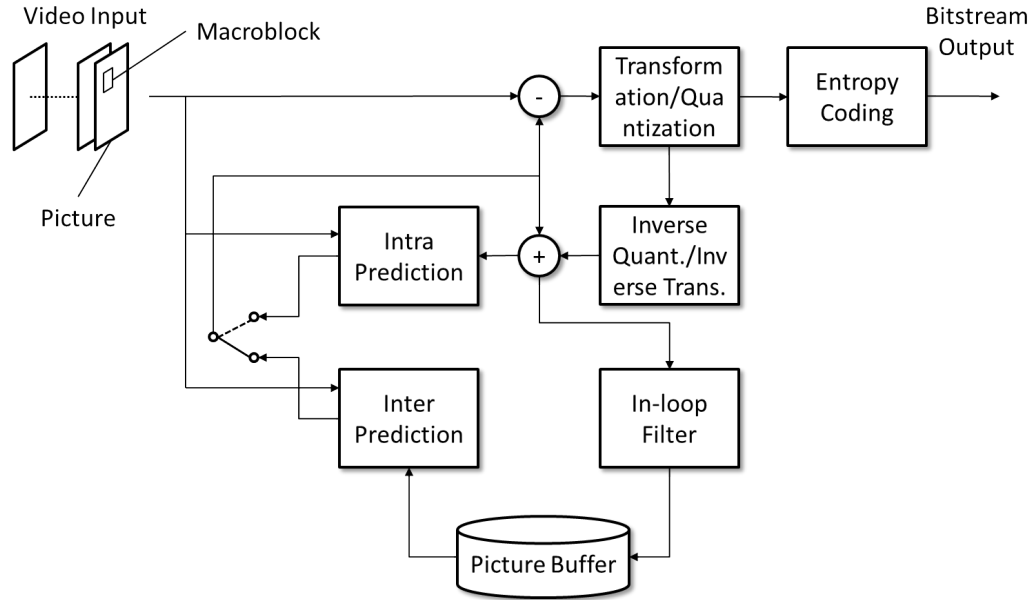


Figure 1.1: Video coding framework.

1.2 Video encoder LSIs

Video coding framework based on MC+DCT is described in Figure 1.1. Video data are input to an encoder as a series of frames or fields, called *pictures*. Each picture is divided into small regions, called *macroblocks*. A macroblock (MB), a unit of coding process, typically consists of an array of 16×16 pixels of luminance and two arrays of 8×8 pixels of chrominance. A MB is predicted with intra prediction (prediction within a picture) or inter prediction (prediction between pictures). In inter prediction, the ME/MC is performed. The prediction errors, differences between the macroblock and the predicted samples, are transformed from spatial domain to frequency domain. Quantization is processed to each coefficient of the frequency domain. The quantized coefficients, together with the selected prediction mode, are encoded with entropy coding, and are output as a bitstream. Local decoded images are made by inverse quantization, inverse transformation, and addition to the predicted samples. They are used by the intra prediction, and also stored into picture buffers after in-loop filter in order for the ME/MC to use as reference pictures.

The video coding framework described above is implemented on video encoder LSIs. Many video encoder LSIs have already been proposed. Figure 1.2 shows the number of papers related to video encoder LSIs presented at Interna-

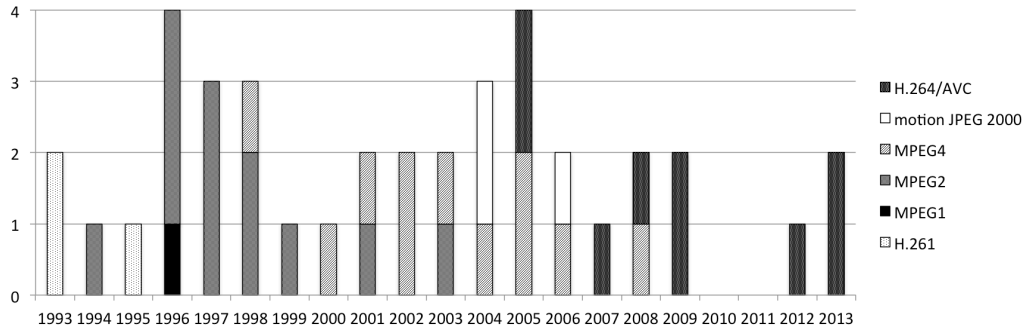


Figure 1.2: The number of papers related to video encoder LSIs presented at ISSCC since 1993 to 2013.

tional Conference on Solid-State Circuits (ISSCC), one of the greatest authoritative international conferences in semiconductor area, since 1993 to 2013. Note that all of these LSIs, encoder or codec (encoder and decoder), have the whole video encoding function with the chip or chipset. The papers related to video decoder LSIs or only to partial implementation of video encoding are not counted. In the case that it can support multiple video coding standards, an encoder LSI is regarded as supporting the latest standard. It is found that some papers about video encoder LSIs were presented almost every year. As a matter of course, the number of papers tends to increase for a few years just after a video coding standard was settled. Video encoder LSIs for major standards, such as MPEG-2, MPEG-4, or H.264/AVC, had been reported for about a decade.

The author, as a part of Nippon Telegraph and Telephone corporation (NTT), also developed several video encoder LSIs. In 1995, an MPEG-2 video encoder chipset, ENC-C and ENC-M[45, 29, 14], was developed for video communication. In the chipset, ME/MC could only support forward prediction (see Sec. 2.3). SuperENC[31, 37, 11, 12, 13, 38], which is a single-chip MPEG-2 video encoder LSI fabricated in 1997, was aimed mainly at consumer video encoding devices, such as personal computer (PC) card[6]. The ME/MC architecture could support bi-directional prediction, as well as field/frame adaptive prediction and half-pel prediction. Because the SuperENC could support 4:2:2 chroma format, as well as conventional 4:2:0 chroma format, and could also support HD formats with multiple-chip configuration[46, 50], it opened up professional use rather than consumer one. In order to enhance image quality of SuperENC enough for professional use, SuperENC-II[39] was developed in 2000. The ME/MC also had to be enhanced. Moreover, for the digital terrestrial TV broadcasting, the

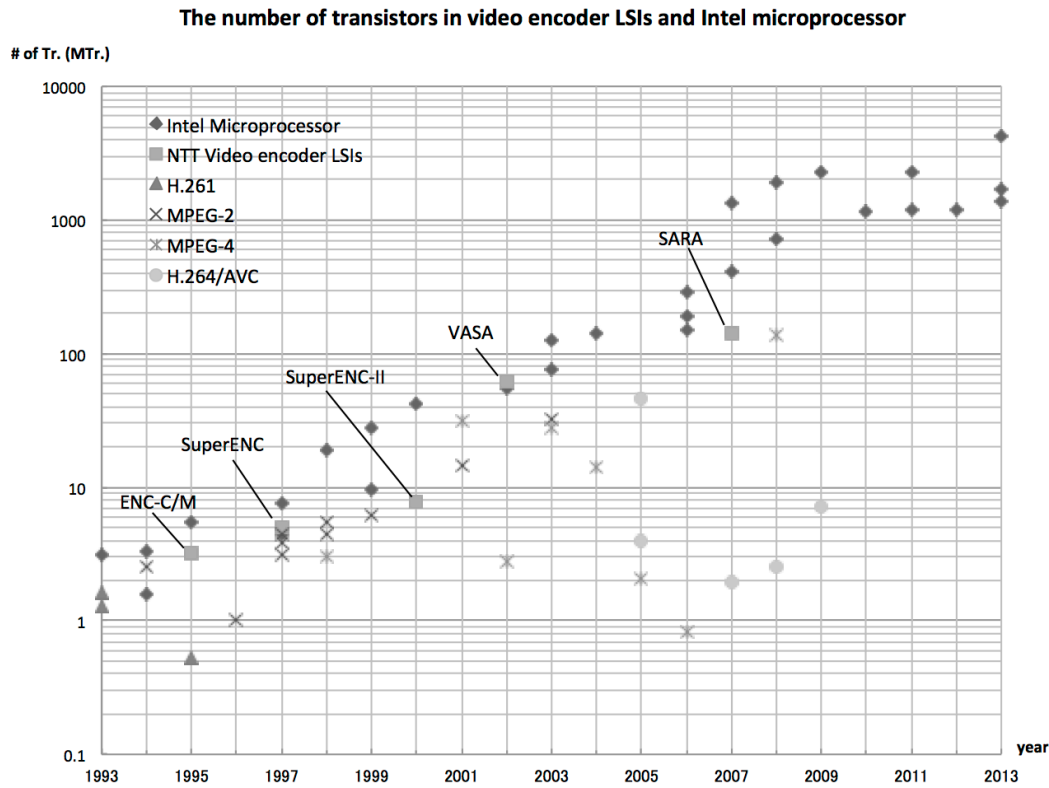


Figure 1.3: The number of transistors in video encoder LSIs and Intel microprocessors.

world-first HDTV MPEG-2 encoder chip, VASA[26, 27], was made for digital TV broadcasting contribution networks in 2002. After H.264/AVC was standardized, broadcasting equipments shifted from MPEG-2 to H.264/AVC steadily. H.264/AVC has also been used in IPTV services. These are the reasons why SARA[10, 35, 36], H.264/AVC and MPEG-2 video encoder LSI, was developed in 2007. The ME/MC in the SARA has to support several coding tools newly introduced in H.264/AVC, as well as ones in MPEG-2. It also has to provide functionality for transcoding, two-pass coding, and low-delay coding.

Figure 1.3 shows the number of transistors in NTT's video encoder LSIs and in other encoder LSIs presented in ISSCC, together with Intel microprocessors for reference. It is found that NTT's video encoder LSIs contain almost as many transistors as Intel microprocessors one generation ago, and that a tendency on circuit scale of NTT's LSIs is quite different from the trend of other video encoder LSIs, because the former aims at professional use while the latter mainly targets at consumer and mobile use.

1.3 The purpose of this dissertation

The main goal of this dissertation is to prove the feasibility of ME/MC hardware architecture that realizes video encoder LSIs with high image quality and high functionality, as described below. High image quality enhances user experiences of viewers, and high functionality expands the application fields of the LSIs and could even create brand-new video services.

Video coding standards can be regarded as a set of coding tools, which are variations or extensions of coding technologies such as inter/intra prediction. Each of the coding tools defines several coding modes to be selected in encoding. Meanwhile, video coding standards only define syntax of bitstreams and provide the way how to decode them. It, therefore, depends on encoder designers how to encode video data, that is, what coding tools are used and how to select coding modes. All what a video encoder LSI has to do is to output bitstreams in conformance with a standard. As a result, the quality of decoded images varies with video encoder LSIs. To achieve high image quality, video encoder LSIs should support as many coding tools specified in a standard as possible, and should decide the most efficient coding modes.

Some video applications require specific functions of encoding. For contributions, which is an application to transmit video materials between broadcasting key stations, a richer color format than usual, called 4:2:2 chroma format, is often used. For distributions, which is an application to distribute video contents from broadcasting key stations to end-user's terminals, two-pass encoding is essential for high compression of video data. In the two-pass encoding, the first video encoder LSI should output encoding information, and the second LSI should receive it to encode with higher compression rate. Low-delay (latency) encoding is indispensable for live broadcasting. In the case of low-delay encoding, a special encoding technique, called *intra refresh*, is usually used. Transcoding, which decodes a bitstream and encodes into another bitstream, is needed in order to bridge between applications. A simple transcoding tends to yield to a degradation of image quality. The video encoder LSIs, therefore, should receive some information of decoding, and should refer to it for encoding. For three dimensional (3D) video services, stereo encoding is indispensable. Thus, it is desirable that video encoder LSIs have high functionality beyond the ordinary encoding.

This dissertation concentrates mainly on the ME/MC architecture, which has great influence on a video encoder LSI. Although many ME/MC hardware architectures have already been proposed, as illustrated in Figure 1.3, most of the

previous studies on ME/MC hardware architecture were focused on circuit area or power dissipation, rather than image quality or functionality. Of course, circuit area and power dissipation are important issues, especially when the target of an encoder LSI is on consumer electronics devices. However, when it comes to professional devices, such as ones used in broadcasting applications, image quality and functional expandability take precedence over area or power. Then, the researches presented here investigate how to achieve high image quality and high functionality. In order to fulfill high image quality and high functionality, the main requirements for the ME/MC architecture are as follows:

1. Wide-ranging support for various ME/MC coding tools, especially tools which are indispensable for broadcasting application, in a standard,
2. Intelligent MC mode decision with adaptability for input video, and
3. Functional expandability of ME/MC beyond the ordinary encoding.

As for an approach to these requirements, a concept of “hierarchy of flexibility” is introduced on ME/MC architecture. The aim of this dissertation is to reveal ME/MC hardware architecture with hierarchy of flexibility, that enables a video encoder LSI to attain full potential of image quality of video coding standards, and even to open up a brand-new services with the functional expandability.

1.4 The overview of dissertation

The studies presented here view and optimize the ME/MC architectures from the perspective of “flexibility.” Hierarchy of flexibility of the ME/MC architectures is considered;

- Functional block level flexibility (Chapter 3),
- Thread level flexibility (Chapter 5),
- Instruction level flexibility (Chapter 4).

Figure 1.4 depicts a schematic structure of this dissertation, on how the requirements for the ME/MC architecture mentioned above are approached. Table 1.1 shows which requirements each level of flexibility will solve.

First, the ME/MC is reviewed in Chapter 2. Then, three of the video encoder LSIs described in the previous section, SuperENC, SuperENC-II, and SARA, are

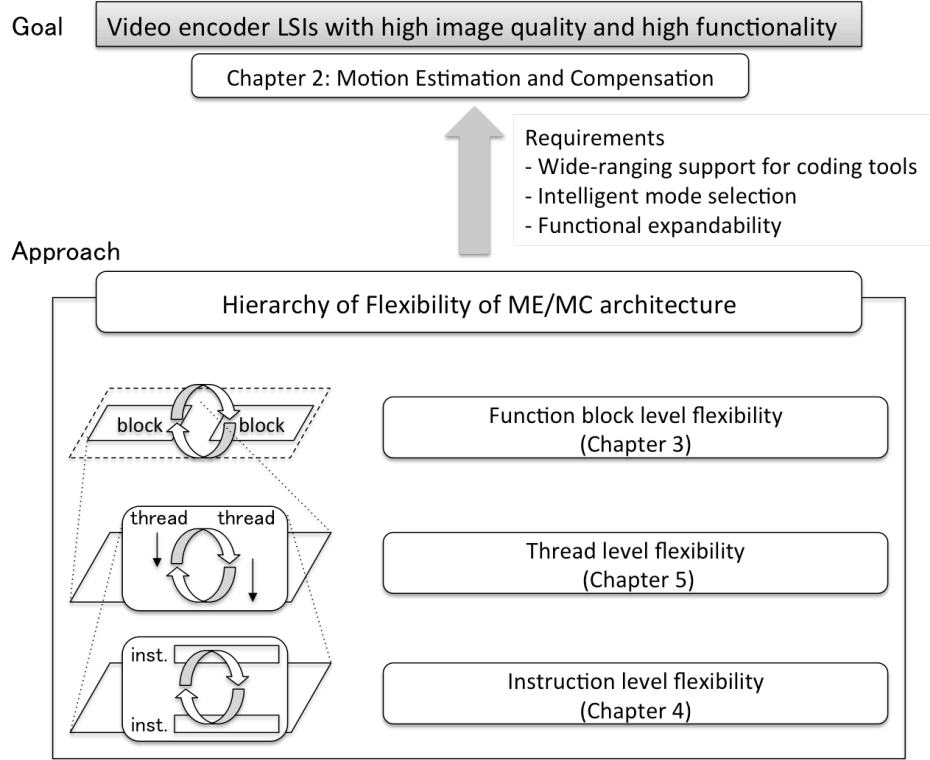


Figure 1.4: Structure of this dissertation.

discussed from the viewpoint of different levels of flexibility in the chronological order by Chapter 3, Chapter 4, and Chapter 5, respectively.

In the next Chapter, a fundamental concept of the ME/MC is introduced. Many ME/MC algorithms have already been proposed. In this dissertation, instead of reviewing them, they are categorized from the point of view of how to reduce computational complexity. Then, extensions of the ME/MC, which requires flexibility of the ME/MC architectures, in various standards are reviewed.

Table 1.1: The requirements that each level of flexibility will solve.

Requirements	Functional block level	Thread level	Instruction level
Wide-ranging support for coding tools		✓	✓
Intelligent mode decision	✓		
Functional expandability		✓	✓

In Chapter 3, the ME/MC architecture of the MPEG-2 4:2:2 Profile at Main Level (422P@ML) video encoder LSI, SuperENC, is described. To enhance image quality, scene adaptivity is proposed. To support the proposed scene adaptive algorithms, which can adjust ME/MC parameters during encoding, the ME/MC architecture is designed with flexibility of the functional block level, which is called “*Flexible Communication Architecture*.” In the proposed ME/MC architecture, a time interval of any length between functional blocks can be inserted. Through subjective and objective evaluations, the enhancement of image quality is shown.

The flexibility of hardware architectures is always on trade-off relation with the computational performance. In Chapter 4, a Single Instruction stream Multiple Data streams (SIMD) macroblock processor with instruction level flexibility, is proposed and optimized. In the proposed SIMD architecture, various coding tools, such as half-pel precision MC, bi-directional prediction MC, and field/frame adaptive prediction MC, can be supported by programmability of the SIMD. Moreover, the performance of the SIMD macroblock processor after optimization is enhanced to about 1.5 times as compared with the one before optimization, without sacrificing the instruction level flexibility. Thus, the SIMD can support various ME/MC coding tools efficiently.

In Chapter 5, the focus moves on to a new video coding standard, H.264/AVC. In order to support newly introduced coding tools, the ME/MC architecture with thread level flexibility is proposed. Through the thread level flexibility, the proposed architecture can support a generalized bi-prediction MC and multiple reference MC. The architecture also has functional block level flexibility for weighted prediction, and instruction level flexibility for variable block size MC. The chapter discusses how the architecture satisfies the requirement of the professional encoder. Then, subjective evaluation results are also presented to show the validity of the architecture.

Chapter 6 summarizes the results of the researches, and concludes the dissertation by how the flexibility of the ME/MC architecture realizes video encoder LSIs with high image quality and high functionality. The future works are also discussed.

Chapter 2

Motion estimation and compensation

Before going to the details of the dissertation, motion estimation and compensation (ME/MC), which is the primary target of this dissertation among video encoding operations, is reviewed from various aspects; concept, formulation, algorithm, and extensions.

2.1 Fundamentals of ME/MC

Video data, in general, have much temporal redundancy. Successive pictures are highly correlated with one another. For example, there are almost no moves in the background of video conference. The same is true for almost all video data shot by a fixed camera. In these cases, the amount of video data can be greatly compressed by encoding only differences between pictures. Such a kind of encoding is called *interframe prediction*. However, video data with a lot of motions, such as camera panning, could not be compressed only by the interframe prediction. The compression efficiency of the interframe prediction can be enhanced more by *motion compensation (MC)*. When a macroblock (MB) of the input picture is encoded, the portion in a certain search range of an already encoded picture (called *reference picture*), where pixel values are the closest to ones of the MB, is searched. Then, the spatial displacement from the MB to the portion and the differences of pixel values between the MB and the portion are encoded. Here, the displacement is called *motion vector (MV)*, and the differences are called *prediction errors*. The search of the closest portion to the MB is called *motion estimation*

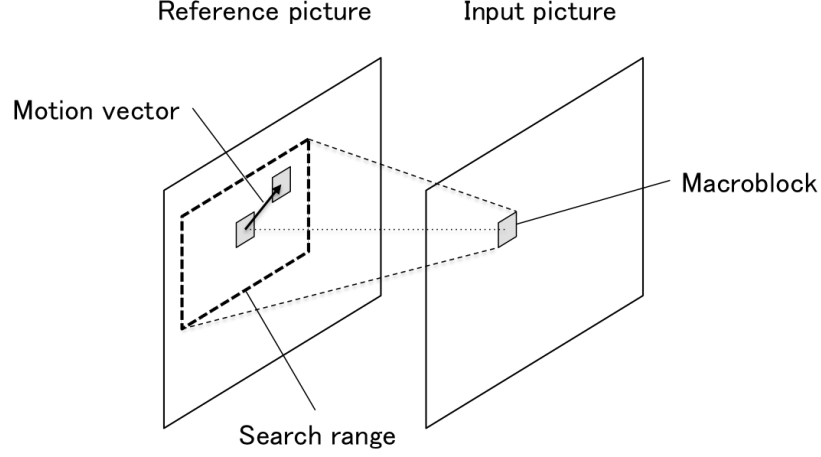


Figure 2.1: Motion estimation and compensation.

(ME) (Fig. 2.1). When encoding video data using the MC, the ME is also needed. In this dissertation, these two operations are regarded to be united, and called ME/MC.

Let $p^{(t)}(x, y)$ be a pixel value at the position (x, y) of the input picture at time t . A MB which consists of 16×16 pixels and whose left-above corner is at (x_0, y_0) can be expressed as $p^{(t)}(x_0 + i, y_0 + j)$ ($0 \leq i, j < 16$). The ME is the operation that finds a MV $mv = (v_x, v_y)$ such that the portion $p^{(t-\Delta t)}(x_0 + i + v_x, y_0 + j + v_y)$ of a search range in a reference picture at time $(t - \Delta t)$ is closest to the MB. The MC is the operation that expresses the MB with the MV $mv = (v_x, v_y)$ and the differences $d_{mv}^{(t)}(x_0 + i, y_0 + j)$ ($0 \leq i, j < 16$):

$$d_{mv}^{(t)}(x_0 + i, y_0 + j) = p^{(t)}(x_0 + i, y_0 + j) - p^{(t-\Delta t)}(x_0 + i + v_x, y_0 + j + v_y). \quad (2.1)$$

Note that a MB consists of a luminance array and two chrominance arrays. The ME usually uses only a luminance array because the human vision system is more sensitive to luminance than chrominance. The MC is processed both on a luminance array and on two chrominance arrays.

2.2 ME/MC algorithms

2.2.1 Full search algorithm

The ME/MC is one of the most computationally intensive operations in video encoding. Since the MC must be processed according to a video coding standard,

there is little room to reduce a computational load in it. On the other hand, the ME itself is not specified by a video coding standard. Thus, it is left to an encoder how to implement the ME. The ME, which has much computational complexity, plays an important role in video encoding, because it affects coding efficiency very much. This is the reason why many ME algorithms, which can reduce the computational load and which try to keep the decoded image quality at the same time, have already proposed.

ME algorithms are based on block matching. Any ME algorithms can be regarded as simplification of the basic algorithm, called *full search (FS)* or *exhaustive search*. In the FS algorithm, all MVs in the search range are evaluated. As an evaluation function of the MVs, the sum of absolute difference (SAD) is usually used. Given a MV mv , SAD_{mv} , which is the SAD of mv , can be expressed using the difference $d_{mv}(x_0 + i, y_0 + j)$ in Eq. (2.1) as follows;

$$SAD_{mv} = \sum_i \sum_j |d_{mv}(x_0 + i, y_0 + j)|. \quad (2.2)$$

Let R be a search range which is a set of the MVs. The FS is the operation that finds mv_{FS} , the MV with the minimum SAD in the range R ;

$$mv_{FS} = \{mv \text{ s.t. } SAD_{mv} = \min_{mv \in R} SAD_{mv}\} \quad (2.3)$$

The FS requires huge computational load. For example, it requires more than 260 giga operations per second (GOPs) in the case that the search range is ± 32 horizontally and vertically in the standard definition (SD) video, which consists of 720×480 pixel per frame, and 60 field per second. It reaches about 12.5 tera operations per second (TOPS) in the case of the search range is ± 64 horizontally and vertically in the high definition (HD) video, $1,920 \times 1,080$ pixel per frame, and 60 frame per second. Therefore, a ME algorithm which can drastically reduce the computational load of the FS is required in order to be implemented on a video encoder LSI.

2.2.2 Categorization of ME/MC algorithms

Many ME/MC algorithms have already been proposed. Instead of reviewing each of them, categorization of them is presented here. They can be classified into three categories according to the ways of the reduction of the computational load of the FS.

Category 1: The reduction of the number of MV evaluations

The first category, related to Eq. (2.3), is the most effective among the three. Therefore, most of the ME algorithms already proposed fall into this category. As for the FS, computational complexity is in proportion to the number of the MVs in the search range. For example, when the search range is ± 32 horizontally and vertically, the number of MV evaluations for a MB becomes $(32 \times 2 + 1)^2 = 4,225$ times. With the search range of ± 64 horizontally and vertically, it becomes 16,641. It is true that making the search range smaller can reduce the computational load, but it also leads to the decrease of the coding efficiency. Thus, it is required to reduce the number of MV evaluations while keeping the search range wide enough.

There are many algorithms proposed that can reduce the number of MV evaluations. One of the simplest ways is a step search, which searches the MV stepwise as described below. Let R be the search range with $\pm a$ horizontally and vertically. It evaluates such eight MVs as each component is $-a/2$, 0, or $a/2$ around the center $(0, 0)$, as well as the center of the search, in the first step. In the second step, it evaluates such eight MVs as each component has displacement of $-a/4$, 0, or $a/4$ from the center of the search, which is replaced from $(0, 0)$ to the best MV found in the first step. These steps are repeated for $\log_2 a$ times, and the best MV in the search range R is found. When $a = 32$, the number of MV evaluations is reduced to $1 + 8 \log_2 32 = 41$, about a hundredth compared with the FS. Thus, the step search, just like the “binary search,” can reduce the number of MV evaluations. However, it is not used for hardware implementation, because the degradation of the image quality cannot be neglected.

Another simple way is to evaluate MVs every some points in the search range, instead of every one point like the FS. When the search is operated every b points horizontally and vertically, the number of MV evaluations is $1/b^2$ of the FS. The larger the value b is, the less the number of MV evaluations is. However, there is trade-off between the reduction of the number of evaluations by the value b and the image quality. It is also noted that the search with every b points can only find the MV in the b -pixel precision, thus, an additional search in the $(2b + 1)^2$ area is needed to get the one-pixel precision MV.

One-dimensional search, at first, evaluates horizontal MVs, then vertical ones with the best horizontal MV as the center of the search, or vice versa. The number of MV evaluations is $2(2a + 1)$ when R is $(2a + 1) \times (2a + 1)$.

Telescopic search is on the assumption that motion of objects in video se-

quences can be regarded as uniform during very short periods. Given a current picture and a reference picture, which are apart of the distance d in the display order, telescopic search first operates the FS with the small search range $\pm a/d$ on the picture adjacent to the current picture in the display order, instead of the reference picture. Next, with the best MV found in the intermediate picture as the center, it searches the MV in the same way on the next to the intermediate picture in the display order. This is repeated with d times to the reference picture. The number of MV evaluations is $1/d$ of the FS.

While all of these algorithms described above can search all MVs in the search range uniformly, another approach utilizes the tendency that a MV is likely to be close to the near MVs spatially or temporally. Some vector, instead of $(0,0)$, is used as the center of the search, and the MVs in the small search range $R' \subset R$ are evaluated. For the center vector, called *shift vector* or *hopping vector*, a predicted MV (pmv), a MV among the ones of the blocks adjacent spatially or temporally, a vector calculated from other MVs statistically, or a global MV is used. The Enhanced Predictive Zonal Search (EPZS)[47] implemented in the reference software HM of the latest video coding standard HEVC also utilizes this method. Note that this method always needs a good shift vector in order to compress video efficiently, therefore, other uniform search schemes have to be used together.

Category 2: The reduction of MB calculations

The second category is involved with Eq. (2.2). The MB calculations consist of a summation of $16^2 = 256$ terms. In order to reduce the summation, for example, the vertical index j is counted up only using even numbers. This can reduce the number of the MB calculations by half. The HM, the reference software of HEVC, also adopts this technique in the uniform search. In this scheme, the pixel values in the omitted terms from the summation are never considered. Thus, it should not be used when there is a relatively low correlation between the line j and $j+1$, such as an interlaced format that the present TV systems usually employ.

A hierarchical search can also reduce MB calculations. It searches the MV on the down-sampled pictures. Using 4-to-1 down-sampled pictures, MB calculations can be reduced to $1/4$. Note that it can reduce the MV evaluations to $1/4$ at the same time because the search range R is also shrunk to $R/4$ in the 4-to-1 down-sampled pictures. Thus, an s -to-1 down-sampled hierarchical search can reduce the computational load to $1/s^2$ compared with the FS. This is one of the

reasons why the hierarchical search is often used in the ME/MC implementation, in spite of increase of additional down-sampling calculations.

Category 3: The reduction of pixel calculations

In the third category, the computational complexity of Eq. (2.1) is reduced by decreasing the bit length of the pixel value p . A one-bit search has proposed where the search is operated with one-bit values on edge-detected pictures by the Sobel filter. An eight-bit search is often used even when pixels are 10-bit length or more, since some SIMD instructions in recent CPUs have a restriction of the bit length.

Using any ME algorithm in the three categories, degradation of image quality is inevitable. Although, on software implementation, the EPZS in the HM can reduce computational load drastically and achieve almost the same image quality as the FS, the reduction of the EPZS is mainly owing to truncating the MV evaluations by some threshold. This is a software approach for off-line encoding and is not suitable for hardware implementation for real time encoding, because reduction of complexity depends on inputs, and it may be not able to reduce complexity at all in the worst case. On hardware implementation, where the worst case must always be taken into consideration, no ME algorithm and architecture have yet found that can reduce complexity drastically and can achieve almost the same image quality as the FS.

2.3 Extensions of ME/MC

The ME/MC has been extended according as new video coding standards are standardized¹. Several extensions involved in this dissertation are listed below.

Sub-pixel precision MC

In sub-pixel (sub-pel) precision MC, each component of the MV may not be integer, but can be fractional. The sample between pixels is generated by interpolating neighboring pixels. The sub-pixel precision MC can predict the move of the MB

¹Strictly speaking, though a video coding standard only extends the MC, encoding needs to extend the ME corresponded to the MC.

with higher precision. It also provides an effect of in-loop filtering. First, half-pel precision MC was adopted in the MPEG-1[16], and the successive MPEG-2[18] employed it as well. A sample at the half-pel position is generated by a simple bi-linear interpolation in these standards. Quarter-pel precision MC has been introduced since the H.264/AVC[19]. The latest video coding standard, HEVC[20], also adopts quarter-pel precision MC. A sample at the sub-pel position is calculated by a six-tap filter in the H.264/AVC, and an eight-tap filter in the HEVC, respectively.

Bi-prediction MC

Before the MPEG-2, only a single reference picture is allowed for the ME/MC. The reference picture precedes to the input picture in the display order. This type of the prediction in the ME/MC is called *forward prediction*. By reordering input pictures to encode, the MPEG-2 has introduced *backward prediction*, which uses a future picture in the display order as a reference picture. Moreover, *bi-directional prediction*, which uses two reference pictures for both forward and backward predictions, is also applied. Bi-directional prediction can make the prediction errors smaller by averaging the prediction samples instead of allowing to encode two MVs.

Bi-directional prediction is generalized to *bi-prediction* in the H.264/AVC and after, which no longer has a concept of direction of the prediction such as forward or backward. Two reference pictures of bi-prediction can be selected among already encoded pictures by an encoder.

Field/Frame adaptive MC

Most of the present TV systems usually adopt an interlaced video format. In the interlaced format, a frame is divided into two fields, called *top field* and *bottom field*, one of which consists of even lines of the frame and the other consists of odd lines of the frame. These two complementary fields are captured at different times. Interlaced video is a technique that makes temporal resolution double without increasing data.

MPEG-2 supports several coding tools for the interlaced video format. One of them is field/frame adaptive MC, which can regard a MB as a frame MB or a pair of field MBs adaptively. Each of the field MBs can have its own MV. H.264/AVC has two sorts of field/frame adaptive MC, one is picture adaptive

field/frame coding (PAFF) MC, and the other is macroblock adaptive field/frame coding (MBAFF) MC.

Unrestricted MV

Unrestricted MV mode allows such a MV that points to a portion outside the actual region of the reference picture. Border pixels are copied and used as the samples outside the picture. It can enhance prediction precision of the MBs near the borders of the input picture. It is first adopted in the H.263[24], and the following standards also employ the unrestricted MV.

Multiple reference MC

Since the H.264/AVC, the reference picture can be selected for each MB from a reference picture list. The list is composed of the pictures already encoded and marked as ‘used as a reference’. This is called multiple reference MC. It can extend the MV search range for a MB to the temporal axis. It can enhance coding efficiency on such cases as occlusions, camera flashing, and so forth. In combination with bi-prediction MC described above, two reference picture lists are prepared, from each of which one reference picture can be selected at most.

Variable block size MC

A MB may contain several regions which have different motions. Relatively newer standards such as the H.263, the MPEG-4, and the H.264/AVC allow an encoder, if needed, to divide the MB and select the size of the block. Each of the blocks that the MB is divided to can have its own MVs so as to represent such motions. Larger blocks can reduce the number of bits needed to represent MVs, while smaller blocks can reduce prediction errors to be encoded. In the H.264/AVC, seven kinds of blocks can be used; 4×4 , 4×8 , 8×4 , 8×8 , 8×16 , 16×8 , and 16×16 . Among them, 8×8 blocks and larger are particularly effective in encoding video formats larger than standard definition (SD).

Variable block size MC can be regarded as an efficiency-oriented approximated approach to an object-based video coding. Though it has been standardized in MPEG-4, an object-based video coding could not be spread, because it is very difficult to extract semantic objects automatically from a two-dimensional picture. Although it does not always lead to semantic object segmentation, variable block

size MC has an effect similar to an object-based coding in the sense of coding efficiency.

Weighted prediction

Since it searches the MV using similarity of luminance, the ME can find a portion similar to the MB only when the lighting condition is stable. Otherwise, such as fade scenes, the ME cannot work well. To overcome this drawback, the weighted prediction has been introduced in the H.264/AVC. In the weighted prediction, an encoder can utilize a reference picture with a scaling and an offset. Given a pixel value $p(x, y)$, a scaling factor $scale$, and an offset $offset$, the following value $p'(x, y)$ is used as a reference in the weighted prediction;

$$p'(x, y) = scale \times p(x, y) + offset. \quad (2.4)$$

Because all of these extensions of the ME/MC contribute to coding efficiency, video encoder LSIs should support as many extensions as possible in order to enhance image quality. Nevertheless, there are few encoder LSIs that can support most of these extensions, because they lead to area or power overhead. Most of ME/MC hardware architecture already proposed have given priority to circuit area and power consumption, than image quality. For example, H.264/AVC encoder LSIs early presented in ISSCC [9, 8, 3] support only baseline profile. Thus, they support neither bi-prediction MC nor field/frame adaptive MC (PAFF and MBAFF). Moreover, as far as the author knows, no encoder LSIs can support weighted prediction. Video coding standards usually have their own reference software implementations for the purpose of demonstrating their coding efficiency; Test Model (TM) of MPEG-2 or Joint Model (JM) of H.264/AVC. A reference software, then, acts as one of the indices of the image quality that a standard can achieve. Therefore, it is worth both academically and industrially proving the feasibility of ME/MC hardware architecture that realizes almost the same image quality of the reference software of the video coding standard. Moreover, it is also an issue to be solved how ME/MC architecture can meet the requirements for functional expandability, such as 4:2:2 encoding, low-delay encoding, two-pass encoding, transcoding, and stereo encoding, as well as image quality. From the next chapter on, ME/MC hardware architecture with high image quality and high functionality will be investigated.

Chapter 3

Functional block level flexibility for a scene-adaptive algorithm

3.1 Introduction

Among the proposed hierarchy of flexibility of motion estimation and compensation (ME/MC) hardware architecture, functional block level flexibility is first considered in order to realize intelligent mode selection with adaptability of input video. This chapter discusses ME/MC hardware architecture in a single-chip MPEG-2 4:2:2 Profile and Main Level (422P@ML)¹ encoder LSI, SuperENC.

Several single-chip MPEG-2 Main Profile at Main Level (MP@ML) encoders have already been developed[34, 32, 40, 44, 49]. They have their own ME/MC architecture. Each architecture implements an ME/MC algorithm that can reduce the number of the ME/MC operations and can maintain the quality of the decoded image. Miyagoshi et al.[32] utilized two 256 processing element (PE) arrays that execute an exhaustive search during P-picture² encodings and a horizontal sub-sampling search during B-picture encodings. Ogura et al.[40] used two identical ME blocks that can search an area measuring ± 32 pixels horizontally and ± 16 pixels vertically, one for searching for motion vectors (MVs) around zero vector (0,0) and the other for searching for MVs around some offset vector.

¹Video coding standards provide profiles and levels. Profile defines a subset of coding tools, and level defines the number of pixels per second to be encoded.

²P-pictures (predictive coded pictures) are those coded using motion compensated prediction from a past intra or predictive coded picture. B-pictures (bidirectionally-predictive coded pictures) are those coded using both past and future reference pictures for MC. I-pictures (intra-coded pictures) are those coded using prediction within themselves.

Mizuno et al.[34] developed a PE array in which each PE reads a reference datum and a bit-map datum simultaneously. The bit-map datum indicates the validity of the reference datum. Because no search operation is executed for invalid reference data, the power consumption of the ME can be reduced. All of these ME/MC architectures can be implemented in a small area, and consume very little power, which are required for implementing an MPEG-2 encoder into a single chip. However, none has made enough evaluations of the quality of decoded image, and none has also taken consideration of scene-adaptivity as described below.

The ME/MC has a variety of parameters. The parameters have the appropriate values for a given video sequence, called a *scene*. These values vary from one scene to another. Therefore, to improve the quality of the decoded image, a new ME/MC algorithm will be required that can be controlled adaptively according to the scene being encoded. A scene-adaptive algorithm must be able to *quickly* adjust various encoding parameters so that their values are as appropriate as possible for the scene. The algorithm proposed in this chapter can vary the search area (range and location), the priority given to MVs, and the MC mode selection criteria. Some of the algorithms implemented in previous ME/MC architectures can set an offset vector of the search area in the picture cycle, i.e., once per picture. This may enhance the quality of the image in some scenes, but may degrade it in others. When the difference between the offset vector and the actual movement is large, the quality of the picture deteriorates. Moreover, if the picture is an I-picture or a P-picture, the deterioration spreads to other pictures. The picture cycle is too long to adjust the offset vector of the search area. Thus, a scene-adaptive algorithm must be able to control encoding parameters in the slice cycle or even in the macroblock (MB) cycle³.

This chapter proposes a ME/MC hardware architecture with functional block level flexibility for implementing a scene-adaptive algorithm. The architecture consists of two modules, a search engine (SE) and a single-instruction-stream multiple-data-stream macroblock processor (SIMD). The SE executes wide and coarse MV searches, and the SIMD is responsible for fine searches, MC, and other operations on the MBs. The most significant feature of the architecture is that these two modules have no direct communication with each other. The independence of the two modules enables a time interval of any length to be inserted between their operations. That is, the SE's operation can be done a number of pictures ahead of the SIMD's operation. Thus, statistical information can be ob-

³Slice is a horizontal MB line. The slice cycle means the period while an encoder processes a slice. The MB cycle means the period while an encoder processes a MB.

tained from the scene by using the results of the coarse searches before MC starts. The values of the parameters are calculated by analyzing the information, and are sent to the SE and the SIMD in the slice or MB cycle.

This ME/MC hardware architecture was implemented in the single-chip MPEG-2 MP@ML video encoder[31, 11], SuperENC. The chip can also be used as a single-chip 4:2:2P@ML encoder[50], and several chips can be used together to create a single-board MP@HL (Main Profile at High Level) or 4:2:2P@HL encoder[46].

This chapter is organized as follows. Section 3.2 proposes a scene-adaptive algorithm. The area hopping method as well as other extensions of an ME/MC algorithm is explained in the section. Section 3.3 details the proposed ME/MC architecture and shows how the scene-adaptive algorithm is implemented in it. Section 3.4 discusses the implementation results. Section 3.5 summarizes this chapter.

3.2 Scene-adaptive algorithm

ME/MC is a procedure for finding the MVs that represent spatial displacement from a MB being encoded to the best matching place of the reference pictures that have already been encoded. The procedure also calculates the difference between the MB and the parts of the reference pictures that are pointed to by the MVs. The full search (FS), a classical ME/MC method, exhaustively examines the matching with regard to all possible MVs in the search area. Because heavy computational load is required when the search area is wide, the FS is not suitable for a single-chip MPEG-2 encoder.

3.2.1 Hierarchical telescopic search

The hierarchical telescopic search[45] method is utilized as the basis of ME/MC algorithm. The method greatly reduces computational load and maintains the quality of the decoded image. It was also used in the two-chip MPEG-2 SP@ML (Single Profile at Main Level) encoder, ENC-C and ENC-M, developed in 1995[29]. The hierarchical telescopic search, which is a combination of a hierarchical search and a telescopic search, consists of three steps as shown in Figure 3.1:

1. Search for a 2-pel precision MV with a telescopic search on the original

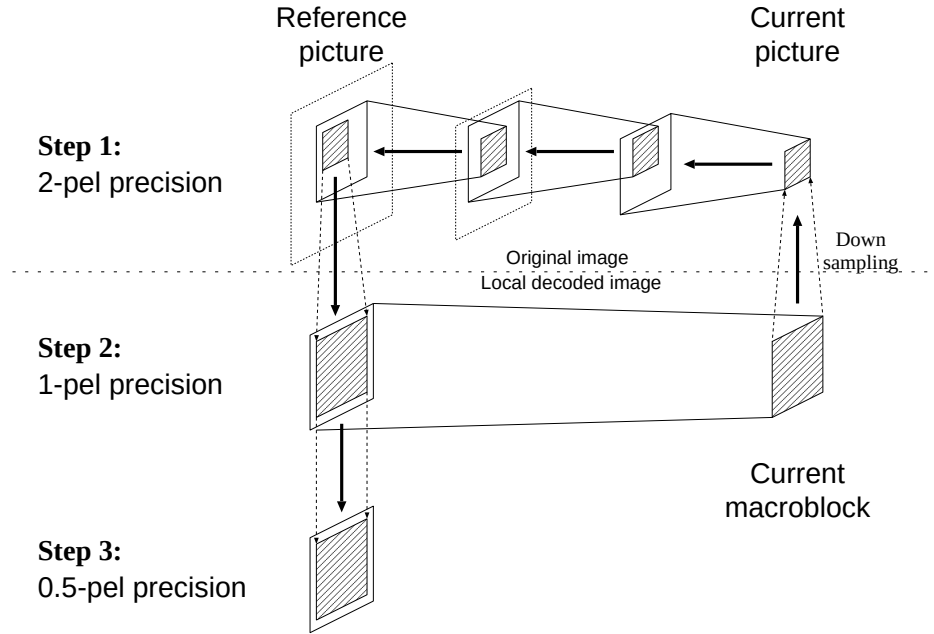


Figure 3.1: Hierarchical telescopic search.

images down-sampled at a ratio of 4:1.

2. Find a fine vector with 1-pel precision around the 2-pel precision vector on the local decoded images.
3. Conduct the same Step 2 search with 0.5-pel precision around the 1-pel precision vector found by Step 2.

The telescopic search, applied at the first step of the hierarchical search, uses intermediate pictures between current and reference pictures. It starts with an exhaustive block matching over a search area in the nearest intermediate picture. Then, the next block matching is executed over a search area in the second nearest intermediate picture. The same matching procedures are continued until a search area is in the reference picture. In each block matching, a search area is centered on the corresponding position of the best matched block in the previous block matching. Let d be a distance between a current picture and a reference picture, and A be a search area of the exhaustive block matching (the solid-line square area in the intermediate pictures and the reference picture in Fig. 3.1). The dotted-line square area d^2A in the reference picture in Fig. 3.1 is covered with d times of block matching whose search area is A . Thus, the telescopic search can reduce the computational load required for the first step of the hierarchical search to $1/d$

compared to the FS. The preliminary experiment shows that, by the hierarchical telescopic search, the quality degradation of the decoded image is less than 0.5 dB.

3.2.2 Scene-adaptive control

The parameters of the hierarchical telescopic search in the two-chip MPEG-2 SP@ML encoder[29] are fixed regardless of the scene being encoded. For example, the range of the search area is pre-defined and constant throughout encoding. This lack of flexibility with respect to the kind of the scene being encoded affects image quality. The quality can be improved if the hierarchical telescopic search can choose the search parameters that best fit the scene being encoded. This concept is called *scene-adaptive control*. In order to introduce scene-adaptive control, this section proposes the following modifications to the hierarchical telescopic search.

Area hopping method

The first modification is to allow the search area to be expanded during the encoding of a scene. The search area is initially set wide enough for ordinary scenes in MP@ML. When a wider area is needed, as in a scene with a large amount of motion or in MP@HL, it is expanded by using some non-zero vector as an offset vector, which is called *hopping vector*. The hopping vectors can be set at the start of every slice process to track various motions within a picture. As a result, the search area can be *hopped* within a picture (Fig. 3.2). This is called *area hopping method*. Although similar methods have already been proposed, they can decide a hopping vector once in a picture. The proposed area hopping method can set a hopping vector every slice.

The hopping vectors can be calculated by analyzing the MVs of the MBs that have already been encoded. In the proposed area hopping method, the hopping vector HV_i for the i -th slice is calculated as a simple average vector in the $(i - 1)$ -th slice such that

$$HV_i = \frac{1}{n} \sum_j MV_{(i-1,j)}, \quad (3.1)$$

where n is the number of MBs in a slice and $MV_{(i-1,j)}$ is the regularized 2-pel MV of the j -th MB in the $(i - 1)$ -th slice. Since the MVs in a slice may have different direction (forward/backward) or different picture distance (the number of pictures), the regularization should be done before summation. It is confirmed

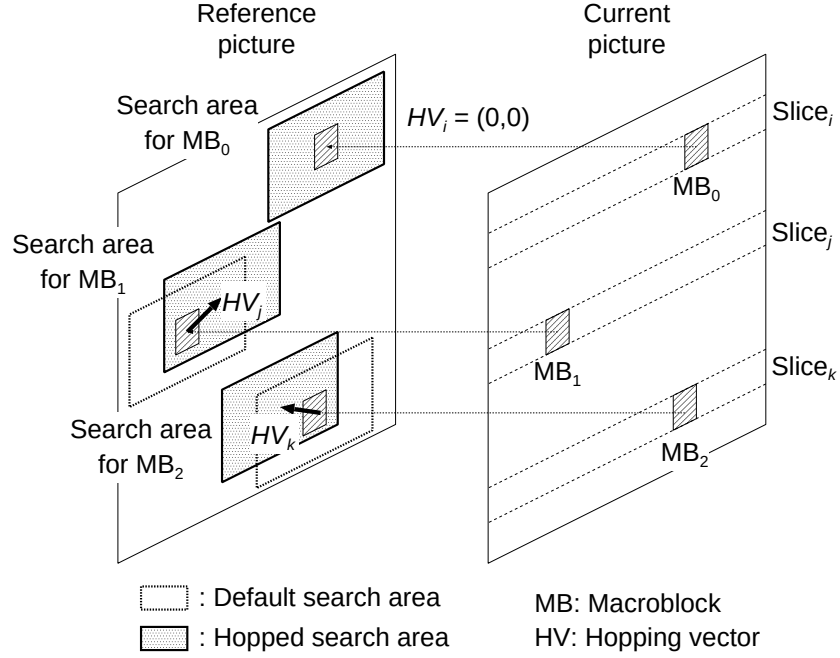


Figure 3.2: Area hopping method.

that this simple average of the MVs of the MBs in the previous slice has a good effect. The results are described in detail in Section 3.4.

As for the search area, MPEG-2 video bitstream has a parameter, f_code , that restricts the range of the motion vector components in a picture. Although the f_code values can be varied from picture to picture, they are fixed in the bitstreams produced by almost all hardware encoders. The reason is explained as follows. The f_code values are placed in a picture header. An encoder produces the f_code at the start of the picture encoding process. The encoder cannot find the lowest f_code values until all MBs in the picture are processed. Therefore, the encoder uses the maximum search area to set the f_code values, even though those values may be too large for the range of the the MV components in the picture. The range of MV components are always equal to or smaller than the maximum search area. However, if the f_code values are larger than needed, more bits are needed for encoding the MVs themselves, which leads to a deterioration of the coding efficiency. An experiment shows that addition one to the normal f_code values results in a 1-dB degradation of peak signal-to-noise ratio (PSNR) of images decoded from a bitstream (MP@ML, 4 Mbps). Because the variable search area of the area hopping method may be about twice as wide as that of the normal mode, optimization of the f_code values is required. Therefore, the

scene-adaptive concept is applied to calculate the f_code values. Before the picture encoding process starts, the MVs of all MBs in a picture are calculated. Then, the lowest f_code values are calculated from the range of the MVs in the picture. Note that the adaptive f_code setting never prevents real-time encoding, though encoding latency become larger. The proposed adaptive f_code setting can enhance the image quality for some scenes, and never degrade the image quality for any scenes.

Motion vector selection

The second modification to the hierarchical telescopic search concerns MV selection. By the area hopping method described above, the search area can be expanded. On the other hand, when the scene is a still image or when there is little motion, the search area should be contracted in order to prevent using bits that are needed for encoding the MVs themselves. In the scene with little motion, the proposed MV selection algorithm can contract the search area. When all MVs of some picture I' are within the area A' , the search area of the following picture I is set to $d/d' \times A'$, where d is the number of pictures between I and its reference picture, and d' is the number of pictures between I' and its reference picture.

When the scene is changing uniformly, such as when a camera pans, the last encoded MV should be given priority, because the MVs themselves are encoded differentially with respect to the last encoded vectors by using variable length codes. The weights of the priority functions proposed are made to be variable. More priority can be given to the last encoded vector in such a scene. The priority function translates the sum of absolute difference (SAD) sad_v of the last encoded vector v into

$$sad'_v = sad_v - th. \quad (3.2)$$

Here, $th > 0$ is a threshold value. The resulting value sad'_v is compared with the other SADs. The priority function overcomes a weak point of the telescopic search that the variance of the vector tends to be large.

Motion compensation mode selection

The third modification is to make the criteria of the MC mode selection variable. The MC mode characterizes the MB properties: the field/frame prediction, the forward/backward/interpolative prediction, the inter/intra decision, and so on. The DCT, quantization, and Variable Length Coding (VLC) are performed in the

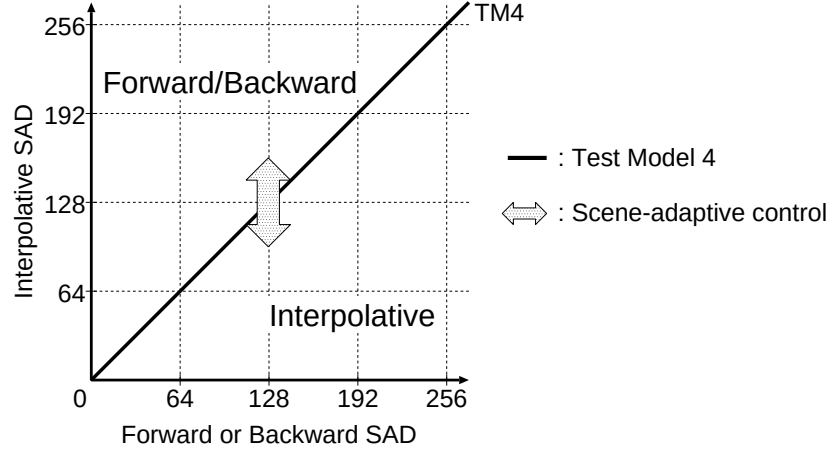


Figure 3.3: Function of the forward/backward/interpolative decision.

selected MC mode. Hence, the MC mode selection has a great effect on the whole coding efficiency.

The MC mode selection criteria in Test Model 4 (TM4)[21], the reference software of MPEG-2, are fixed. For example, the function of the forward/backward/interpolative decision in TM4 is shown in Fig. 3.3. A scene-adaptive algorithm can control the function of the MC mode selection dynamically during encoding, as the arrows in Fig. 3.3 illustrate. In the TM4, the interpolative prediction mode is selected if $(\text{SAD of Interpolative}) < (\text{SAD of forward/backward})$. In the proposed scene-adaptive control, the interpolative prediction mode is selected if $(\text{SAD of Interpolative}) < (\text{SAD of forward/backward}) - f(q)$, where $f(q)$ is a function of the average quantizing scale q in the previous picture. This is because, when q is large, bits for headers, such as MVs, should be decreased. Using $f(q)$, the proposed method can slide the TM4 line downward when the scene is hard to be encode, or upward otherwise.

In addition to the ME/MC algorithm, the rate control operation can be made scene-adaptive by using ME/MC results such as MVs or SADs. Although the conventional rate control is already “adaptive”, it only uses the statistics of the encoded results in a feed-back way. That is, the conventional rate control only uses the information from previously encoded pictures to control the encoding parameters. By using the ME/MC results, it can also use the information from the current picture or even from pictures that have not been encoded yet. This type of “feed-forward” control, which is a kind of a scene-adaptive algorithm, has not

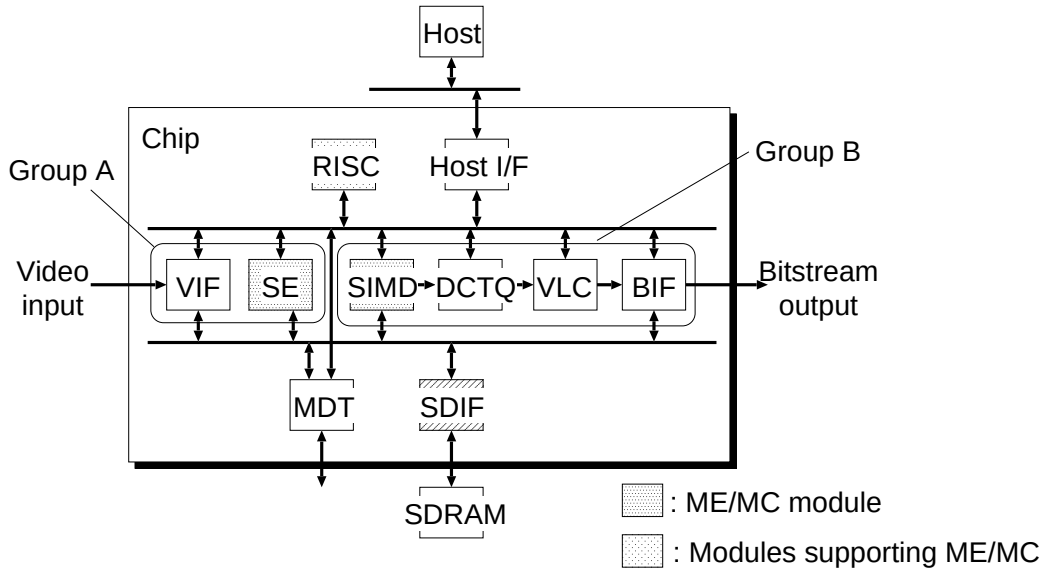


Figure 3.4: Block diagram of the SuperENC.

been realized so far because of the constraint of real-time encoding. However, the scene-adaptive control enables a rate control to utilize the ME/MC results of the future pictures during real-time encoding as described in the next section. Thus, coding efficiency can be improved.

3.3 Hardware architecture

3.3.1 Flexible communication architecture

Figure 3.4 shows a block diagram of the SuperENC. The chip is composed of several functional hardware blocks or modules, as well as a Reduced Instruction Set Computing (RISC) processor module (RISC) that controls all of the functional modules by setting their parameters. The data transfer between the chip and the external synchronous DRAM (SDRAM) is managed by the SDRAM interface module (SDIF).

The hardware implementation of the scene-adaptive algorithm is based on the Flexible Communication Architecture (FCA)[31, 11](Fig. 3.5). Communication on the chip is done by the SDIF's embedded program sequencer. When the data are transferred from module A to module B via the SDIF, the SDIF's sequencer can choose whether or not to store the data in the SDRAM. If the data are not stored in the SDRAM, the SDIF relays them without accessing the SDRAM.

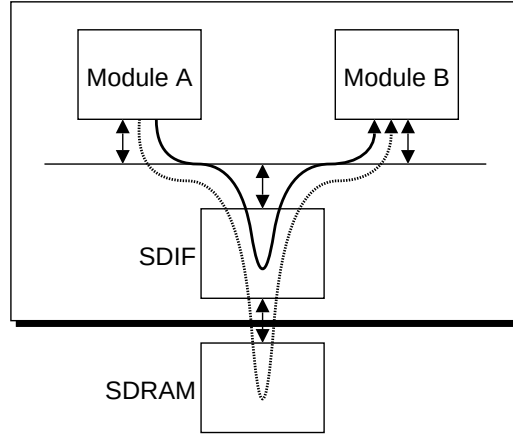


Figure 3.5: Intra-chip communication model with the Flexible Communication Architecture.

Thus, the SDIF can control the data transfer timing and maintain the buffering interval.

The ME/MC has two functional modules. One is a search engine (SE) that executes a wide search with 2-pel precision using the telescopic search method. The other is a single-instruction-stream multiple-data-stream macroblock processor (SIMD) that performs fine searches with 1-pel and 0.5-pel precision, MC mode selection, MC, and local decoded image generation.

The most significant feature of this ME/MC architecture is that the SE and the SIMD have no direct interaction with each other. Neither do they share data memory; the SE uses sub-sampled original images while the SIMD works on local decoded images. Thus, they can operate independently of each other. This means that a time interval of any length can be inserted between the operation of the SE and the operation of the SIMD. The encoding process can therefore be split into two groups of operations (cf. Fig. 3.4):

Group A: filtering by the video interface module (VIF) and a wide 2-pel precision search by the SE,

Group B: MC following fine searches by the SIMD, DCT and quantization by the DCTQ module, variable length coding by the VLC module, and bit-stream output by the BIF module.⁴

⁴Local decoded image generation is also in Group B because it is done by the DCTQ and the SIMD.

By processing Group A several pictures ahead of Group B, the RISC can get the results of the SE for pictures that have not yet been handled by the modules of Group B. The SE's results, such as MVs or SADs, give information about the scene. For example, when the SADs are large for almost all the MBs in a picture, the picture is considered to contain a scene cut. If almost all the MVs in a picture are (0,0) or nearly (0,0), the picture is considered to be a still image, and so on. Furthermore, the number of bits generated by encoding can also be estimated from the SADs obtained by the SE. This helps the rate control operation to allocate the bits more precisely. Hence, the time interval between the SE's operation and the SIMD's operation enables the RISC to analyze a scene in advance of the encoding process. The results of the analysis can be sent to each module as its parameters.

Note that splitting the encoding process into Group A and Group B has no effects on memory bandwidth, because the SE and the SIMD do not share data memory.⁵ When two 64-Mbit SDRAMs are used, 20 frames can be inserted if the application accepts the encoding delay. This number of frames is enough to analyze the scene.

3.3.2 Search engine

The SE requires much computational power when executing a wide search even if using the telescopic search method. The design approach used is to hard-wire the SE for high performance. The RISC and the SDIF, which are the only modules that interact with the SE, have their own programs. When area hopping is needed, the RISC sends the hopping vector to the SDIF instead of the SE. The SDIF then provides the hopped area to the SE. In this manner, the RISC and the SDIF allow the SE to operate in the same way whether the given area is hopped or not. The expansion of the search area by the area hopping method can be controlled in the slice cycle because the SDIF checks the hopping vector before each slice cycle is started.

A block diagram of the SE is shown in Fig. 3.6. The array of the PEs is based on the search unit of the previous two-chip MPEG-2 encoder[45]. Although it is a systolic array with only 32 PEs, the SE can search ± 210 horizontally and ± 112 vertically at a P-picture in $M=3$ by the combination of the hierarchical telescopic search and the area hopping search⁶.

⁵Though, in fact, the results of the SE must be stored in and out of the SDRAM, the input and output (I/O) increase can be negligible.

⁶ M is the number of pictures between I-, or P-pictures.

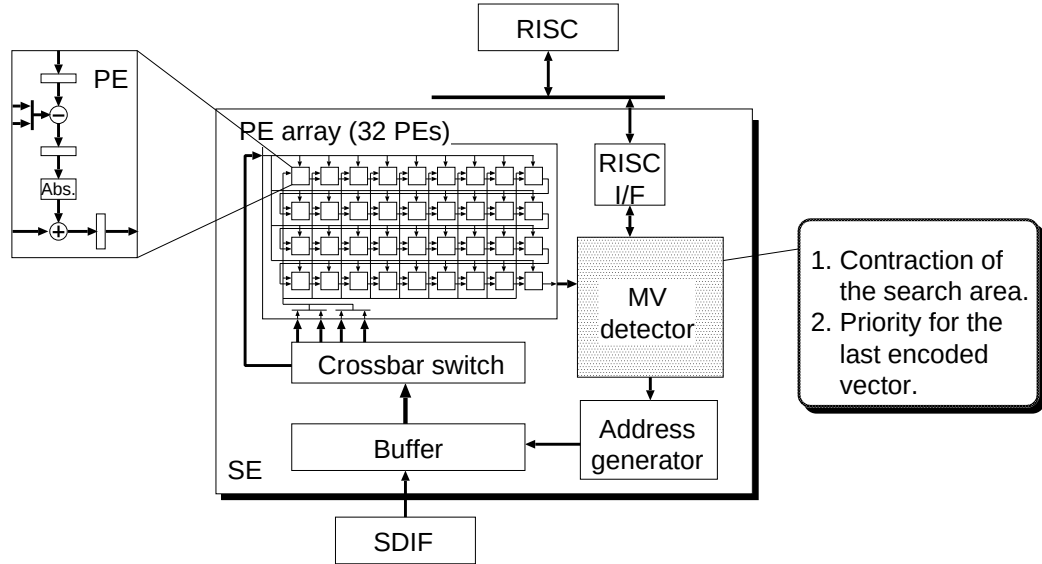


Figure 3.6: Block diagram of the SE.

The key feature of the SE is its intelligent MV detector. It plays two roles in the scene-adaptive algorithm.

- It contracts the search area dynamically, and
- It gives priority to the last encoded vector.

To contract the search area, the detector has parameters for determining the range of the permitted search area. When a vector is out of the permitted area, the detector regards it as invalid and never selects it. Scene adaptivity is achieved by setting the parameters in the slice cycle. To give priority to the last encoded vector, the detector receives the threshold th in Equation (3.2). The threshold th can also be changed when the slice is changed.

The other modules of ME/MC is the SIMD macroblock processor. The architecture of the SIMD is described in Chapter 4 in detail. Here, scene adaptivity in the SIMD is explained. The SIMD has three thresholds, one for field/frame prediction, one for forward/backward/interpolative prediction, and the last one for inter/intra prediction. All three can be set by the RISC for every MB. Thus, the MC mode selection criteria can be changed during encoding.

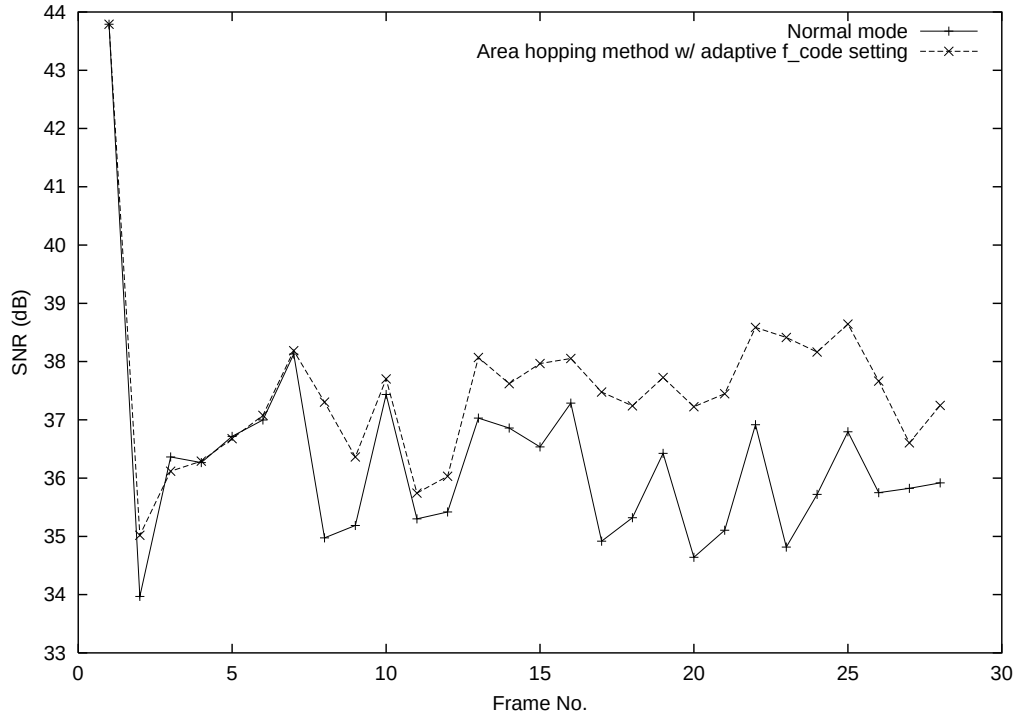


Figure 3.7: PSNR of “football” sequence.

3.4 Implementation results

3.4.1 Image quality evaluation

The area hopping method with the adaptive f_code setting was evaluated for a number of MPEG-2 test sequences. The hopping vector HV_i for the i -th slice is calculated using Equation (3.1). The range of the hopping vector is limited such that the hopped search area includes the zero vector $(0,0)$. After the 2-pel MVs of all MBs in a picture are found, the f_code of the picture is calculated by using the maximum absolute vector components.

The area hopping method works best for scenes containing a large amount of motion, such as the “football” sequence. Figure 3.7 shows the PSNR of the frames of the “football” sequence. The latter half of the scene, which has large amount of motion, is clearly enhanced. The average PSNR of the area hopping method is 1.2 dB higher than that of the normal mode. In the 23rd frame (Fig. 3.8), the block noise in the frame of the normal mode can be suppressed by the area hopping method (Fig. 3.9). In the normal mode, since the motion is too large



Figure 3.8: The original 23rd frame in the “football” sequence. The rectangle shows the part enlarged in Fig. 3.9.

to find appropriate MVs, intra prediction is likely to be selected, which is the cause of the block noise. The area hopping method can find MVs that give small SADs by hopping the search area. Therefore, MBs that select inter prediction are increased. This is the reason why the block noise is suppressed in the area hopping mode. Even in the case of scenes with little motion, the area hopping method causes negligible degradation in the quality. This is because the adaptive f_code setting optimizes the number of bits for encoding MVs. The area hopping method is available for scenes with the size of ML, and it is indispensable for scenes with the size of HL, which require a wider search area than scenes of ML.

It was confirmed that the rest of the modifications to the hierarchical telescopic search, the MV selection and the MC mode selection, also enhance the image quality about 0.1 dB to 0.2 dB. For example, by the adaptive forward/backward/interpolative prediction decision, the “bicycle” sequence is enhanced 0.14 dB.

3.4.2 Chip implementation

The ME/MC architecture was implemented in a single-chip MPEG-2 encoder, SuperENC. Figure 3.10 is a microphotograph of the chip. Table 3.1 shows the physical and functional specifications of the encoder.

The profiles and levels that the encoder can perform are MP@ML and 4:2:2P@ML with a single chip, and MP@HL and 4:2:2P@HL with multiple chips. When multiple chips are used for high level, the reference image data are transferred between chips via a multichip data transfer module (MDT). The search range of the ME/MC of a P-picture at M= 3 is $-113.5/+99.5$ horizontally and ± 57.5 vertically. It can be expanded by the area hopping method to ± 211.5 horizontally and ± 113.5 vertically. This wide area is searched efficiently by the proposed ME/MC hardware architecture. The external memories in the case of the MP@ML are two 16-Mbit SDRAMs. In other profiles or levels, such as the 4:2:2P or the HL, a 64-Mbit SDRAM is used. When the picture delay between Groups A and B is inserted, 64-Mbit SDRAMs are needed in order to store the results of the SE along with the pictures that have been processed by Group A but not by Group B.

Comparison between the proposed architecture and other MPEG-2 encoder LSIs is shown in Table 3.2. The proposed ME/MC architecture has a wider search range than others. The range expansion can be varied in slice cycles. Scene adaptivity, such as f_code optimization or feed-forward rate control, is also supported so that the image quality can be enhanced.

3.5 Chapter summary

This chapter described a ME/MC hardware architecture with functional block level flexibility for implementing the scene-adaptive algorithm. The most important feature of the architecture is the independence of the search engine (SE) and the single-instruction-stream multiple-data-stream processor (SIMD). This enables the encoder to be scene-adaptive because the encoder can analyze the scene before it is encoded by using the results of the SE. Both the SE and the SIMD have ports for encoding parameters that can be changed in the slice cycle or the MB cycle. This means that the search area and MC mode selection criteria can be adjusted quickly, and that the encoder can track the scene condition. The ME/MC hardware architecture was implemented in a single-chip MPEG-2 video encoder.

The concept of scene-adaptive control and the functional block level flexibility presented here will be progressed to the advanced coding control with pre-analysis engines as described in Chapter 5.



Figure 3.9: The part of the 23rd frame in decoded images of “football” sequence. Top: the 23rd frame of the normal mode; bottom: the 23rd frame of the area hopping method with adaptive f_code setting. The block noise in the frame of the normal mode is suppressed in the frame of the area hopping method.

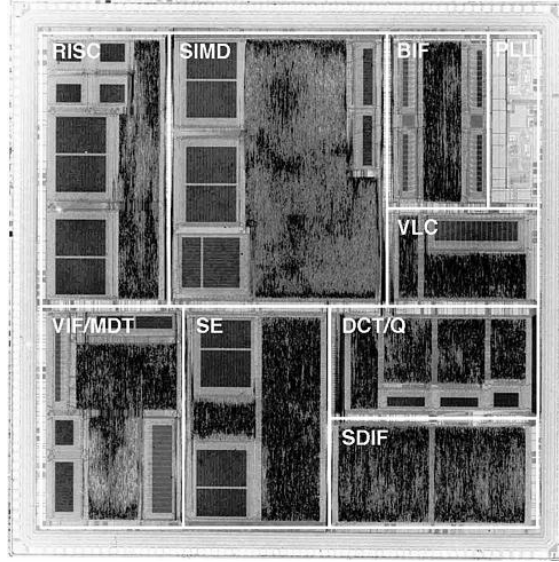


Figure 3.10: Chip photograph of SuperENC.

Table 3.1: Specifications of the SuperENC.

Characteristic	Description
Die Size	$9.8 \times 9.8 \text{ mm}^2$
Technology	$0.25\text{-}\mu\text{m}$ four-level metal CMOS
Supply Voltage	2.5 V (internal), 3.3 V (I/O)
No. of Transistors	5.0 million Tr.
Clock Frequency	81 MHz
Power Consumption	1.5 W
Package	208-pin QFP
Profile and Level	MP@ML, 4:2:2P@ML MP@HL, 4:2:2P@HL (multi-chip)
Search Range (P-pic. at M= 3)	$-113.5/+99.5(\text{H})$, $\pm 57.5(\text{V})$ $\pm 211.5(\text{H})$, $\pm 113.5(\text{V})$ (Area hopping)
External Memory	16-Mbit SDRAM $\times 2$ or 64-Mbit SDRAM $\times \{1,2\}$

Table 3.2: Comparison between MPEG-2 encoder LSIs.

Characteristic	SuperENC	[29]	[34]	[32]
Year	1997	1995	1997	1998
Technology	0.25 μm	0.5 μm	0.35 μm	0.25 μm
No. of Tr.	5.0 MTr.	3.2 MTr.	3.1 MTr.	5.5 MTr.
Power Consumption	1.5 W	3.5 W	1.5 W	0.98 W
Profile/Level (multi-chip)	MP@ML 4:2:2P@ML MP@HL 4:2:2P@HL	SP@ML	MP@ML	MP@ML
Search Range (H) (V)	$-113.5/+99.5$ ± 57.5	± 48.5 ± 32.5	$-47.5/+48$ $-15.5/+16$	± 63.5 ± 47.5
Range expansion (H) (V)	± 211.5 ± 113.5	— —	± 96 ± 32	+offset +offset

Chapter 4

Instruction level flexibility — SIMD macroblock processor

4.1 Introduction

This chapter concentrates on motion estimation and compensation (ME/MC) hardware architecture with instruction level flexibility. A single-instruction-stream-multiple-data-stream (SIMD) macroblock processor is proposed in order to support wide ranging ME/MC coding tools in MPEG-2. High image quality can be reached by combining wide ranging support for ME/MC coding tools with intelligent MC mode selection with scene adaptivity describe in Chapter 3. The instruction level flexibility of the SIMD is also suitable for 4:2:2 encoding, and thus, it contributes to functional expandability of video encoder LSIs.

The SIMD macroblock processor is a module for ME/MC, together with the SE (Section 3.3.2). The first step of the hierarchical telescopic search is realized in the SE. Then, the rest of ME/MC operations become several complicated operations. In conventional encoders, including ENC-M which is a predecessor of SuperENC, these operations are assigned to different specific hardware modules, because such architectures can optimize circuit area. Nevertheless, with the increase of the scale of LSIs, a paradigm shift has occurred in the hardware design field. Design facilitation or flexibility become more important at the sacrifice of circuit area. In this context, the SIMD macroblock processor has been adopted for all of those operations in SuperENC.

The SuperENC was originally targeted mainly at consumer video appliances.

It is, therefore, focused on encoding standard definition (SD, 720×480)¹ videos, as other single-chip MPEG-2 LSIs [34, 32, 40] are. However, according as the start of the digital broadcasting satellite (BS) services and the progress of the plan of the digital terrestrial broadcasting in Japan, the importance of high definition (HD, 1920×1080)² video encoding by MPEG-2 was increased. From these background, SuperENC, which has scalability for video formats³, was paid attention to as a key device of broadcast equipments. However, there is room for SuperENC to be improved in image quality when it is used as broadcast equipments. Therefore, in order to enhance image quality, SuperENC-II was developed in 2000. Although the basic architecture of SuperENC-II (Fig. 4.1) consists of the same modules as SuperENC, each module has been improved in various aspects in order to realize higher image quality. Among these improved modules, this chapter concentrates on SIMD macroblock processor.

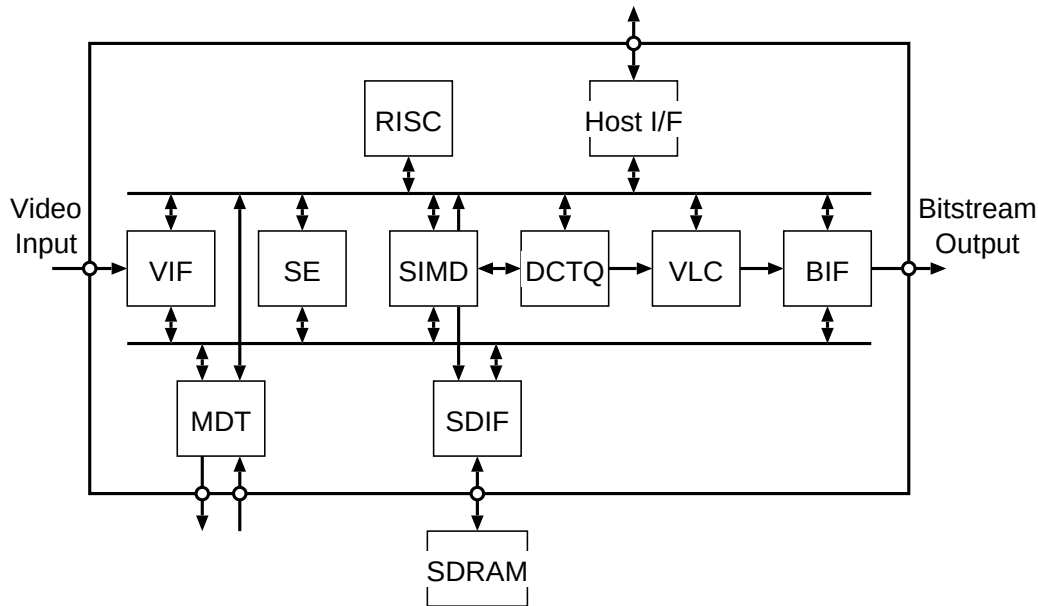
The SIMD in SuperENC-II has to execute more complicated and advanced ME/MC algorithms than SuperENC in order to enhance image quality. To do this, the SIMD is required to increase computational performance. On the other hand, since the instruction memory of the SIMD in SuperENC is almost full, the SIMD is needed to reduce the program size in order to store a new algorithm in SuperENC-II. To meet these two requirements, several improvement methods are proposed, such as an addition of hardware for specific operations, an adoption of a memory architecture which increases input and output (I/O) throughput for pixel data, and an introduction of a new instruction set. As a result, compared with the one in SuperENC, the SIMD macroblock processor in SuperENC-II can achieve 1.5 times performance, and can reduce the program size to 64 % simultaneously.

This chapter is organized as follows. At first, Section 4.2 shows the SIMD architecture in SuperENC. Next, the proposed improvement methods to enhance image quality are explained in details in Sec. 4.3. The results from these methods are evaluated in Sec. 4.4. The implementation of the SIMD to SuperENC-II is explained in Sec. 4.5, and Section 4.6, at last, summarizes this chapter.

¹The SD format is included in Main Level (ML) in MPEG-2.

²The HD is in High Level (HL) in MPEG-2.

³As described in Chapter 3, SuperENC supports standalone encoding of ML videos. It supports encoding of HL with a multiple-chip configuration[46].



VIF: Video Interface	DCTQ: DCT and Quantization
MDT: Multi-chip Data Transfer	VLC: Variable Length Coding
SE: Search Engine	BIF: Bitstream Interface
SIMD: SIMD Macroblock Processor	RISC: RISC Processor
SDIF: SDRAM Interface	Host I/F: Host Interface

Figure 4.1: Block diagram of the SuperENC-II.

4.2 SIMD macroblock processor

ME/MC operations except for the SE in MPEG-2 video encoding needs various operations as shown in Table 4.1. In other encoders as well as the previous two-chip SP@ML encoder, ENC-C and ENC-M, these operations were mapped to different functional modules. In such architectures, many modules simultaneously access a shared memory that stores the local decoded images. Since MP@ML, unlike SP@ML, has many prediction modes for high coding efficiency, more flexible access to the local decoded images is needed in order to use all the modes. As a result, a complicated memory management system is needed, and a real time scheduling for the operations becomes tight and difficult.

To solve these problems, a SIMD macroblock processor (SIMD) is utilized for all the above operations. Since the SIMD is the only module that accesses the local decoded images, no memory management system is needed. The above operations can be scheduled by programming the software for the SIMD.

Table 4.1: MC operations assigned to SIMD macroblock processor and their complexity (MOPS) in SuperENC-II.

Operation	Content	Complexity (MOPS)
Variance calculation	Calculate variance of macroblock for Inter/Intra decision.	83
1-pel precision search	Motion estimation with 1-pel precision around specified vectors.	1,680
half-pel precision search	Motion estimation with 0.5-pel precision around specified vectors.	1,003
Prediction error image generation	Generate prediction error images in accordance with MC mode.	93
DCT type decision	Decide field/frame DCT using prediction error images.	47
Local decoded image generation	Generate local decoded images from prediction errors made by IDCT.	93

Figure 4.2 depicts the SIMD macroblock architecture in SuperENC. In order to process a 16-pixel line of a MB in parallel, it comprises 18 PEs, two of which work as edge registers. Pixel data supplied from the buffer are processed in the PEs, and the results are accumulated in the tree adder. The controller can get calculation results such as SADs by accessing the tree adder.

The PE consists of an arithmetic logic unit (ALU) and a register file. The ALU can execute absolute difference and saturation operations, as well as addition and subtraction. The PEs are connected linearly, and can transfer pixel data to the next PE, which is used for reference image data transfer in MV searches.

The functions in the operations of the SIMD include this type of expression:

$$\sum_{column} \left\{ \sum_{line} (expression\ of\ pixel\ level) \right\}. \quad (4.1)$$

The inner summation is done in parallel, and the outer is done serially. In other words, all PEs operate one line at a time and with all lines in a line-serial manner. This enables the SIMD to handle both field and frame data easily. It is also useful for the 4:2:2 profile because the difference between the 4:2:0 and 4:2:2 formats is only the vertical size of chrominance signals.

The SIMD has its own program in an instruction memory. The controller

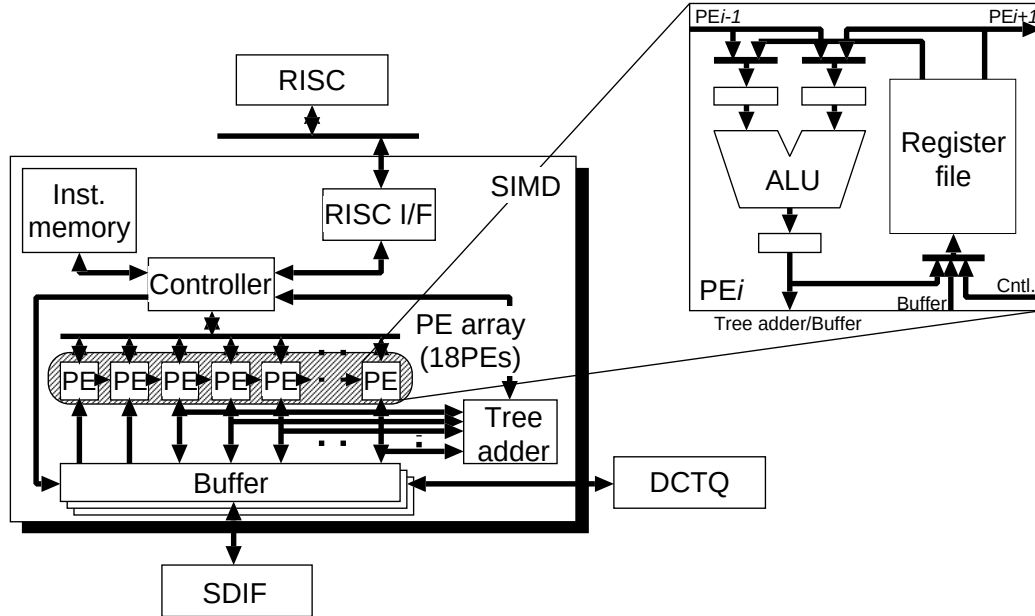


Figure 4.2: SIMD macroblock processor architecture in SuperENC.

issues a double instruction 32-bit in length. One is for the PE array and the other for the controller itself. Figure 4.3 shows a double-issued instruction of the SIMD. The instruction for the PE has three operands that can specify its own register file as well as the register file of the PE to the left. The instruction for the controller is used to control the whole SIMD and for MC mode selection. Thus, in the SIMD of SuperENC, all operations are written in software programs. This flexibility enables to support Simple Profile, Main Profile, and 4:2:2 Profile only by rewriting the software programs.

4.3 Improvement on SIMD

4.3.1 Approaches for improving the SIMD performance

SuperENC-II has to implement an advanced and complicated ME/MC algorithm in order to enhance image quality. The computational load of the algorithm sums up to 3 GOPS (Table 4.1), while the SIMD of SuperENC substantially has execution power of 2 GOPS. Thus, it is required that the performance of the SIMD is improved 1.5 times.

To meet this requirement, two approaches have been taken. At first, in a straightforward way, specific execution hardware units are added. More precisely,

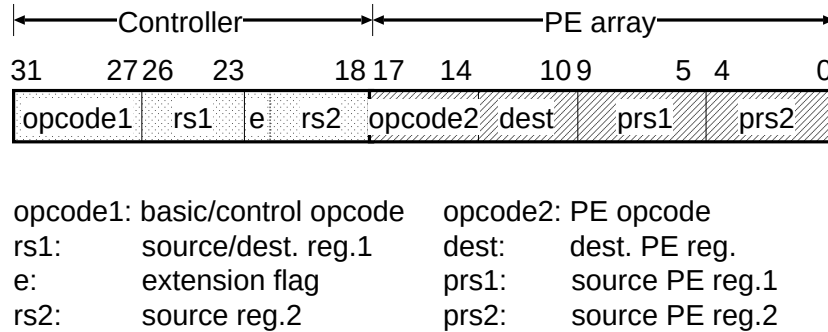


Figure 4.3: Double-issued instruction of the SIMD in the SuperENC.

hardware units to generate interpolated pixel data and to calculate a variance of pixel values in a MB are newly introduced. These hardware units are expected to save several hundreds of dynamic steps in the MB cycle.

Second, to increase the effective ratio of the number of the PE arrays operations issued to the whole dynamic steps, which is about 50 % in the case of SuperENC, pixel data input and output (I/O) throughput are improved by 50 %. The throughput improvement makes reduction of the stall period for the PE to be supplied with the data, and the increase of the rate of PE operations is expected.

For the SIMD of SuperENC, there is another problem that the program size (the number of the static steps) of the software is large. Although the SIMD has an instruction memory with the size of 4 kilo (= 4,096) words, it is almost full because the program size is 4,031 words. It is very hard to add or modify the software in SuperENC. To settle this problem, the instruction format is optimized to utilize registers efficiently and to shorten instruction steps for controller, and a loop instruction, which compresses the number of static steps for repetitive PE operations, is introduced. These optimization for instruction set can also reduce dynamic steps, as well as static steps.

4.3.2 Addition of specific execution hardware

The most straightforward way to improve performance is to add hardware resources. It should be paid attention to so that the addition of hardware resources does not arise memory conflicts with the PEs, because the conflicts make the stall periods of the PEs. Taking this into consideration, the following hardware units are introduced to the SIMD of the SuperENC-II.

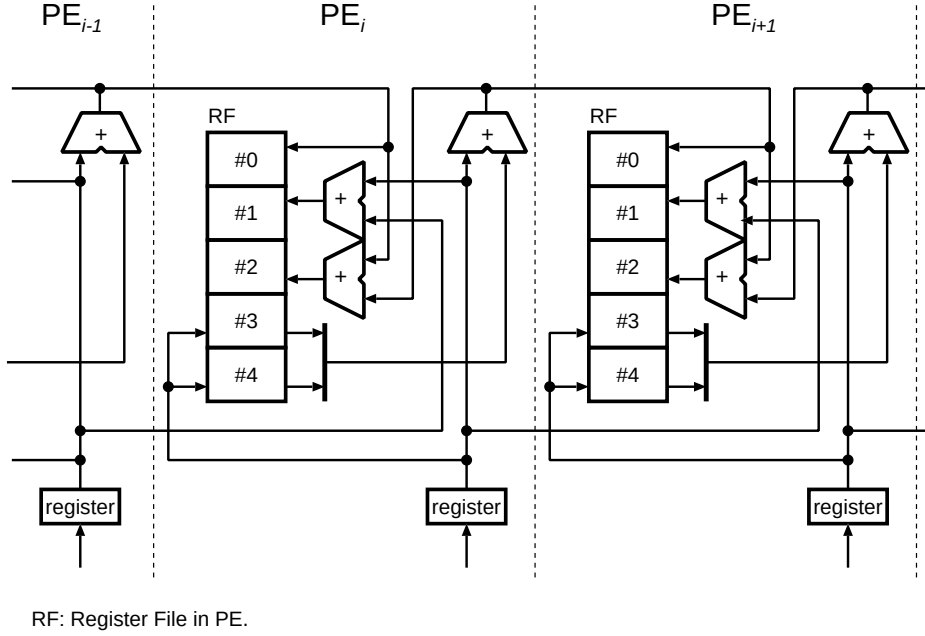


Figure 4.4: Pixel interpolator.

Pixel interpolator

The SIMD has to generate the interpolated pixels of the reference picture at half-pel precision search and prediction error image generation. These operations were processed by the ALU of the PE in the SIMD of the SuperENC. It takes a few instructions to generate a line of interpolated pixels. Therefore, a dedicated pixel interpolator is newly added. When the PE reads the reference data, it generates interpolated pixels and stores them into registers in the PE.

The pixel interpolator is shown in Figure 4.4. Each PE, PE_i , has original pixels in the i -th column. The original pixels in the i -th column is processed with the nine interpolated pixels generated from the reference pixels in the $(i-1)$ -th column, the i -th column, and $(i+1)$ -th column, which is surrounded in Figure 4.5. There are three kinds of interpolated pixels; vertical half pixels ($\triangle$ in Fig. 4.5), horizontal half pixels ($\square$ in Fig. 4.5), and horizontal and vertical half pixels ($\times$ in Fig. 4.5). When every integer pixel is read, these half pixels are simultaneously generated and stored #0, #1, and #2 in the register file of the PEs, respectively. Because PE_i can access the register file of PE_{i-1} , as well as the one of itself, the pixel interpolator can provide all of interpolated pixels needed by the PEs.

The pixel interpolator works as follows. The reference pixel $r_{i+1,j}$ in the $(i+1)$ -th column of the reference image is input to PE_i . If j is odd, $r_{i+1,j}$ is stored into

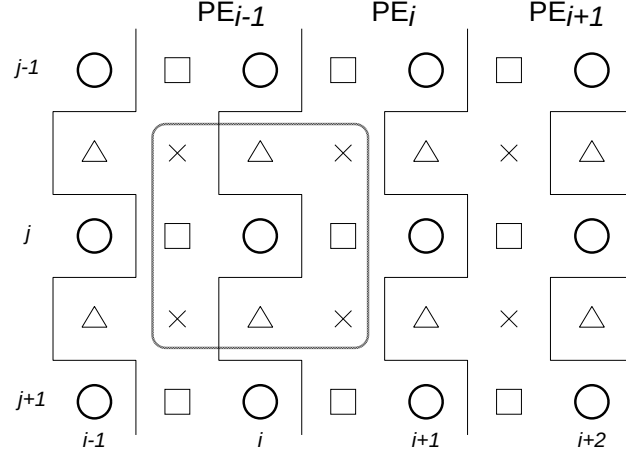


Figure 4.5: Interpolated pixels stored in PE_i .

the register #3 in PE_i . Otherwise, $r_{i+1,j}$ is into the register #4. Simultaneously, the pixel interpolated horizontally between $r_{i+1,j}$ and $r_{i,j}$ ($\square$ in Figure 4.5) is generated and stored into the register #1. The pixel interpolated vertically between $r_{i+1,j}$ and $r_{i+1,j-1}$ ($\triangle$ in Figure 4.5) is stored into the register #0 in the next PE, PE_{i+1} . (The register #0 in PE_i stores the pixel generated by PE_{i-1} .) At last, the pixel interpolated horizontally and vertically ($\times$ in Figure 4.5) is stored into the register #2. When PE_i needs the interpolated pixels stored in PE_{i-1} , they are transferred between PEs.

Variance calculator

The variance of a MB is used for Inter/Intra decision. It is calculated by the PEs of the SIMD in the SuperENC, and it takes about 160 cycles. While other operations executed by the SIMD use two kinds of pixel data, such as the original and the reference, the variance calculation needs only the original pixel data. A variance calculator, therefore, is added, which calculates a variance of a MB in parallel when the original pixel data are stored into the buffer in the SIMD.

When the variance calculator accesses the buffer, the cycle-steal architecture described below is adopted in order to prevent the PEs from stalling. Although the variance calculator may stall, it does not matter because the number of operations of the variance calculation is much less than the other PE operations.

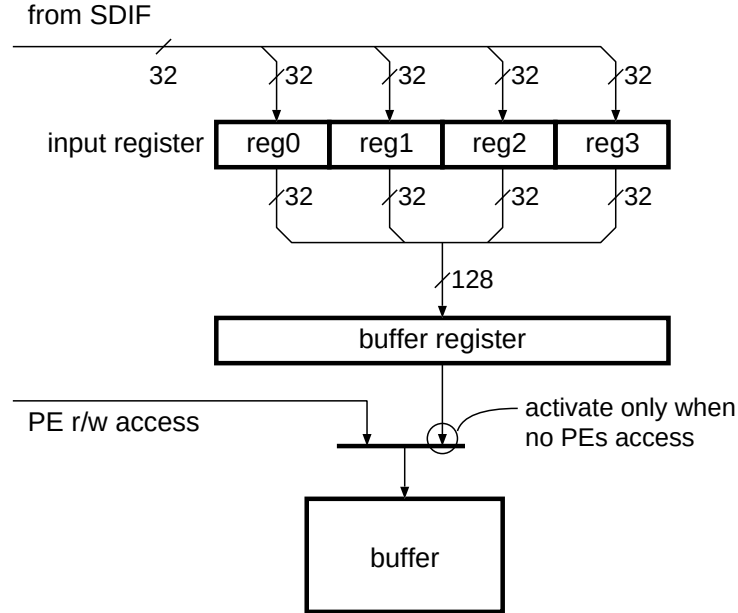


Figure 4.6: Cycle-stealing architecture.

4.3.3 Improvement on I/O throughput of image data

As the Amdahl's law shows, it is important for parallel processing to increase the rate of operations that can be parallelized. As for SIMD, steady data supply to the PEs is desired so as not to decrease the rate of PE operations. To improve the I/O throughput of the PE, two methods are utilized. One is to utilize the two-port register file, and the other is to introduce the cycle-stealing architecture as described below.

Cycle-stealing architecture

The buffer of 128-bit width in the SIMD is accessed by the PE array from inside of the SIMD, and the SDRAM interface (SDIF) from outside, respectively. In the SuperENC, they are permitted to access the buffer alternately in the clock cycle level so as to realize a pseudo two-port access. As a result, the PEs cannot access the buffer continuously. The SIMD in the SuperENC-II introduces a cycle-stealing architecture, which gives priority to the access by the PEs and allows the SDIF to access the buffer only while there is no accesses by the PEs (Figure 4.6).

Figure 4.7 illustrates an example of the write access by the SDIF. The 32-bit data sent from the SDIF are stored into input registers, reg0, reg1, reg2, reg3, reg0, reg1, ..., in every cycle. The data from reg0 to reg3 are moved to a buffer register

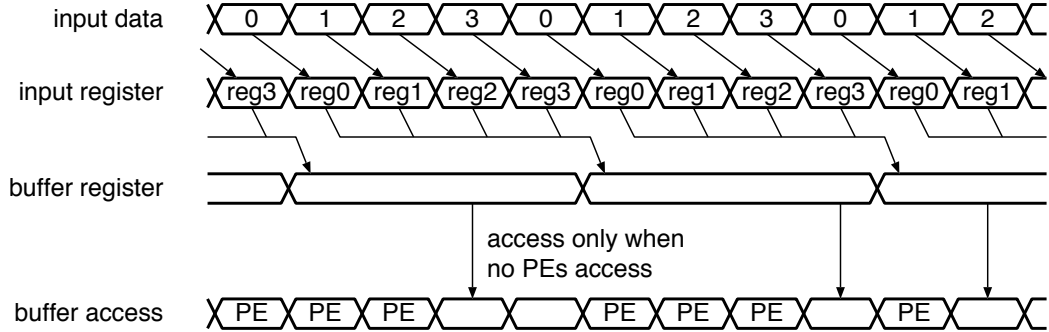


Figure 4.7: Example of write access by the SDRAM interface.

at the time all of the data from reg0 to reg3 are stored. Since the access by the PEs has priority over the access by the SDIF, the write access from the buffer register to the buffer is allowed only when the PEs do not access the buffer. However, if the PEs continue to access the buffer during four cycles, the buffer register may overflow. Therefore, the access by PEs is restricted so that at most three cycles of continuous access are allowed. The cycle-stealing architecture leads to the improvement of the I/O throughput because the PEs can make successive accesses, while the access by the SDIF has to wait at most four cycles. The check whether this restriction is fulfilled or not is in charge of the assembler of the software, instead of hardware, so that the cycle-stealing architecture can be realized by a simple circuit.

Note that, in SuperENC-II, the variance calculator (Sec. 4.3.2) also accesses the buffer. In the same way as the SDIF access, the access to the buffer by the variance calculator is allowed during the PEs do not access.

4.3.4 Optimization of instruction set architecture

The instruction format issued by the SIMD controller is a type of long instruction word (LIW), in which two instructions are issued at the same time. One instruction is for execution within the controller and data transfer between the controller and other blocks (the PE array, the tree adder, and the buffer), called instruction #0. The other is for the PE operations, called instruction #1 (Figure 4.8).

In SuperENC, while the instruction #1 has three operands, the instruction #0 has only two operands in order to configure the length of the instruction word in 32 bit (Figure 4.3). This causes the increase of the static steps of the instruction #0. Therefore, SuperENC-II extends the length of the instruction #0 and makes to

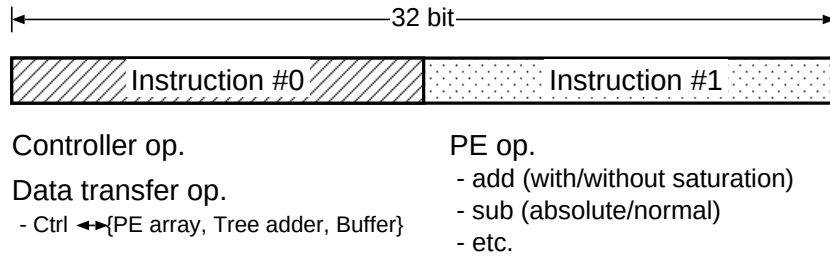


Figure 4.8: LIW instruction format issued by SIMD controller.

have three operands. Since the length of the LIW remains 32 bit, the length of the instruction #1 becomes shorter. It is settled by the reduction of the register words in the PE register file. Although the data transfer between the PE register file and the buffer increases owing to the reduction of the register words, the two-port register file, as described above, can make the data transfer efficiently.

Introduction of loop instruction

Another way of reducing the static steps is to introduce a specific instruction. Since the operations by the SIMD involve a lot of repetitive executions, a substantial reduction of the static steps is expected by adding a loop instruction. A loop instruction is realized by adding hardware unit controlling loop instruction in the controller.

Figure 4.9 illustrates software examples of SIMD in SuperENC and SuperENC-II. While it needs 13 steps in SuperENC, the same executions can be written only in four steps in the case of SuperENC-II. In the SuperENC, a loop is realized as follows. A general register (r4 in Fig. 4.9) is used as a loop register. The 0-th line sets the number of loops to the loop register, and it is subtracted by one in the 5-th line. Then, it is compared and branched in the 7-th line and the 8-th line, respectively. Moreover, each register is incremented by one according to the loop number in the 9-th to the 11-th lines. In SuperENC-II, the loop control unit can realize a loop by setting the loop scope and the number of the loop (the 0-th line). The general registers in the loop are implicitly incremented.

Note that making instruction #0 three operands and the continuous access by the PE described in Sec. 4.3.3 also contributes the reduction of the static steps⁴.

⁴In SuperENC, the PE cannot access the buffer continuously. Thus, the load to the PE is realized by a couple of instructions, `pld` and `plddst`. The PE can access the buffer only when `plddst` is issued.

# SuperENC software example		
#		
# inst.#0	; inst.#1	# comment
#	-----	
0: li	r4,#8 ;	# r4 <- 8
@LOOP:		# label "LOOP"
1: pld	r1,#0 ;	# pr1 <- (r1+0)
2: plddst	pr1 ;	#
3: pld	r2,#0 ;	# pr2 <- (r2+0)
4: plddst	pr2 ;	#
5: subi	r4,#1 ;	# r4--
6: pst	r3,#0 ; padds pr3,pr1,pr2	# (r3+0) <- pr1 + pr2
7: slt	r4,r0 ;	# set flag if r4 < r0
8: bnez	LOOP ;	# branch if flag is set
9: addi	r1,#1 ;	# r1++ (delayed jump)
a: addi	r2,#1 ;	# r2++ (delayed jump)
b: addi	r3,#1 ;	# r3++ (delayed jump)

# SuperENC-II software example		
#		
# inst.#0	; inst.#1	# comment
#	-----	
0: loop	#3,#8 ;	# loop
1: plda	pr1,r1,#1 ;	# pr1 <- (++r1)
2: plda	pr2,r2,#1 ;	# pr2 <- (++r2)
3: psta	r3,#1 ; padds pr3,pr1,pr2	# (++r3) <- pr1 + pr2

Figure 4.9: SIMD software examples. Upper is for SuperENC, lower is for SuperENC-II.

4.4 Evaluations

Figure 4.10 shows the reduction of the dynamic steps by improvement on the SIMD macroblock processor in Sec. 4.3. While the SIMD in SuperENC cannot perform the operations even in P-pictures under the real-time condition (within 2,000 cycles), the SIMD in SuperENC-II is satisfied with the real-time condition in all kinds of pictures, by reducing about 700 cycles in both P-pictures and B-pictures. The performance improvement is about 1.5 times in P-pictures.

The breakdown of the reduction of the dynamic steps is shown in Fig. 4.11. It is found that there are much reduction on variance calculation, 0.5-pel search, and prediction error image generation. This shows effectiveness given by addition of

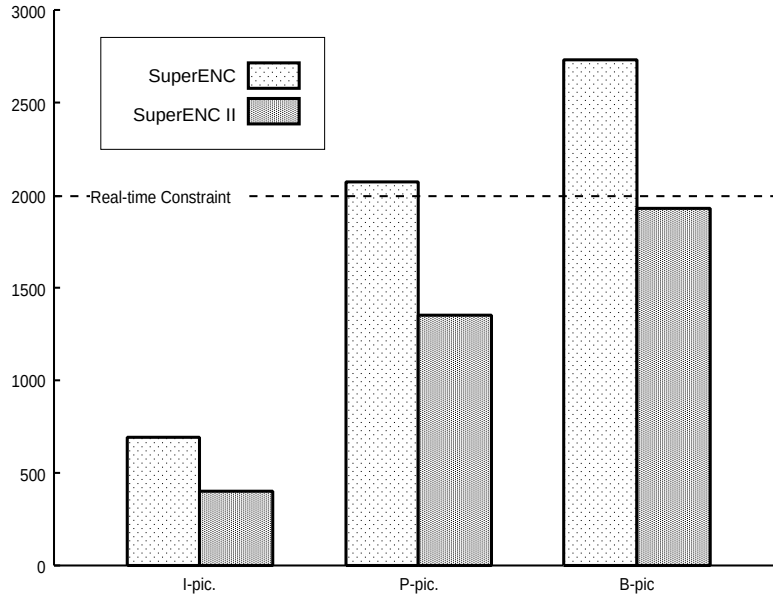


Figure 4.10: Comparison of the numbers of dynamic steps.

pixel generator and variance calculator.

The performance enhancement by the improvement on the SIMD can also be understood from the rate of the PE operations, which is defined as the ratio of the issue of instruction #1 to the total number of the dynamic steps. The rate of the PE operations, which was 51 % (P-pictures) and 60 % (B-pictures) in SuperENC, is enhanced to 55 % and 66 % in P-pictures and B-pictures, respectively (Fig. 4.12). It is because the PEs can execute the instruction #1 successively without waiting for the data transfer by the improvement on the I/O throughput. It is another reason that the three-operand instruction #0 reduces the number of the instruction #0 steps, and consequently, increases the ratio of the instruction #1.

Next, the effect on the reduction of the static steps is shown. Figure 4.13 represents the comparison of the number of the SIMD static steps between SuperENC and SuperENC-II. The static steps of the SIMD in the SuperENC is 4,031, which is almost full of the instruction memory with capacity of 4,096 words. In SuperENC-II, the static steps is reduced to 2,581, which is the 64 % of SuperENC. Especially, the reduction on prediction error image generation is remarkable. In SuperENC, four kinds of subroutines are used depending on whether the horizontal and/or vertical interpolations are carried out or not. These subroutines are unified by using the pixel interpolator in SuperENC-II. Moreover, the subroutines, which involve repetitive operations, also become compact owing to the loop in-

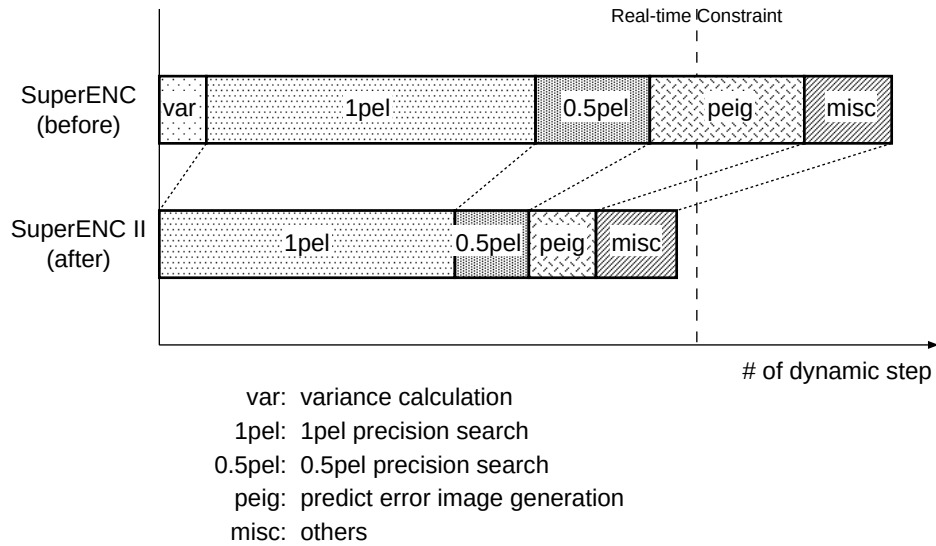


Figure 4.11: Comparison of the numbers of dynamic steps for each operation (B-pictures).

struction. Note that it arises from an advanced algorithm in SuperENC-II that the number of the static steps of the 1-pel precision search is increased.

4.5 Implementation

A chip photograph of the SuperENC-II, in which the improved SIMD is implemented, is shown in Fig. 4.14. The gate count of the SIMD is 117 K. In spite of the remarkable improvement as mentioned in Sec. 4.4, the size of the SIMD is reduced by a few percentage. This is due to the reduction on the number of the words in the PE register file by optimization of the instruction format. It turned out that the effect of reducing the register file words is more than the increase by addition of the specific execution hardware.

Specifications of the SuperENC-II is presented in Table 4.2. SuperENC-II can support Simple, Main, and 4:2:2 Profiles. With a single chip, Main Level can be encoded and multiple chips can encode High Level.

Coding tools supported by the flexibility of the SIMD are half-pel precision MC, forward/backward/bi-directional prediction MC, field/frame adaptive prediction MC, and 4:2:2 MC. Although almost all other ME/MC architectures also support half-pel precision MC and bi-directional prediction MC, there are few ME/MC architectures that can support field/frame adaptive MC and 4:2:2 MC.

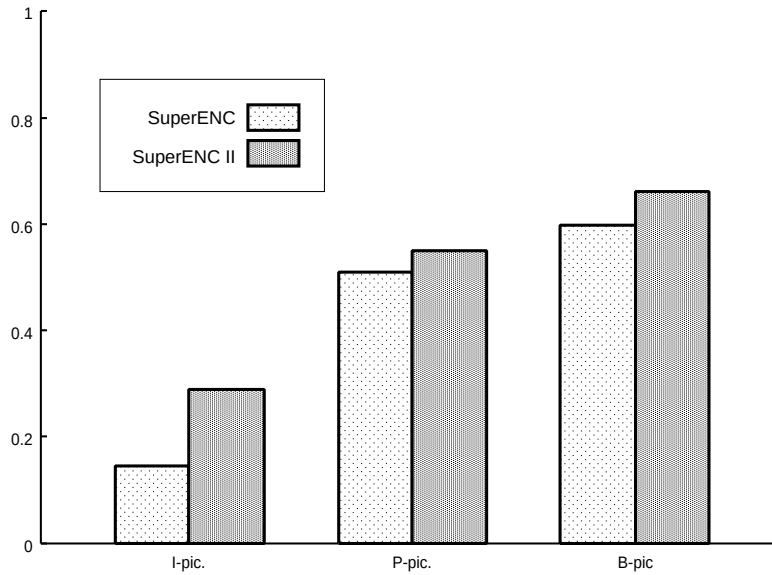


Figure 4.12: Comparison of the rate of PE operations.

These coding tools, as well as field/frame adaptive DCT decision, can be realized by the instruction level flexibility of the SIMD

Since many modules, as well as the SIMD, in SuperENC-II are improved, it is difficult to show a contribution of the SIMD to image quality enhancement quantitatively. Nevertheless, the whole of SuperENC-II can achieve a quite higher image quality compared with SuperENC.

4.6 Chapter summary

In this chapter, the SIMD macroblock processor in SuperENC and the improvement on the SIMD in SuperENC-II are described. By addition of specific execution hardware, improvement on the I/O throughput, and optimization and introduction of the instruction set, the SIMD can achieve a real-time performance in all kinds of pictures, even when more complicated algorithm is used. Moreover, the number of the static steps, the program size of the software, can be also reduced to 64 %, which enables modifications to algorithms in future. The SIMD is also implemented in a single chip MPEG-2 422P@HL codec LSI[26] and a full-duplex MPEG-2 codec LSI[15]. In the H.264/AVC encoder LSI described in the next chapter, the SIMD will be enhanced as the SME.

Instruction level flexibility of the SIMD enables to support wide-ranging

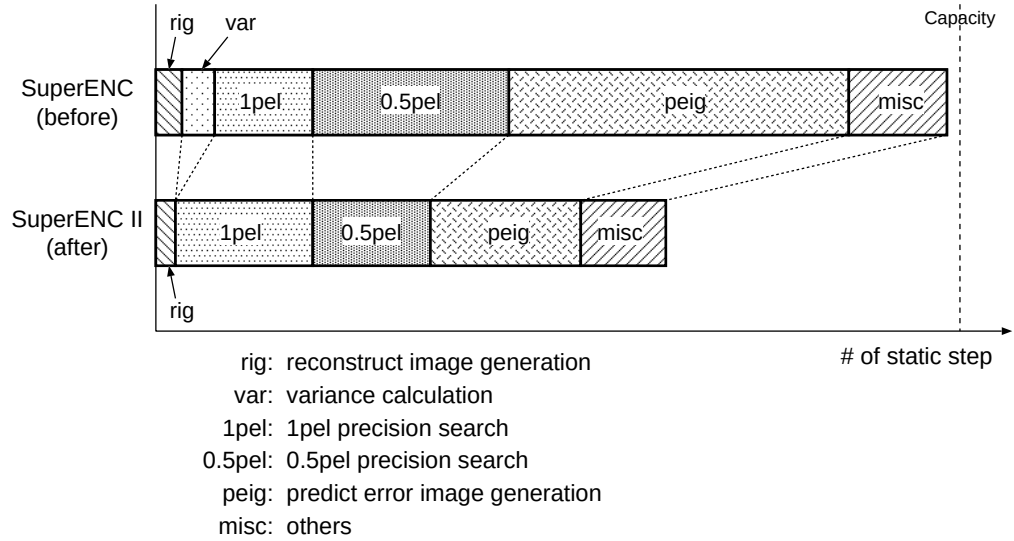


Figure 4.13: Comparison of the numbers of static steps.

ME/MC coding tools, such as half-pel prediction MC, bi-directional prediction MC, and field/frame adaptive MC. Moreover, by optimization of the hardware architecture, the performance is also improved, which leads to enhance image quality. It is also shown that the flexibility of the SIMD expands the functionality of the LSIs to 4:2:2 encoding.

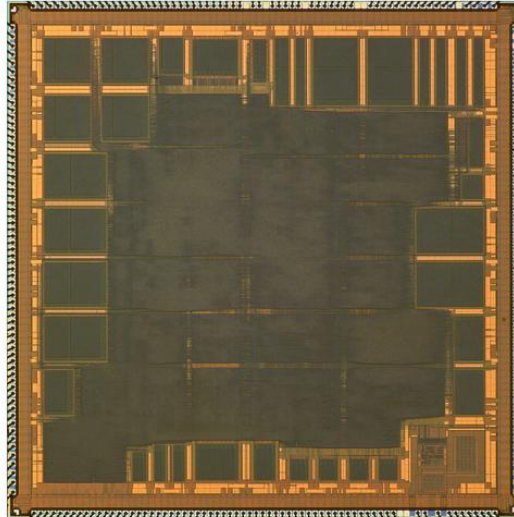


Figure 4.14: Chip photograph of the SuperENC-II.

Table 4.2: Specifications of the SuperENC-II.

Characteristic	Description
Process	0.18 μm four layered CMOS
Size	$7.5 \times 7.5 \text{ mm}^2$
# of Transistor	7.7 MTr.
Supply voltage	3.3 V (I/O), 1.5 V (Core)
Clock frequency	81 MHz
Power consumption	$\leq 1.0 \text{ W}$ (typically 800-900 mW)
Package	208-pin QFP
External memory	64 Mbit SDRAM $\times 1$ or $\times 2$
Video coding standard	MPEG-2
Profile	Simple/Main/4:2:2 Profile
Level	Main Level (a single chip) High Level (multiple chips)

Chapter 5

Thread level flexibility for H.264/AVC High422 Profile encoder LSI

5.1 Introduction

Due to transition from MPEG-2 to H.264/AVC, more flexibility of motion estimation and compensation (ME/MC) is required. Then, ME/MC hardware architecture with thread level flexibility is proposed in this chapter. The first layer of the MV search is decomposed into a unit search, called *thread*. Flexible ME searches can be realized by arranging combination of the threads. The architecture, which also has functional block level flexibility (Chapter 3) and instruction level flexibility (Chapter 4), is implemented in an H.264/AVC High422 Profile and MPEG-2 422 Profile encoder LSI, SARA.

The H.264/AVC[19] has been widespread over many applications, such as broadcasting, storage media, video conference, or web streaming. It will, especially, play an important role in the field of high definition TV (HDTV) broadcasting infrastructures. It is true that, in several countries including Japan and the United States, digital HDTV broadcasting services have already begun with use of MPEG-2[18] as video coding standard, and that the H.264/AVC seems to be used mainly in mobile broadcasting like Digital Video Broadcasting - Handheld (DVB-H)[7] in Europe, 1seg of Integrated Services Digital Broadcasting - Terrestrial (ISDB-T)[2] in Japan, and Advanced Television Systems Committee - Mobile/Handheld (ATSC-M/H)[1] in the US. Nevertheless, there are still many

professional HDTV applications that need H.264/AVC encoder systems because of their image quality and coding efficiency.

The applications in HDTV broadcasting infrastructures are classified into three types of categories, each of which requires different functions for H.264/AVC encoder systems. First, a contribution is an application to transmit video materials to or between broadcasting key stations. The video materials should be encoded in a higher bit rate and in the 4:2:2 chroma format in order to maintain the image quality even if they might be repetitively encoded and decoded during a production process. Second, in a relay broadcasting, low delay (latency) for encoding is indispensable for a live broadcasting. Third, a distribution, from broadcasting stations to TV receivers, needs to encode or transcode video data with high compression.

Consequently, the requirements for an H.264/AVC encoder LSI for professional use are quite different from the ones for a consumer encoder LSI. While a consumer LSI has to support only the 4:2:0 chroma format, the professional encoder LSI should support the 4:2:2 chroma format, as well as the 4:2:0 format. This yields that the 33% more memory bandwidth is needed. The encoding delay condition is very severe. In general, the relationship between the delay and the image quality is a trade-off for video encoder LSIs. Therefore, some advanced coding control mechanisms are required in order to realize both the low delay and the high image quality at the same time. Because of the connectivity to the existing MPEG-2 codec (encoder and decoder) systems in broadcasting infrastructures, the professional H.264/AVC LSI had better also support MPEG-2, and transcode video formats from H.264/AVC to MPEG-2, or vice versa. To achieve high compression ratio, a two-pass (tandem) encoding technology is often utilized. The professional LSI should be able to read the coding results of the first pass encoding, and reflect them in the second pass encoding. Moreover, it is taken as granted that the professional LSI can encode a variety of video scenes with high image quality in a wide range of bit rate, from low bit rate used for distribution to high bit rate for contribution. The professional LSI, thus, must have no “weak” scenes, even if they are, for example, fast moving, camera flash, scene cut, and fade-in/-out, and so on.

Several H.264/AVC encoder chips have already been developed[9, 8, 3, 30, 33, 5]. Almost all of them, however, are aimed at only consumer applications, and none of these chips have the functionality required for HDTV broadcasting. It is important for encoders to support various coding tools so as to fully extract H.264/AVC performance. The chips as described above, however, can support

limited coding tools in order to reduce the circuit size and power consumption. Although Huang et al.[9], Fujiyoshi et al.[8], and Chang et al.[3] early developed their H.264/AVC encoder chips, they can support only Baseline profile¹. High profile encoders were reported by Chang et al.[30] and Chen et al.[5], though they seem not to support interlaced video formats, say 1080/60i, which are often used in HDTV broadcasting. The chip by Mizosoe et al.[33], which also supports High profile and a transcoding function, was designed for digital video consumer applications, and does not support the 4:2:2 chroma format.

To meet the requirements for HDTV broadcasting infrastructures, it is indispensable for ME/MC hardware architecture to support coding tools newly introduced in H.264/AVC, in addition to support MPEG-2 encoding in Chapter 3 and Chapter 4. It is also required to have functionality to realize low-delay encoding, two-pass encoding or transcoding, besides 4:2:2 encoding. Thus, an H.264/AVC encoder LSI[10, 35, 43, 42], named “SARA”, has been developed that supports High422 profile, as well as 422 profile in MPEG-2. The SARA covers almost all of ME/MC coding tools specified in the H.264/AVC, such as multiple reference frames, variable block size, quarter-pel prediction, picture and macroblock adaptive field/frame prediction (PAFF/MBAFF), temporal and spatial direct mode, and weighted prediction (WP). In order to support a lot of ME/MC coding tools, the SARA contains powerful and flexible ME/MC engines that can search a wide search range with -217.75 to $+199.75$ horizontally and -109.75 to $+145.75$ vertically. It also has pre-analysis engines to realize various functions needed for HDTV broadcasting and to achieve high image quality. The SARA is the successor to the previous professional MPEG-2 422P@HL codec chip (VASA)[27, 41].

This chapter is organized as follows. Section 5.2 describes the system architecture of the SARA. The advanced coding control, which is a progressed version of the functional block flexibility in Chapter 3, is argued in the section. The ME/MC architecture is explained in section 5.3. After implementation results are mentioned in section 5.4, Section 5.5 shows some image quality evaluations. Then, section 5.6 summarizes this chapter.

¹Profiles are subsets of coding tools specified in the H.264/AVC standard. Baseline profile is the smallest coding tool subset, and High profile is the largest tool set. High422 profile is a 4:2:2 version of High profile.

5.2 System Architecture

Before going on to the details of ME/MC hardware architecture, a system architecture of the chip is explained. The system architecture provides a framework needed for a professional H.264/AVC encoder LSI: advanced coding control for high functionality, memory bandwidth reduction and memory bus architecture used by ME/MC architecture, and multiple chip configuration for HDTV encoding.

5.2.1 SARA architecture

Figure 5.1 shows a block diagram of the SARA architecture. The SARA consists of a 64-bit RISC processor (TRISC), two video coding cores (M-CORE and C-CORE), a video interface (VIF), pre-analysis engines (IR, MBP, RIT), a multiplexer (MUX), a multiple-chip data transfer (MDT), a memory interface (MIF), and embedded DRAMs (eDRAMs). Each of the M-CORE and the C-CORE has a 32-bit RISC processor (MRISC and CRISC, respectively). The M-CORE has triple ME/MC engines (TME, FME, and SME), an intra prediction (IPD), and a transform and quantization (TQ) as application-specific hardware blocks. An entropy coding (EC) and a loop filter (LF) are in the C-CORE.

The SARA is a multiple-processor system-on-chip (MPSoC). While the TRISC manages picture layer and the above, the MRISC and the CRISC handle slice layer and MB layer. Owing to this hierarchical RISC configuration, the MRISC and the CRISC can concentrate on controlling the hardware functional blocks in each core, and the hardware blocks can have a lot of encoding modes or parameters which are set up according to encoding conditions. This helps to inherit previous coding results in two-pass encoding or transcoding. Note that it is crucial to include some processors in a professional video encoder chip, though several developed chips have no processor inside[9, 4, 3], because encoder systems cannot always have a processor with enough performance outside of a encoder chip. Moreover, it is a more important reason that data on a bus between a processor and hardware functional blocks contain a lot of very confidential encoding know-how.

The SARA has powerful pre-analysis engines (IR, MBP, and RIT). The IR, image reduction, makes shrunk images used in the ME/MC engines. The macroblock preprocessing unit, MBP, can calculate statistical information of input video signals, process spatial and temporal filters, and detect scene cuts or fade

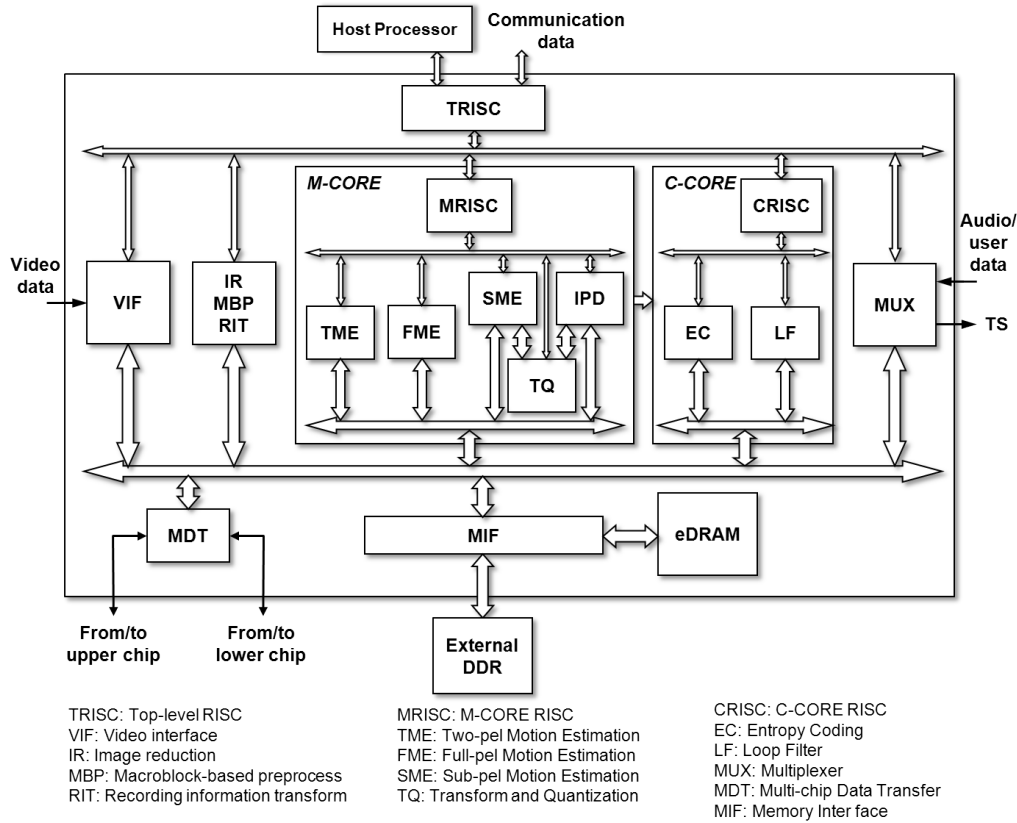


Figure 5.1: Block diagram of the SARA.

scenes. In case of two-pass encoding or transcoding, previous coding information, like picture types or motion vectors (MVs), are extracted by the RIT. Together with the three RISC processors (TRISC, MRISC, and CRISC), they can realize an “advance coding control” scheme (Fig. 5.2). Before encoding process, video signals and previous coding information, if any, are input into these pre-analysis engines, which output statistical information and previous coding modes to the TRISC. The TRISC make the MRISC and CRISC to control their hardware functional blocks with these information. Using this advance coding control scheme, the SARA can encode fade scenes with automatic weighted prediction, low-delay encoding by acquiring statistical information of video signals, two-pass encoding and transcode with inheriting previous coding modes.

A 4:2:2 chroma format support increases the memory bandwidth because chroma data, Cr and Cb, of 4:2:2 are double of those of 4:2:0. Although an embedded DRAM (eDRAM) can reduce external memory bandwidth itself, memory

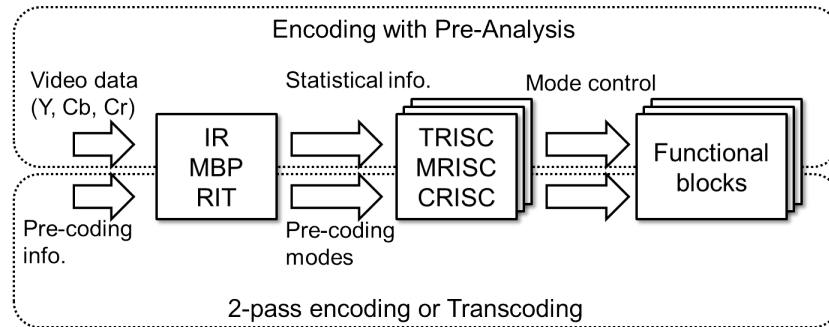


Figure 5.2: Advanced coding control scheme with pre-analysis engines.

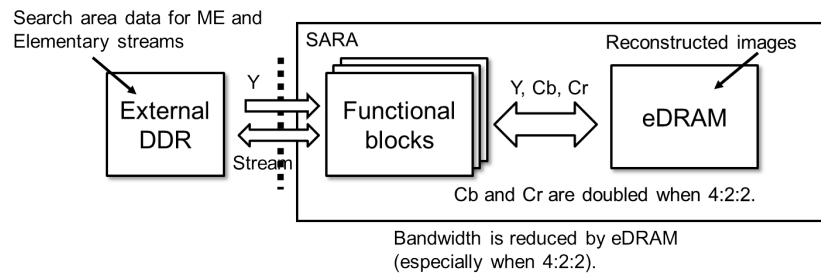


Figure 5.3: Memory mapping to reduce bandwidth when 4:2:2 encoding.

mapping is significant for the reduction (Fig. 5.3). Search area data for ME are mapped into an external double data rate SDRAM (DDR-SDRAM), because no Cb and Cr data are needed. Reconstructed images are mapped into the eDRAM, which are used as reference pictures in MC. This mapping can reduce external memory bandwidth especially when 4:2:2 encoding.

Besides memory bandwidth reduction, it is also essential to use a data bus efficiently. the SARA adopts an original data bus protocol in order to make the rate of data bus usage higher. Unlike a general data bus protocol, such as Open Core Protocol (OCP) bus, the MIF is a master and each functional block is a slave in the original protocol. The MIF has a sequence program that contains an order set of data transfer kinds. During some data transfer, it makes a request to the functional block involved in the next data transfer. The functional block responses a parameter set needed for the next transfer. When it receives the parameter set, the MIF calculates a physical address of external SDRAM or eDRAM from the parameter set, and carries out memory read or write transactions. A preliminary experiment shows that this protocol can increase the average ratio of active bandwidth to 67 - 70%.

It is notable that the SARA has both content adaptive variable length coding (CAVLC) and content adaptive binary arithmetic coding (CABAC) as entropy coding (EC) of the H.264/AVC. While it can compress more than CAVLC, CABAC is not suitable for a higher bit rate because of its essentially serial operation. As described above, an H.264/AVC encoder chip can be used with wide range of bit rate in professional applications. This is the reason why the SARA has both entropy coding tools.

All functional blocks in M-CORE and C-CORE, except for LF, have an MPEG-2 mode so as that the SARA can support MPEG-2 encoding.

5.2.2 HDTV configuration

While HDTV becomes more and more popular, there are still quite a few countries where broadcasting is done mainly with standard definition television (SDTV) resolution. This is why the SARA was designed as an SDTV video encoder chip. With a multiple-chip configuration, it can encode full HDTV video data ($1,920 \times 1,080$ pixels, 60 field/sec). An HDTV configuration is depicted in Fig. 5.4. An input video image is divided into several regions by horizontal lines, and each region is encoded by the SARA. All output streams are gathered into one of the SARAs, and multiplexed into a single stream.

To realize this HDTV configurations, some blocks have extended functions. The VIF can receive the whole HDTV signals and select and store only the divided region of the chip. The MUX can multiplex streams generated by several SARAs [41]. With this configuration, a picture is encoded into several slices². The MDT can transfer reference image data between chips so that the ME/MC architecture can search MVs beyond slice boundaries. Deblocking filter by the LF can also be executed over all of a picture.

5.3 ME/MC Architecture

5.3.1 ME/MC algorithm

The proposed motion estimation and compensation (ME/MC) algorithm is based on the one of [45], and extended and optimized for the H.264/AVC (Fig.5.5). It is comprised of a four-layer hierarchical search. The first layer applies a combination of a telescopic search (TS) and a direct search (DS) with two-pel precision.

²In H.264/AVC, slice means a set of continuous MBs or MB lines.

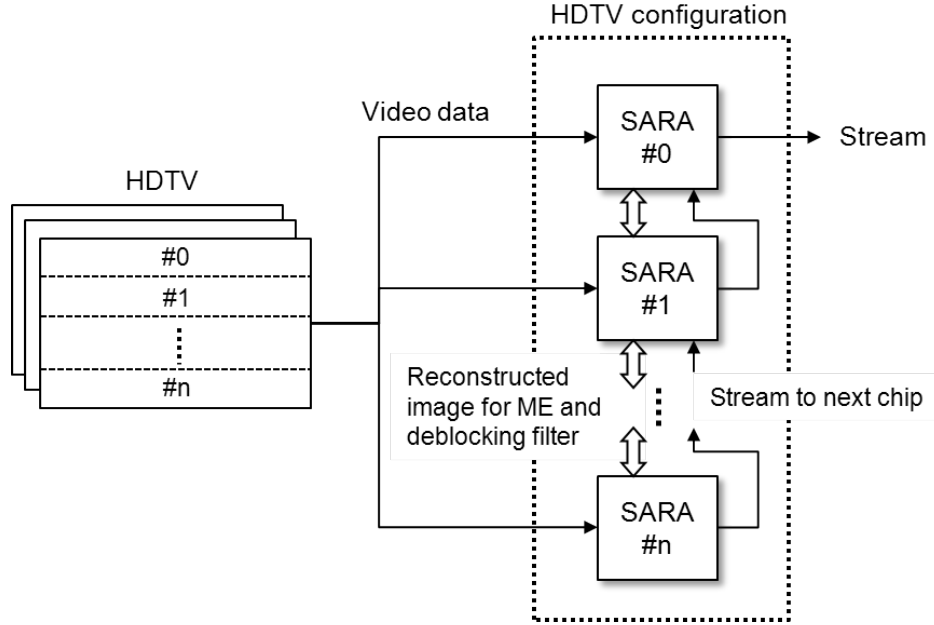


Figure 5.4: HDTV configuration.

While it can realize a wide search range, $d^2 \times a$, where d is the distance from the current picture to the reference picture, and a is a range of one-step of a telescopic, the TS might be misled to a local minimum MV in case of, for example, camera flash scenes. Then, several direct searches between the current picture and the reference pictures are added for reinforcement. The range of each DS is also a , centering some points such as (0,0), predicted motion vectors (PMVs), or the MVs of the previous coding in case of transcoding or two-pass encoding. The second to fourth layers are full searches with ± 1 -pel, ± 0.5 -pel, and ± 0.25 -pel ranges, respectively, centering the MVs obtained from the upper layers. While the first and second layers search MVs of four 8×8 blocks in a MB, the third and fourth layers evaluate MVs of all block sizes supported (8×8 , 8×16 , 16×8 , and 16×16) obtained from the 8×8 MVs.

In this ME/MC algorithm, all ME/MC coding tools required for broadcasting application are covered³. A support of multiple reference frames is straightforward because the TS itself searches MVs with multiple pictures regardless of reference or non-reference. Variable block size and spatial/temporal direct mode is

³The ME/MC algorithm does not support MC with smaller block size than 8×8 , such as 4×4 , 4×8 and 8×4 . The MC with these block sizes is aimed at lower definition than SDTV, and it has almost no influence on encoding of SDTV or higher definition.

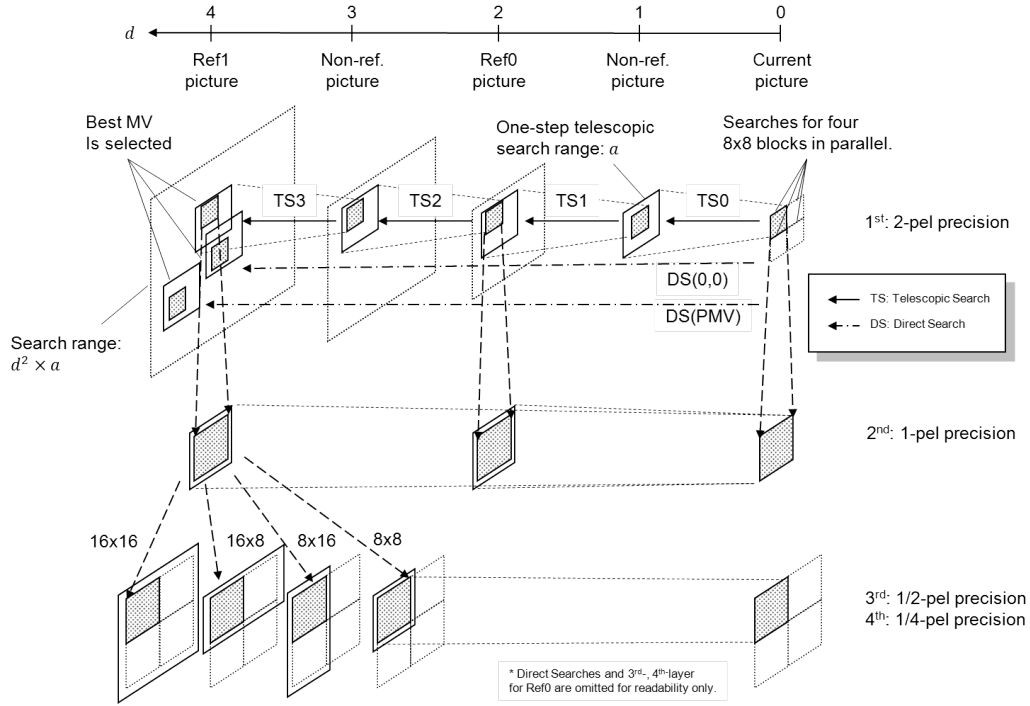


Figure 5.5: ME/MC algorithm used in SARA.

supported at the third and fourth layers, while MBAFF is evaluated at the first and second layers. For all layers, weighted reference pictures can be used in weighted prediction mode.

5.3.2 Two-pel motion estimation architecture

The ME/MC architecture consists of three ME/MC engines, TME, FME, and SME. The TME executes two-pel ME at the first layer of the algorithm, the FME is responsible for the second layer, and the third and fourth layers are mapped into the SME. Among these three engines, the TME and the SME has flexibility. The SME, which is an enhancement of the SIMD in Chapter 4, has instruction level flexibility in order to support variable block size MC, quarter-pel prediction, spatial/temporal direct mode[42]. This section focuses on the TME architecture, which new thread level of flexibility is introduced into. It is described how the TME architecture achieves a wider search area, high precision MVs, and flexibility to satisfy the requirement of the professional encoder.

The architecture of the TME is shown in Fig.5.6. The TME is controlled

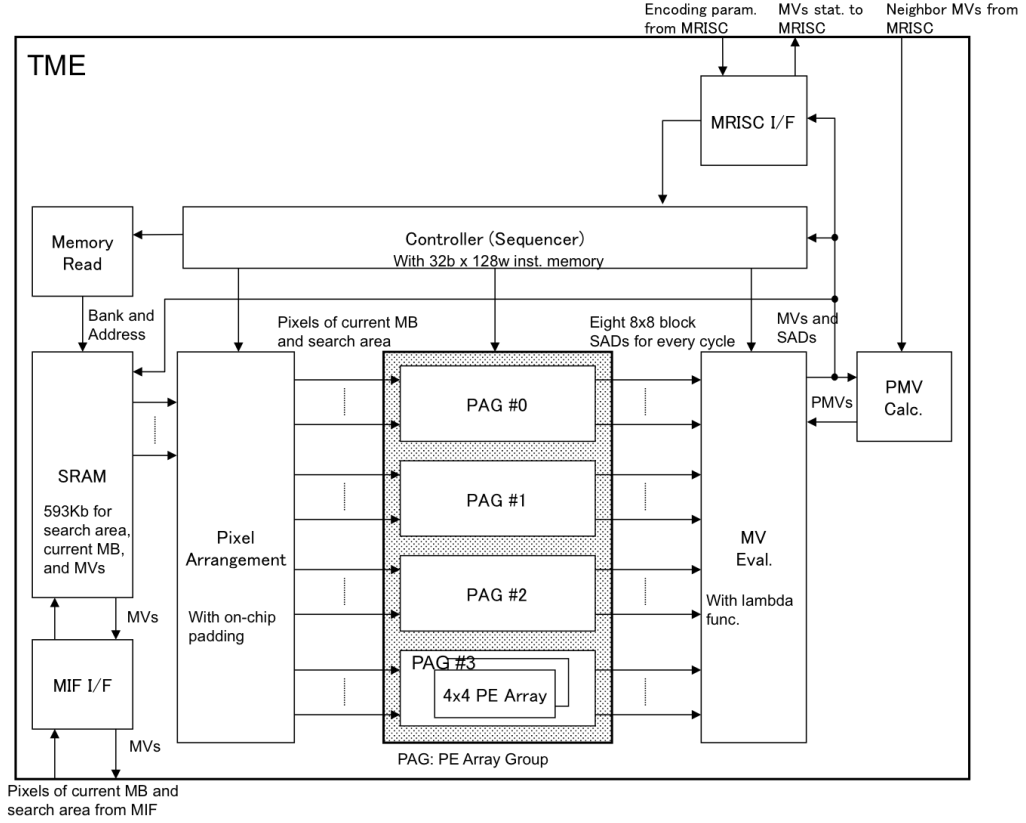


Figure 5.6: The TME architecture.

by a controller with a sequence program. The current MB data and the reference picture data are read from memory, arranged with on-chip padding for unrestricted MVs, and fed into processing element array groups (PAGs). The PAGs outputs MVs, which are evaluated with lambda functions using predicted motion vectors (PMVs).

Four PAGs are usually corresponded with four 8×8 blocks in a MB, respectively, and make MV searches in parallel⁴. Two types of parallelism are additionally introduced in each of the PAGs to widen a search range (Figure 5.7). First, in order to make a one-step search range, a , double, each of the PAGs has twin 4×4 processing element (PE) arrays that search left and right halves of the search range. This makes the range a to $-12/+11$ horizontally and $-3/+4$ vertically on 2-

⁴As described later, the mapping of the current MB into four PAGs is variable by thread level flexibility.

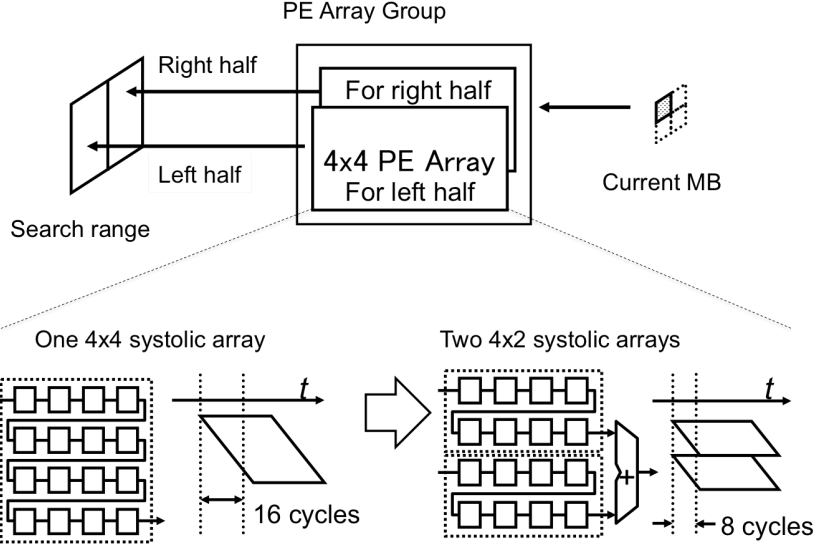


Figure 5.7: Two types of parallelism introduced in the PE array group.

pel precision field images⁵. It corresponds to $-24/+22$ horizontally and $-12/+16$ vertically in 1-pel frame image precision. Secondly, a 4×4 systolic array in a PE array is divided into two 4×2 systolic arrays. It takes 16 cycles from the start of a one-step search for a 4×4 systolic array to output the first sum of absolute difference (SAD). The next step search cannot start until the previous search results are fixed in the telescopic search. Two 4×2 systolic arrays can make the start-up cycles half. As a result, the distance from the current picture to the farthest reference picture, d , is increased from seven to nine. Due to the two types of parallelism, the search area becomes $-216/+198$ horizontally and $-108/+144$ vertically. Preliminary experiments show that this search area is wide enough to encode usual HDTV scenes.

The TME, as well as the other two ME/MC engines, evaluates MVs, mv , with a following MV cost function, $mvcost$, as the reference software JM,

$$mvcost = SAD + \lambda (|mv_x - pmv_x|) + \lambda (|mv_y - pmv_y|) \quad (5.1)$$

where, SAD is a sum of absolute difference, λ is the Lagrange multiplier and pmv is a predicted motion vector, a median of the MVs of the neighbor (upper,

⁵Asymmetry of the search range arises from that the search range a is asymmetric. The horizontal range of a is a multiple of two because of the twin PE arrays in a PAG. The vertical range comes from implementation issues.

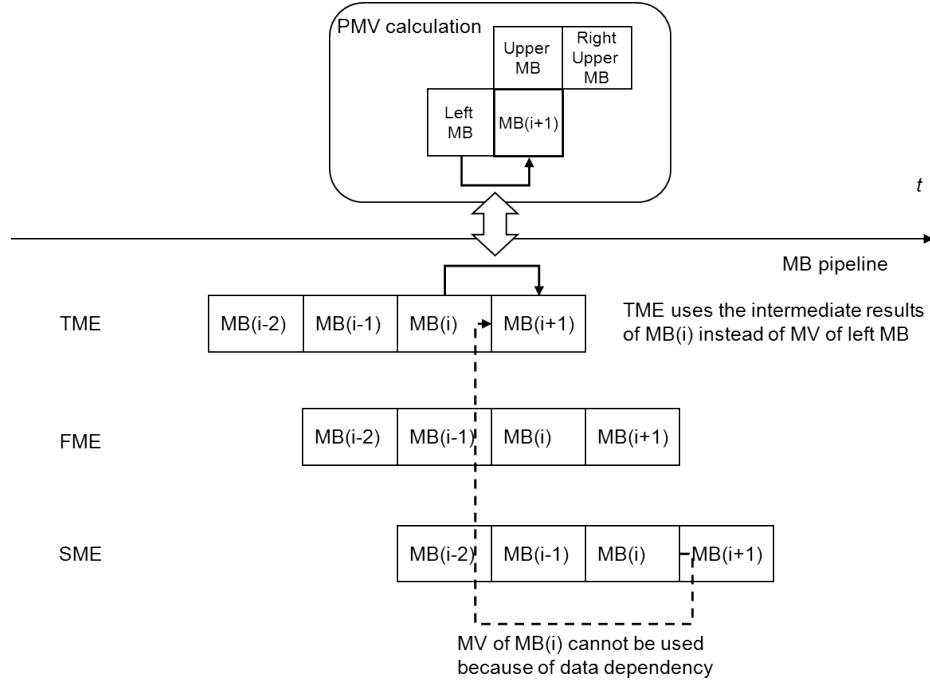


Figure 5.8: Approximation of predicted motion vector.

right upper, and left) MBs. The three engines work in a MB pipeline manner⁶. When the TME is processing the i -th MB, the FME is at $(i - 1)$ -th MB, and the SME is at $(i - 2)$ -th MB, respectively. Thus, a precise *pmv* cannot be used by the TME because the MV of the left MB is not yet decided. To solve this problem occurred by pipeline architecture, Wang et al.[48] utilized MVs of the left upper MB instead of the left MB. It is true that this method can get approximate *pmv* for almost all MBs except for the MBs in the first MB line in the slice, but it is a critical issue in case of the SARA because the HDTV configuration (Fig.5.4) has a lot of those MBs in the first MB lines in the slices. Therefore, the TME uses, as the MVs of the left MB, the results of the $(i - 1)$ -th MB calculated by the TME itself (Fig.5.8). By this technique, the TME can find high precision MVs.

The TME has a sequence program in the controller. The searches by the TME can be decomposed into a unit search, called *thread*, with the search range a . A thread is instructed with a set of parameters, as shown in Fig.5.9, the mapping of the current MB into four PAGs, the picture number to be searched and a search center point, and so on. The sequence program, an ordered set of the

⁶In MBAFF mode, they work in a MB-pair pipeline manner, instead of MB pipeline.

sm	tm	bs	pn	cv	lf
----	----	----	----	----	----

sm: search method (telescopic/direct)
tm: template mapping pattern
bs: block size (8x8, 16x8, 8x16, 16x16, etc.)
pn: picture number
cv: center vector (previous search result, pmv, (0,0), etc.)
lf: last instruction flag

Figure 5.9: Instruction of the TME sequencer.

thread instructions, can realize flexible searches, such as a combination of the TS and DS, various distance d according to the M values⁷, field/frame searches, forward/backward searches. As a search center, the origin vector (0,0) and/or approximated pmv is usually utilized. In case of two-pass encoding or transcoding, the MV extracted by the RIT is also used as a center of a direct search to inherit from the previous encoding.

Thread level flexibility is explained by examples below. A sequence program in Figure 5.5 is a series of TS0, TS1, TS2, TS3 for a telescopic search, and DS(0,0) and DS(PMV) for direct searches. In this case, the sequence program consists of six times of thread searches. When eight times of thread searches can be executed for a MB and a current picture including the MB has multiple reference pictures as shown in Figure 5.10, four threads of telescopic searches, TS0, TS1, TS2 and TS3, can find MVs for Ref0, Ref1, Ref2. Then, four threads of direct searches, $DS^0(0,0)$, $DS^0(PMV)$, $DS^2(0,0)$, and $DS^2(PMV)$, can enhance the MVs for Ref0 and Ref2. If the reference picture Ref0 has less correlation with the current picture because of flash, two threads of direct searches to Ref0 can be changed to Ref1, $DS^1(0,0)$, and $DS^1(PMV)$ by a sequence program. When a scene cut exists between the current picture and the reference picture Ref2, all eight threads can be used in one direction searches, and the search range a can be widened to $4a$ by mapping a 8×8 block into four PAGs. Widening search range is also utilized in a scene with very fast motion. Although the normal search range $d^2 \times a$ is wide enough for usual HDTV scenes, there are, in fact, rare scenes that have a faster move than expected. The chip finds out the scenes with pre-analysis, and make the TME widen the search range to $4a$ in order to find appropriate MVs.

Thus, thread level flexibility can support a generalized bi-prediction MC and multiple reference MC, which are newly introduced in H.264/AVC. The flexibility

⁷The M value means the frame number between I-, or P-pictures.

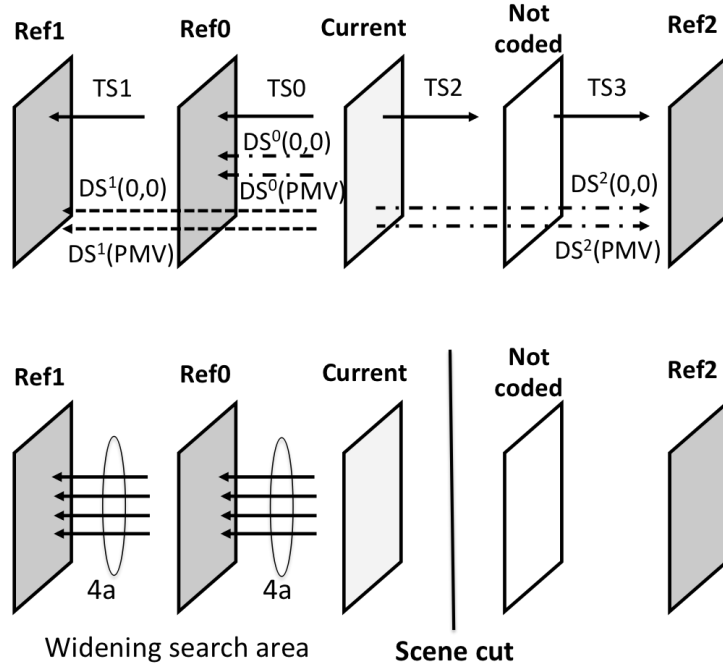


Figure 5.10: Examples of thread level flexibility. Upper: In the case that Ref0 has less corelation with the current picture. Lower: In the case that a scene cut exists between the current picture and Ref2.

also play an important role on functional expandability of the chip, in the case of two-pass encoding and transcoding for professional use.

5.4 Implementation

A microphotograph of the SARA is shown in Fig. 5.11, and the chip specifications are summarized in Table 5.1. It was successfully fabricated in a 90 nm technology. The chip integrates 140 million transistors. The chip can encode SDTV (720×480 , 60 fields/sec) in real time. With a multiple-chip configuration on a post-card size module (Fig. 5.12), it can encode full HDTV ($1,920 \times 1,080$, 60 field/sec). The maximum transport stream rate is 160 Mbps. The coding delay, including both encoding and decoding time⁸, is 800 msec with an $M=3/N=15$

⁸The coding delay was evaluated using the SARA encoder equipment and the SARA/D decoder equipment.[25]

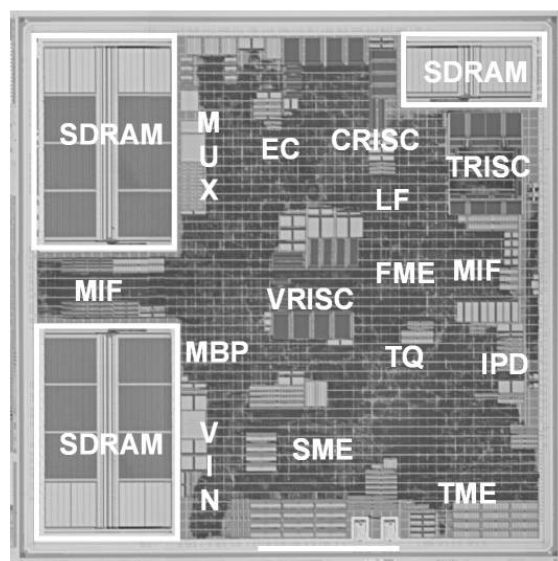


Figure 5.11: Microphotograph of the SARA.

group of pictures (GOP) structure⁹, and can be shortened as much as 300 msec with an all-P-pictures structure and a well-known cyclic intra refresh technique. The SARA supports High422 profile (8bit only) of H.264/AVC and 422 profile of MPEG-2. All of coding structures, field, frame, PAFF, MBAFF, are supported. A search range of ME is $-217.75/+199.75$ horizontally, and $-109.75/+145.75$ vertically. All ME/MC coding tools required for broadcasting applications are supported, such as multiple reference frames, variable block size, quarter-pel prediction, spatial/temporal direct mode, and weighted prediction. Broad coverage of the H.264/AVC coding tools, together with pre-analysis engines means the high potential for a professional encoder chip.

5.5 Evaluations

To evaluate performance of the ME/MC architecture, image quality experiments were made. Fig. 5.13 shows a image quality comparison between the SARA and the JM, a reference software of the H.264/AVC. Eight scenes of $1,440 \times 1,080$ pixel and 450 frames from the Institute of Image Information and Television Engineers (ITE)/Association of Radio Industries and Businesses (ARIB) Hi-Vision test sequences were encoded in the 4:2:0 chroma format at 6 Mbps, and decoded

⁹N is the number of frames between I-pictures.

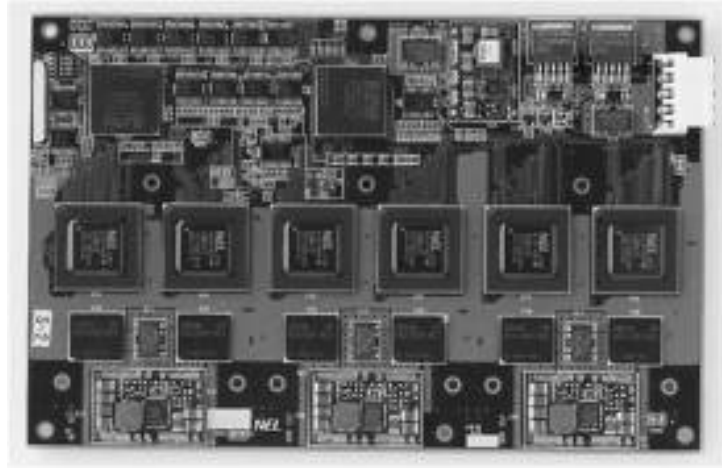


Figure 5.12: The SARA HD module.

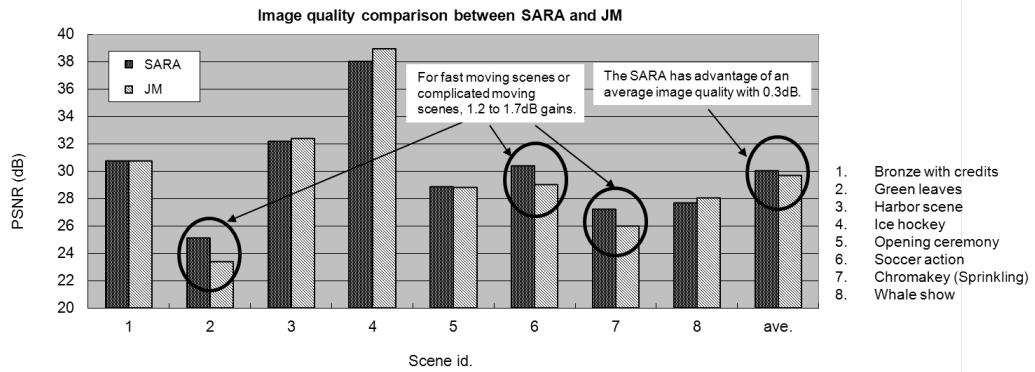


Figure 5.13: Image quality comparison between SARA and JM.

images were evaluated in average peak signal-to-noise ratio (PSNR). The search algorithm and the search range of the proposed ME/MC architecture are as described in Sec. 5.3 and Sec. 5.4. The JM uses a full search, and the search range is ± 32 horizontally and vertically around a predicted motion vector. It is found that, for almost all scenes, the SARA has competitive performance. Especially, for fast moving scenes or complicated motion scenes, like scene id 2, 6, and 7, SARA takes 1.2 to 1.7 dB gain compared to the JM. This is because the ME/MC architecture can find better MVs and the pre-analysis engines helps an adequate rate control with the advanced coding control scheme described in Sec. 5.2. The SARA also has advantage of an average image quality with 0.3 dB.

In HDTV broadcasting applications, there are faster moving scenes than the

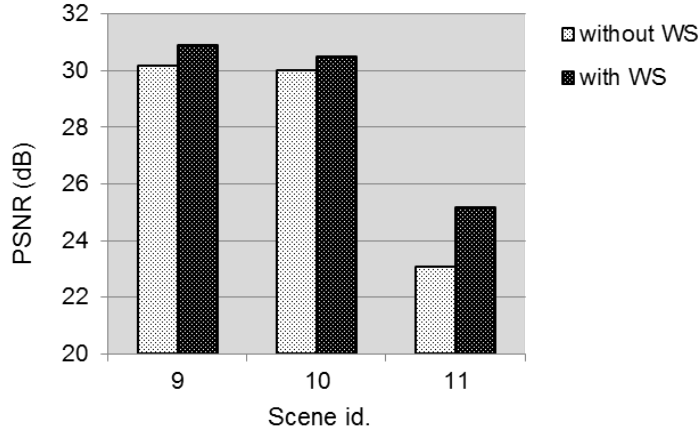


Figure 5.14: Adaptive widening search area scheme.

above test sequences to encode. When the SARA detects such scenes from statistics of MVs, it can change the M value to 1 and widen the search range to $4a$ as described in Sec.5.3. Figure 5.14 shows the image enhancement by this adaptive widening search area (WS) scheme. Some test scenes with $1,920 \times 1,080$ are encoded at 8 Mbps with and without the WS scheme. Scene id 9 and 10 are sports scenes with faster moves horizontally (panning) and/or vertically (tilting). Scene id 11 is an artificial scene for motion vector search evaluation. It combines two different scenes, upper half of image is from one scene and lower half is from the other. Each half has a fast move of different direction. It is found that coding gain is 0.48 dB to 2.1 dB.

Figure 5.15 shows an effect on fade scenes by automatic weighted prediction (WP) using the advanced coding control scheme. If pre-analysis engines detect a fade scene, the difference of average values of luminance in successive pictures can be utilized as an explicit weight. In these graphs, the x -axis shows the number of pictures, and the y -axis shows average values of luminance (Y) in decoded pictures. Without WP, the lines of fade-in and fade-out are jagged, which degrades subjective evaluations. With the automatic WP proposed in Sec. 5.2.1, the lines of fade-in and fade-out become smooth. Thus, the ME/MC architecture can encode a variety of scenes efficiently.

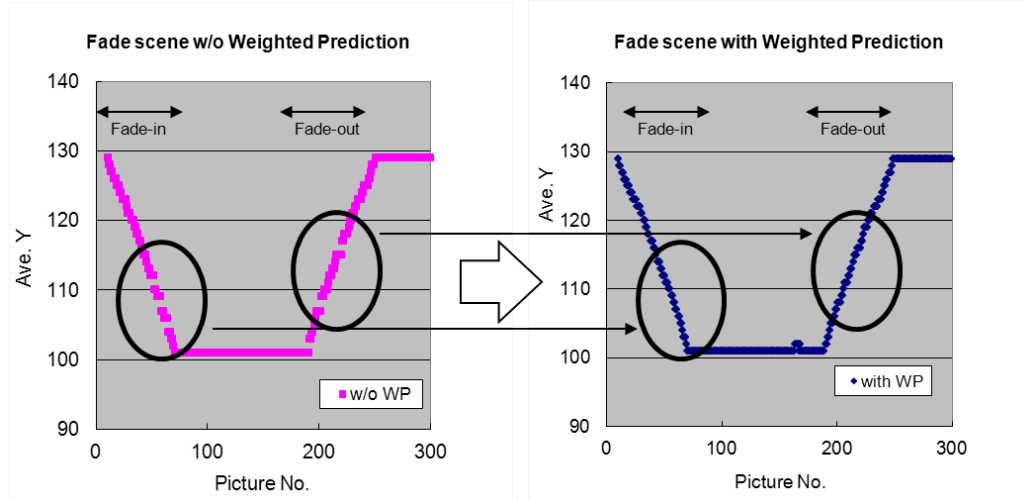


Figure 5.15: Fade scene with or without automatic weighted prediction.

5.6 Chapter summary

A powerful and flexible ME/MC hardware architecture is proposed which is implemented in an H.264/AVC and MPEG-2 encoder LSI for HDTV broadcasting infrastructures. It can realize wide search range, multiple reference MC, variable block size MC, quarter-pel MC, spatial/temporal direct mode, and weighted prediction. Thread level flexibility of the TME, together with functional block level flexibility between pre-analysis engines and ME/MC engines and instruction level flexibility of the SME, realizes supports for wide ranging ME/MC coding tools, and also contributes functional expandability of the chip, such as 4:2:2 encoding, low-delay encoding, two-pass encoding, and transcoding.

The SARA chips are compactly mounted on a post-card size HD module, which enables to build up various codec equipments. Besides a simple (one-passed) encoder (Fig.5.16), a transcoder with a MPEG-2 decoder can be used for re-transmission services of digital terrestrial broadcasting over IP network, a two-pass encoder with two HD modules can be developed as a high compression encoder, and a full HD real-time 3D encoder using MVC, specified in Annex H of [19], can also be developed [28]. The chip with the proposed ME/MC hardware architecture is a key device for implementing various professional H.264/AVC and MPEG-2 applications for broadcasting infrastructures.



Figure 5.16: HDTV H.264/AVC encoder equipment using the SARA chips.

Table 5.1: Specifications of the SARA.

Technology		90-nm CMOS Technology
Number of transistors		140 million transistors
Clock frequency		200 MHz
Supply voltage		Core: 1.2 V, eDRAM: 2.5 V, I/O: 3.3 V
Power consumption		3.0 W
Package		625-pin FCBGA (21 mm × 21 mm)
Memory		eDRAM: 72 Mbit external: 512 Mbit (32-bit width)
Video	Profile	H.264: Main / High / High422, MPEG-2: Main / 422P
	Level	H.264: 3.0 / 4.0 [†] / 4.1 [†] , MPEG-2: ML / H14L [†] / HL [†] [†] : With HDTV configuration.
	Resolution and video rate	Single chip: 720 × 480, 60 field/sec Multiple chip: 1,920 × 1,080, 60 field/sec
	Coding structure	Field, Frame, PAFF, MBAFF
	Motion estimation	Search range: –217.75/+199.75 (H), –109.75/+145.75 (V) Max. number of reference frames: 4 Supported block size: 8 × 8, 8 × 16, 16 × 8, 16 × 16 Direct mode: spatial, temporal Weighted prediction mode: explicit
	Pre-processing	Adaptive spatial and temporal filter Macroblock-based feature extraction
Audio	Input	Linear PCM, AAC stream
System	Output	MPEG-2 TS, Max. 160 Mbps

Chapter 6

Concluding remarks

This dissertation has discussed ME/MC hardware architecture in video encoder LSIs. Video encoder LSIs are required to encode high image quality for any scenes, and to facilitate to realize high functionality, especially for professional use. The main focus of this dissertation was on investigating flexibility of ME/MC architecture in order to meet the requirements. Hierarchy of flexibility has been considered; functional block level, instruction level, and thread level.

At first, functional block level flexibility has been introduced into ME/MC hardware architecture in SuperENC, MPEG-2 MP@ML and 422P@ML video encoder LSI. It is called Flexible Communication Architecture (FCA). The scene-adaptive algorithms were considered in order to enhance image quality. The FCA enables to implement the scene-adaptive algorithms, such as the area hopping method and the adaptive f_code setting. It was showed that subjective and objective image quality has been enhanced.

Secondly, instruction level flexibility has been investigated. Video coding standards have been extending ME/MC technologies. As a result, video encoder LSIs should support as many coding tools as possible, so as to attain the full coding efficiency. To achieve this aim, a SIMD macroblock processor has been proposed in SuperENC, and it has been optimized from the perspective on computational performance in SuperENC-II. The proposed instruction-level flexible architecture can realize to encode all kinds of picture, I-picture, P-picture, and B-picture, as well as various MC related operations and 4:2:2 encoding, in real time.

Finally, ME/MC hardware architecture in SARA, an H.264/AVC Hig422 Profile and MPEG-2 4:2:2 Profile encoder LSI, has been discussed. Adding to the above two levels of flexibility, thread-level flexibility has been proposed in order to

realize new coding tools, such as multiple reference MC. Through objective evaluations, it was found that, for almost all scenes, the image quality is competitive with the H.264/AVC reference software, JM. SARA has also high functionality needed for professional uses: multiple chip configuration for high resolution encoding, transcoding support by inheriting various coding modes, 4:2:2 encoding for contribution application. The ME/MC hardware architecture with the three levels of flexibility plays an important role in realizing all of these professional functions.

It is noted that hardware architecture of video encoder LSIs has been closely related to that of the cutting-edge CPU and other LSI technologies. While the concept of SIMD was proposed by Flynn in 1966, SIMD instruction extension was first implemented in UltraSPARC microprocessor in 1995. It was only two years ahead of SuperENC, which used a SIMD macroblock processor for ME/MC architecture. The SIMD macroblock processor adopted an instruction set of very long instruction word (VLIW), which is a processor architecture with instruction set parallelism. VASA[27] introduced multi-core architecture, which is now a common technology for CPUs. An embedded DRAM, utilized in SARA, is one of the technologies that can reduce memory bandwidth. This relationship will be supposed to be continued in the future.

Video formats are expanding in several aspects: resolution, frame rate, color space, bit depth, and the number of viewpoints. For example, 4K UHD TV[22] has $3,840 \times 2,160$ pixel per frame, four times as high resolution as HDTV, and 59.94 frame per second, twice as high frame rate as HDTV. It also has a wider color space and includes a bit depth of 10 bits per sample. In encoding these emerging video formats, more and more computational performance will be required. Whereas, video coding standards are increasing coding tools and coding modes to be selected in encoding, as the latest standard, HEVC. Applications using video encoding are being spreading. The specialized way of video encoding according to applications, for example, low latency, high efficiency, or low power, must be required. These will require more flexible operations in video encoding. Thus, in video encoder LSI, both high performance and high flexibility will be needed. To achieve high performance, parallel encoding architecture will be advanced. However, parallel encoding yields to degradation of decoded image quality. It will be an issue in the future how parallel encoding architecture is designed which enhances image quality. In order to settle the parallel encoding issue, flexibility, a subject of this dissertation, will also play more important role on video encoding.

This dissertation attempts to reveal how flexibility can work for ME/MC hard-

ware architecture in video encoder LSIs that can provide high image quality and high functionality. While the works presented here are attempts in illuminating a path toward ME/MC architecture that makes full use of coding efficiency and functionality, these are meant to be foundation stones for prospective paradigm shift on hardware architecture of future video encoder LSIs.

Bibliography

- [1] Advanced Television Systems Committee, Inc. (ATSC), “ATSC-mobile DTV standard, part 1 - ATSC mobile digital television system,” Document A/153 Part1:2011, June 2011.
- [2] Association of Radio Industries and Businesses (ARIB), “Transmission system for digital terrestrial television broadcasting,” ARIB STD-B31, Nov. 2005.
- [3] H.-C. Chang, J.-W. Chen, C.-L. Su, Y.-C. Yang, Y. Li, C.-H. Chang, Z.-M. Chen, W.-S. Yang, C.-C. Lin, C.-W. Chen, J.-S. Wang, and J.-I. Quo, “A 7mW-to-183mW dynamic quality-scalable H.264 video encoder chip,” in *Solid-State Circuits Conference, 2007. ISSCC 2007. Digest of Technical Papers. IEEE International*, Feb. 2007, pp. 280–603.
- [4] T.-C. Chen, S.-Y. Chien, Y.-W. Huang, C.-H. Tsai, C.-Y. Chen, T.-W. Chen, and L.-G. Chen, “Analysis and architecture design of an HDTV720p 30 frames/s H.264/AVC encoder,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 16, no. 6, pp. 673 – 688, June 2006.
- [5] Y.-H. Chen, T.-D. Chuang, Y.-J. Chen, C.-T. Li, C.-J. Hsu, S.-Y. Chien, and L.-G. Chen, “An H.264/AVC scalable extension and high profile HDTV 1080p encoder chip,” in *VLSI Circuits, 2008 IEEE Symposium on*, June 2008, pp. 104 – 105.
- [6] M. Endo, J. Naganuma, Y. Nakajima, and T. Ogura, “An MPEG-2 encoding PC card system for real-time mobile applications,” in *Proceedings of International Conference on Consumer Electronics 2001 (ICCE2001)*. IEEE, Jan. 2001, pp. 160–161.

- [7] European Telecommunications Standards Institute (ETSI), “Digital video broadcasting (DVB); transmission system for handheld terminals (DVB-H),” ETSI EN 302 304, Nov. 2004.
- [8] T. Fujiyoshi, S. Shiratake, S. Nomura, T. Nishikawa, Y. Kitasho, H. Arakida, Y. Okuda, Y. Tsuboi, M. Hamada, H. Hara, T. Fujita, F. Hatori, T. Shimazawa, K. Yahagi, H. Takeda, M. Murakata, F. Minami, N. Kawabe, T. Kitahara, K. Seta, M. Takahashi, Y. Oowaki, and T. Furuyama, “A 63-mW H.264/MPEG-4 audio/visual codec LSI with module-wise dynamic voltage/frequency scaling,” *Solid-State Circuits, IEEE Journal of*, vol. 41, no. 1, pp. 54 – 62, Jan. 2006.
- [9] Y.-W. Huang, T.-C. Chen, C.-H. Tsai, C.-Y. Chen, T.-W. Chen, C.-S. Chen, C.-F. Shen, S.-Y. Ma, T.-C. Wang, B.-Y. Hsieh, H.-C. Fang, and L.-G. Chen, “A 1.3TOPS H.264/AVC single-chip encoder for HDTV applications,” in *Solid-State Circuits Conference, 2005. Digest of Technical Papers. ISSCC. 2005 IEEE International*, Feb. 2005, pp. 128 – 588.
- [10] 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,” in *Symposium on High Performance Chips (HOT CHIPS 19)*, Aug. 2007.
- [11] 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.
- [12] 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,” in *Design Automation and Test in Europe Conference 1999*. IEEE, Mar. 1999, pp. 44–50.
- [13] —, “Three-layer cooperative architecture for MPEG-2 video encoder LSI,” *IEICE Transactions on Electronics*, vol. E80-C, no. 2, pp. 170–178, Feb. 2000.
- [14] 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,” in *Proceedings of the European Design & Test Conference 1996 (ED&TC)*. IEEE, Mar. 1996, pp. 320–326.
- [15] M. Inamori, H. Iwasaki, T. Onishi, M. Ikeda, J. Naganuma, and Y. Yashima, “New set-top box for interactive visual communication of home entertainment using MPEG-2 full-duplex codec LSI,” *IEEE Transactions on Consumer Electronics*, vol. 51, no. 2, pp. 640–641, May 2005.
- [16] 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.
- [17] —, “ISO/IEC 14496-2: Information technology — Coding of Audio-Visual Objects — Part 2: Visual,” Dec. 2000.
- [18] 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.
- [19] —, “ITU-T Recommendation H.264 and ISO/IEC 14496-10: Information technology — Coding of Audio-Visual Objects — Part 10: Advanced Video Coding,” Mar. 2003.
- [20] —, “ITU-T Recommendation H.265 and ISO/IEC 23008-2 MPEG-H part 2: High efficiency video coding,” Apr. 2013.
- [21] ISO/IEC JTC1/SC29/WG11, “Test model 4,” Apr. 1993.
- [22] ITU-R, “ITU-R Recommendation BT.2020: Parameter values for ultra-high definition television systems for production and international programme exchange,” Aug. 2012.
- [23] ITU-T, “ITU-T Recommendation H.261: Line transmission of non-telephone signals: Video codec for audiovisual services at $p \times 64$ kbits,” Nov. 1988.
- [24] —, “ITU-T Recommendation H.263: Video coding for low bit rate communication,” Mar. 1996.
- [25] H. Iwasaki, M. Ikeda, K. Nitta, T. Onishi, T. Sano, A. Sagata, Y. Nakajima, M. Inamori, T. Yoshitome, H. Matsuda, and J. Naganuma, “Professional

- H.264/AVC decoder LSI for high-quality HDTV broadcast infrastructure,” in *Low-Power and High-Speed Chips (COOL Chips XI), 2008 IEEE Symposium on*, Apr. 2008, pp. 287 – 293.
- [26] 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,” in *Proceedings of Design, Automation and Test in Europe Conference and Exhibition, 2003*, Mar. 2003, pp. 2 – 7 suppl.
- [27] 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, Sept. 2007.
- [28] H. Iwasaki, T. Onishi, T. Sano, K. Yokohari, Y. Nakajima, D. Kobayashi, A. Sagata, Y. Takahashi, K. Nitta, and K. Kamikura, “Full HDTV real-time 3D video encoder,” IPSJ SIG Technical Report, Tech. Rep. 72, Mar. 2011.
- [29] 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.
- [30] Y.-K. Lin, D.-W. Li, C.-C. Lin, T.-Y. Kuo, S.-J. Wu, W.-C. Tai, W.-C. Chang, and T.-S. Chang, “A 242mW 10mm² 1080p H.264/AVC high-profile encoder chip,” in *Solid-State Circuits Conference, 2008. ISSCC 2008. Digest of Technical Papers. IEEE International*, Feb. 2008, pp. 314 – 615.
- [31] 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,” in *Proceedings of HOT Chips X*. IEEE, Aug. 1998, pp. 123–131.

- [32] E. Miyagoshi, T. Araki, T. Sayama, A. Ohtani, T. Minemaru, K. Okamoto, H. Kodama, T. Morishige, A. Watabe, K. Aoki, T. Mitsumori, H. Imanishi, T. Jinbo, Y. Tanaka, M. Taniyama, T. Shingou, T. Fukumoto, H. Morimoto, and K. Aono, "A 100mm² 0.95W single-chip MPEG2 MP@ML video encoder with a 128GOPS motion estimator and a multi-tasking RISC-type controller," in *Proceedings of ISSCC98*, Feb. 1998, pp. 30–31.
- [33] H. Mizosoe, D. Yoshida, and T. Nakamura, "A single chip H.264/AVC HDTV encoder/decoder/transcoder system LSI," *Consumer Electronics, IEEE Transactions on*, vol. 53, no. 2, pp. 630 – 635, May 2007.
- [34] M. Mizuno, Y. Ooi, N. Hayashi, J. Goto, M. Hozumi, K. Furuta, A. Shibayama, Y. Nakazawa, O. Ohnishi, S.-Y. Zhu, Y. Yokoyama, Y. Katayama, H. Takano, N. Miki, Y. Senda, I. Tamitani, and M. Yamashina, "A 1.5-W single-chip MPEG-2 MP@ML video encoder with low power motion estimation and clocking," *IEEE Journal of Solid-State Circuits*, vol. 32, no. 11, pp. 1807–1816, Nov. 1997.
- [35] 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," in *VLSI Circuits, 2008 IEEE Symposium on*, June 2008, pp. 106–107.
- [36] 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.
- [37] 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," in *Proceedings of IS&T/SPIE Conference on Visual Communications and Image Processing '99*, vol. 3653. IS&T/SPIE, Jan. 1999, pp. 874–882.
- [38] —, "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.

- [39] 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, pp. 377–385, Apr. 2004, (in Japanese).
- [40] E. Ogura, M. Takashima, D. Hiranaka, T. Ishikawa, Y. Yanagita, S. Suzuki, T. Fukuda, and T. Ishii, "A 1.2-W single-chip MPEG2 MP@ML video encoder LSI including wide search range (H: ± 288 , V: ± 96) motion estimation and 81-MOPS controller," *IEEE Journal of Solid-State Circuits*, vol. 33, no. 11, pp. 1765–1771, Nov. 1998.
- [41] T. Onishi, K. Nakamura, T. Yoshitome, and J. 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.
- [42] T. Onishi, K. Nitta, T. Sano, H. Iwasaki, M. Ikeda, J. Naganuma, and K. Kamikura, "A motion estimation and motion compensation architecture for professional H.264/AVC encoder LSI," *IEICE Transactions on Information and Systems*, vol. J93-D, no. 10, pp. 2148 – 2155, Oct. 2010, (in Japanese).
- [43] T. Onishi, T. Sano, K. Nitta, M. Ikeda, and J. Naganuma, "Multi-reference and multi-block-size motion estimation with flexible mode selection for professional 4:2:2 H.264/AVC encoder LSI," in *Circuits and Systems, 2008. IS-CAS 2008. IEEE International Symposium on*, May 2008, pp. 800 – 803.
- [44] T. Otsuka, K. Sakai, T. Hamano, H. Watanabe, T. Kitaura, Y. Nishizawa, K. Eguchi, Y. Sabetto, and H. Yamashita, "Development of a single chip MPEG2 video encoder," *IEICE, Tech. Rep. ED98-51*, June 1998, (in Japanese).
- [45] 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.
- [46] 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,” in *Proceedings of IS&T/SPIE Conference on Visual Communications and Image Processing '99*, vol. 3653. IS&T/SPIE, Jan. 1999, pp. 895–904.
- [47] A. M. Tourapis, “Enhanced predictive zonal search for single and multiple frame motion estimation,” in *Proceedings of Visual Communications and Image Processing*, Jan. 2002, pp. 1069–1079.
- [48] T.-C. Wang, Y.-W. Huang, H.-C. Fang, and L.-G. Chen, “Performance analysis of hardware oriented algorithm modifications in H.264,” in *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on*, Apr. 2003, pp. II – 493–6 vol.2.
- [49] T. Yamakage, Y. Uetani, T. Kodama, H. Muraoka, N. Kawahara, T. Musha, and T. Inagaki, “A single-chip encoder LSI for MPEG2 MP@ML,” IEICE, Tech. Rep. ED98-52, June 1998, (in Japanese).
- [50] 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.

Appendix A

List of publications

Journal articles related to this dissertation

1. Koyo Nitta, Toshihiro Minami, Toshio Kondo, and Takeshi Ogura, “Motion Estimation and Compensation Hardware Architecture for a Scene-Adaptive Algorithm on a Single-Chip MPEG-2 Video Encoder,” *IEICE Transactions on Information and Systems*, Vol. E84-D, No. 3, pp. 317–325, Mar., 2001.
2. Koyo Nitta, Takeshi Yoshitome, Toshio Kondo, Hiroe Iwasaki, and Jiro Naganuma, “Improvements on SIMD Macroblock Processor in MPEG-2 Video Encoder LSI,” *IEICE Transactions on Electronics*, Vol. J87-C, No. 4, pp. 377–385, Apr., 2004 (in Japanese).
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

Other journal articles

1. Kazuyoshi Takagi, Koyo Nitta, Hironori Bouno, Yasuhiko Takenaga, and Shuzo Yajima, “Computational Power of Nondeterministic Ordered Binary Decision Diagrams and Their Subclasses,” *IEICE Transactions on Fundamentals of Electronics, Communications, and Computer Sciences*, Vol. E80-A, No. 4, pp. 663–669, Apr., 1997.

2. Mitsuo Ikeda, Toshio Kondo, Koyo Nitta, Kazuhito Suguri, Takeshi Yoshitome, Toshihiro Minami, Hiroe Iwasaki, Katsuyuki Ochiai, Jiro Naganuma, Makoto Endo, Yutaka Tashiro, Hiroshi Watanabe, Naoki Kobayashi, Tsuneo Okubo, Takeshi Ogura, and Ryota Kasai, “SuperENC: MPEG-2 VIDEO ENCODER CHIP,” *IEEE Micro*, Vol. 19, No. 4, pp. 56–65, July/Aug., 1999.
3. Takeshi Yoshitome, Toshihiro Minami, Mitsuo Ikeda, Koyo Nitta, and Kazuhito Suguri, “A 4:2:2P@ML MPEG-2 video encoder board using an enhanced MP@ML video encoder LSI,” *IEEE Transactions on Consumer Electronics*, Vol. 45, No. 4, pp. 1130–1133, Nov., 1999.
4. Mitsuo Ikeda, Toshio Kondo, Koyo Nitta, Kazuhito Suguri, Takeshi Yoshitome, Toshihiro Minami, Jiro Naganuma, and Takeshi Ogura, “Three-Layer Cooperative Architecture for MPEG-2 Video Encoder LSI,” *IEICE Transactions on Electronics*, Vol. E83-C, No. 2, pp. 170–178, Feb., 2000.
5. 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 VLSI*, Vol. 15, Issue 9, pp. 1055–1059, Sept., 2007.
6. 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*, Vol. J93-D, No. 10, pp. 2148–2155, Oct., 2010 (in Japanese).
7. Daisuke Kobayashi, Koyo Nitta, Mitsuo Ikeda, Kazuto Kamikura, and Hirohisa Jozawa, “A Study on Complexity Reduction Methods for HEVC Intra Prediction,” *IEICE Transactions on Information and Systems*, Vol. J95-D, No. 9, pp. 1661–1663, Sept., 2012 (in Japanese).

Refereed conference papers

1. Koyo Nitta, Toshihiro Minami, Toshio Kondo, and Takeshi Ogura, “Motion Estimation and Compensation Hardware Architecture for a Scene-Adaptive

-
- Algorithm on a Single-Chip MPEG-2 Video Encoder,” in *Proceedings of SPIE Visual Communications and Image Processing '99 (VCIP'99)*, Vol. 3653, No. 1, pp. 874–882, Jan., 1999.
2. 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*, pp. 106–107, June, 2008.
 3. Toshihiro Minami, Toshio Kondo, Koyo Nitta, Kazuhito Suguri, Mitsuo Ikeda, Takeshi Yoshitome, Hiroshi Watanabe, Hiroe Iwasaki, Katsuyuki Ochiai, Jiro Naganuma, Makoto Endo, Eiichi Yamagishi, Takuro Takahashi, Koichi Tadaishi, Yutaka Tashiro, Naoki Kobayashi, Tuneo Okubo, Takeshi Ogura, and Ryota Kasai, “A single-chip MPEG2 MP@ML video encoder with multi-chip configuration for a single-board MP@HL encoder,” in *Proceedings of Hot Chips: A Symposium on High Performance Chips (HOT CHIPS X)*, pp. 123–131, Aug., 1998.
 4. Mitsuo Ikeda, Toshio Kondo, Koyo Nitta, Kazuhito Suguri, Takashi Yoshitome, Toshihiro Minami, Jiro Naganuma, and Takeshi Ogura, “An MPEG-2 video encoder LSI with scalability for HDTV based on three-layer cooperative architecture,” in *Proceedings of Design, Automation and Test in Europe Conference and Exhibition 1999 (DATE'99)*, pp. 44–50, Mar., 1999.
 5. Takeshi Yoshitome, Toshihiro Minami, Mitsuo Ikeda, Koyo Nitta, Kazuhito Suguri, “A 4:2:2P@ML MPEG-2 video encoder board using an enhanced MP@ML video encoder LSI,” In *Proceedings of International Conference on Consumer Electronics 1999 (ICCE1999)*, pp. 56–57, June, 1999.
 6. Mitsuo Ikeda, Toshio Kondo, Koyo Nitta, Kazuhito Suguri, Takeshi Yoshitome, Toshihiro Minami, Jiro Naganuma, and Takeshi Ogura, “SuperENC: MPEG-2 Video Encoder LSI based on Three-Layer Cooperative Architecture,” in *Proceedings of IEEE Symposium on Low-Power and High-Speed Chips (COOL Chips II)*, pp. 95–106, Apr., 1999.
 7. Takeshi Yoshitome, Ken Nakamura, Koyo Nitta, Mitsuo Ikeda, and Makoto Endo, “Development of an HDTV MPEG-2 encoder based on multiple en-

- hanced SDTV encoding LSIs,” in *Proceedings of International Conference on Consumer Electronics 2001 (ICCE2001)*, pp. 160–161, June, 2001.
8. 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.
 9. 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 suppl., Mar., 2003.
 10. Jiro Naganuma, Hiroe Iwasaki, Mitsuo Ikeda, Koyo Nitta, Ken Nakamura, Takeshi Yoshitome, 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.
 11. 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.
 12. 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.

13. 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 IEEE International Symposium on Circuits and Systems, 2008 (ISCAS2008)*, pp. 800–803, May, 2008.
14. Takefumi Yamazaki, Hiroto Inagaki, Koyo Nitta, and Masayuki Ito, "The intermediate service platform architecture for home networking services," in *Proceedings of IEEE 14th International Symposium on Consumer Electronics (ISCE2010)* pp. 1–6, June, 2010.
15. 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.
16. 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.
17. Takashi Sano, Kazuya Yokohari, Koyo Nitta, Mitsuo Ikeda, Kazuto Kamikura, and Atsushi Shimizu, "Inter-channel Rate Control Method for H.264/AVC Multi-channel Video Encoder," in *Proceedings of The 1st IEEE Global Conference on Consumer Electronics (GCCE2012)*, pp. 629–632, Oct., 2012.
18. Kazuya Yokohari, Koyo Nitta, Mitsuo Ikeda, and Atsushi Shimizu, "Parallel Design Methodology for Video Codec LSI with High-Level Synthesis and FPGA-Based Platform," *Design Automation Conference (DAC2013)*, No. 8D.4, June, 2013.
19. Jia Su, Koyo Nitta, Mitsuo Ikeda, and Atsushi Shimizu, "Residue role assignment based transform partition predetermination on HEVC," in *Proceedings of 20th IEEE International Conference on Image Processing (ICIP2013)*, pp. 2019–2023, Sept., 2013.

Reviews

1. Koyo Nitta, Hiroe Iwasaki, and Jiro Naganuma, “H.264/AVC Codec LSI Configuration Technology and Application to IP Retransmission Services,” *NTT Technical Journal*, Vol. 21, No. 3, pp. 24–25, Mar., 2009 (in Japanese).
2. Koyo Nitta, Hiroe Iwasaki, Jiro Naganuma, “H.264/AVC Codec LSI Configuration Technology and Application to IP Retransmission Services,” *NTT Technical Review*, Vol. 7, No. 11, Nov. 2009.
3. Jiro Naganuma, Hiroe Iwasaki, Koyo Nitta, Ken Nakamura, Takeshi Yoshitome, Mitsuo Ogura, Yasuyuki Nakajima, Yutaka Tashiro, Takayuki Onishi, Mitsuo Ikeda, Makoto Endo, and Yoshiyuki Yashima, “Design Technology for a Single-Chip HDTV MPEG-2 CODEC LSI — VASA,” *NTT Technical Review*, Vol. 15, No. 9, pp. 12–15, Sept., 2003 (in Japanese).
4. Hiroto Inagaki, Takehiko Ohno, Koyo Nitta, Akira Nakagawa, and Shinichiro Chaki, “Service and Terminal Component Technology Development Initiatives for the Optical Fiber Broadband Era (Hikari Era),” *NTT Technical Review*, Vol. 22, No. 12, pp. 12–14, Dec., 2010 (in Japanese).
5. Hiroto Inagaki, Takehiko Ohno, Atsushi Sato, Koyo Nitta, Hisashi Uematsu, Akira Nakagawa, and Shinichiro Chaki, “Service and Terminal Component Technology Development Initiatives for the Optical Fiber Broadband Era (Hikari Era),” *NTT Technical Review*, Vol. 9, No. 6, June, 2011.
6. Takashi Sano, Ryuichi Tanida, Masaki Kitahara, Koyo Nitta, “Research and development of high-compression encoder for spread of high resolution video beyond High Definition,” *Business Communication*, Vol. 50, No. 2013/03, pp. 26–27, Mar., 2013 (in Japanese).

Reports

1. Koyo Nitta, Yasuhiko Takenaga, and Shuzo Yajima, “Expressive power of Binary Decision Diagrams Representing Sum-of-product Form,” in *Proceedings of IEICE General Conference 1995*, Vol. 1995, No. 1, p. 4, Mar., 1995.

-
2. Koyo Nitta, Yasuhiko Takenaga, and Shuzo Yajima, “Expressive power of Binary Decision Diagrams Representing Sum-of-product Form,” in *Kyoto University Research Institute for Mathematical Sciences (RIMS)*, Vol. 906, pp. 196–203, Apr., 1995.
 3. Koyo Nitta, Takeshi Yoshitome, Toshihiro Minami, and Toshio Kondo, “A SIMD Motion-Estimation Processor for Half Pel Precision Search,” in *Proceedings of IEICE General Conference 1997*, C–12–30, Mar., 1997 (in Japanese).
 4. Koyo Nitta, Takeshi Yoshitome, Toshihiro Minami, and Toshio Kondo, “Mono-field Motion Vector Search Method,” in *Proceedings of IEICE General Conference 1998*, D–11–43, Mar., 1998 (in Japanese).
 5. Koyo Nitta, Toshihiro Minami, and Toshio Kond, “MPEG-2 Video Encoder LSI: ME/MC Architecture for Scene-Adaptive Control,” in *Proceedings of IEICE General Conference 1999*, C–12–41, Vol. 1999, No. 2, p. 139, Mar., 1999 (in Japanese).
 6. Koyo Nitta, Takeshi Yoshitome, Toshihiro Minami, Mitsuo Ikeda, and Kazuhito Suguri, “4:2:2P@ML MPEG-2 Video Encoder Board,” in *Proceedings of IEEEJ Media Computing Conference 1999* Vol. 1999, No. 16, June, 1999 (in Japanese).
 7. Koyo Nitta and Takeshi Ogura, “Pixel-level parallel watershed algorithm for region segmentation,” in *Proceedings of IEICE General Conference 2000*, D–12–71, Vol. 2000, No. 2, p.241, Mar., 2000 (in Japanese).
 8. Koyo Nitta and Takeshi Ogura, “Pixel-level parallel watershed algorithm for region segmentation,” in *Proceedings of IEEEJ Media Computing Conference 2000* Vol. 2000, No. 19, June, 2000 (in Japanese).
 9. Koyo Nitta and Takeshi Yoshitome, “Offset updating and adaptive f_code setting in an offset motion vector search,” in *Proceedings of IEICE Society Conference 2000*, D–11–10, p. 120, Sept., 2000 (in Japanese).
 10. Koyo Nitta, Ken Nakamura, and Takeshi Yoshitome, “Variable GOP structure in MPEG-2 Multi-view profile,” in *Proceedings of IEICE General Conference 2001*, D–11–58, Vol. 2001.2, p. 58, Mar., 2001 (in Japanese).

11. Koyo Nitta, Hiroe Iwasaki, Mitsuo Ikeda, Kazuhito Suguri, Takeshi Yoshitome, and Toshio Kondo, “Control structure of an MPEG-2 video encoder LSI for HDTV,” *IEICE Technical Report*, Computer Systems (CPSY), CPSY2001-9, Vol. 101, No. 2, pp. 65–72, Apr., 2001 (in Japanese).
12. Koyo Nitta, Hiroshi Ogawa, Takeshi Yoshitome, and Makoto Endo, “Study on hardware architecture of digital watermarking,” in *Proceedings of IEICE Society Conference 2001*, C–12–10, No. 2, p. 71, Sept., 2001 (in Japanese).
13. Koyo Nitta, Yasuhiko Sato, Takeshi Yoshitome, and Toshio Kondo, “Improvement on SIMD Macroblock Processor in MPEG-2 Video Encoder LSI,” *IEICE Technical Report*, Integrated Circuits and Devices (ICD), ICD2001-127, Vol 101, No. 386, pp. 91–96, Oct., 2001 (in Japanese).
14. 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, “SARA: A Professional H.264/AVC Encoder LSI for HDTV CODEC Systems — Design Concepts and Basic Architecture —,” in *Proceedings of IEICE General Conference 2008*, D–11–110, Vol 2008, No. 2, Mar., 2008 (in Japanese).
15. Koyo Nitta, Hiroe Iwasaki, and Jiro Naganuma, “Ultra Low-Delay H.264/AVC Intra Encoder on FPGA with SystemC Behavioral Synthesis,” *IEICE Technical Report*, Functional Integrated Information System (FIIS), Vol. 36, No. 8, Aug., 2008.
16. 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,” *IEICE Technical Report*, Integrated Circuits and Devices (ICD), ICD2008-149, pp. 117–122, Jan., 2009.
17. Koyo Nitta, Mitsuo Ikeda, Hiroe Iwasaki, Jiro Naganuma, and Hirohisa Jozawa, “An H.264/AVC video encoder LSI for broadcasting infrastructure and its applications (Invited talk),” *IEICE Technical Report*, Integrated Circuits and Devices (ICD), Vol. 111, No. 258, ICD2011-68 (SIP2011-65, IE2011-64), pp. 19–24, Oct., 2011 (in Japanese).

-
18. Toshihiro Minami, Koyo Nitta, and Toshio Kondo, “A Proposal of a Pixel-decimation-type Motion-estimation Method,” in *Proceedings of IEICE General Conference 1997*, D-11-59, Mar., 1997 (in Japanese).
 19. Toshio Kondo, Toshihiro Minami, Koyo Nitta, Kazuhito Suguri, Mitsuo Ikeda, Hiroe Iwasaki, Katsuyuki Ochiai, and Hiroshi Watanabe, “An MPEG2 video encoder LSI expandable into HDTV,” *IEICE Technical Report*, Integrated Circuits and Devices (ICD), ICD98-117, Vol. 98, No. 244, pp. 37-43, Aug., 1998 (in Japanese).
 20. Mitsuo Ikeda, Kazuhito Suguri, Koyo Nitta, Toshihiro Minami, and Toshio Kondo, “MPEG-2 Video Encoder LSI: Three-Layer Cooperative Architecture,” in *Proceedings of IEICE General Conference 1999*, C-12-39, Vol. 1999, No. 2, p. 137, Mar., 1999 (in Japanese).
 21. Toshihiro Minami, Koyo Nitta, and Toshio Kondo, “MPEG-2 Video Encoder LSI : Memory Architecture for Adaptive Search,” in *Proceedings of IEICE General Conference 1999*, C-12-42, Vol. 1999, No. 2, p. 140, Mar., 1999 (in Japanese).
 22. Ken Nakamura, Koyo Nitta, Takeshi Yoshitome, and Makoto Endo, “A Study on Rate Control using Results of Pre-processing and 1st Motion Estimation,” in *Proceedings of IEICE General Conference 2001*, D-11-19, Vol. 2001.2, p. 19, Mar., 2001 (in Japanese).
 23. Ken Nakamura, Koyo Nitta, Mitsuo Ikeda, Takeshi Yoshitome, and Makoto Endo, “MPEG-2 HDTV Encoder System based on Multiple SDTV Encoder LSIs,” *IEICE Technical Report*, Integrated Circuits and Devices (ICD), ICD2001-78, Vol. 101, No. 266, pp. 91-96, Aug., 2001 (in Japanese).
 24. Jiro Naganuma, Hiroe Iwasaki, Koyo Nitta, Ken Nakamura, Takeshi Yoshitome, Mitsuo Ogura, Yasuyuki Nakajima, Yutaka Tashiro, Takayuki Onishi, Mitsuo Ikeda, Makoto Endo, and Yoshiyuki Yashima, “Single-chip MPEG-2 422P@HL CODEC LSI (VASA) — Design Concept and Basic Architecture —,” in *Proceedings of Forum on Information Technology (FIT2003)*, J-101, Sept., 2003 (in Japanese).
 25. Hiroe Iwasaki, Jiro Naganuma, Koyo Nitta, Ken Nakamura, Takeshi Yoshitome, Mitsuo Ogura, Yasuyuki Nakajima, Yutaka Tashiro, Takayuki Onishi,

- Mitsuo Ikeda, Makoto Endo, and Yoshiyuki Yashima, “Single-chip MPEG-2 422P@HL CODEC LSI (VASA) — Encoder and External Memory IF —,” in *Proceedings of Forum on Information Technology (FIT2003)*, J-102, Sept., 2003 (in Japanese).
26. Yasuyuki Nakajima, Yutaka Tashiro, Mitsuo Ogura, Jiro Naganuma, Hiroe Iwasaki, Koyo Nitta, Ken Nakamura, Takeshi Yoshitome, Takayuki Onishi, Mitsuo Ikeda, Makoto Endo, and Yoshiyuki Yashima, “Single-chip MPEG-2 422P@HL CODEC LSI (VASA) — Architecture of decoder and external memory interface —,” in *Proceedings of Forum on Information Technology (FIT2003)*, J-103, Sept., 2003 (in Japanese).
27. 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, Sept., 2003 (in Japanese).
28. Ken Nakamura, Takeshi Yoshitome, Jiro Naganuma, Hiroe Iwasaki, Koyo Nitta, Mitsuo Ogura, Yasuyuki Nakajima, Yutaka Tashiro, Takayuki Onishi, Mitsuo Ikeda, Makoto Endo, and Yoshiyuki Yashima, “Single-chip MPEG-2 422P@HL CODEC LSI (VASA) — Enhanced encoding and decoding functions —,” in *Proceedings of Forum on Information Technology (FIT2003)*, J-106, Sept., 2003 (in Japanese).
29. Jiro Naganuma, Hiroe Iwasaki, Mitsuo Ikeda, Ken Nakamura, Takeshi Yoshitome, Takayuki Onishi, Yasuyuki Nakajima, Yutaka Tashiro, Koyo Nitta, Mitsuo Ogura, Toshihiro Minami, Makoto Endo, and Yoshiyuki Yashima, “Single-chip MPEG-2 422P@HL CODEC LSI and its applications,” *IEICE Technical Report*, Integrated Circuits and Devices (ICD), ICD2004-104, Vol. 104, No. 288, pp. 41–46, Sept., 2004 (in Japanese).
30. Jiro Naganuma, Hiroe Iwasaki, Mitsuo Ikeda, Ken Nakamura, Takeshi Yoshitome, Takayuki Onishi, Yasuyuki Nakajima, Yutaka Tashiro, Koyo Nitta, Mitsuo Ogura, Toshihiro Minami, Makoto Endo, and Yoshiyuki Yashima, “Single-chip MPEG-2 422P@HL CODEC LSI and its applications,” *ITE Technical Report*, Vol. 28, No. 49, pp. 41–46, Sept., 2004 (in Japanese).

-
31. Yasuyuki Nakajima, Mitsuo Ikeda, Hiroe Iwasaki, Koyo Nitta, Takayuki Onishi, Takashi Sano, Atsushi Sagata, 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 — Architecture of HDTV Encoder —,” in *Proceedings of IEICE General Conference 2008*, D-11-111, Vol 2008.2, Mar., 2008 (in Japanese).
 32. 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).
 33. Atsushi Sagata, Mitsuo Ikeda, Hiroe Iwasaki, Koyo Nitta, Takayuki Onishi, Takashi Sano, 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 — Architecture of Firmware for Video Quality Improvement —,” in *Proceedings of IEICE General Conference 2008*, D-11-113, Vol 2008.2, Mar., 2008 (in Japanese).
 34. Hiroe Iwasaki, Mitsuo Ikeda, Koyo Nitta, Takayuki Onishi, Takashi Sano, Atsushi Sagata, Yasuyuki Nakajima, Minoru Inamori, Takeshi Yoshitome, Hiroaki Matsuda, Ryuichi Tanida, Atsushi Shimizu, Ken Nakamura, and Jiro Naganuma, “SARA/D: A Professional HDTV H.264/AVC Decoder LSI,” in *Proceedings of IEICE General Conference 2008*, D-11-114, Vol 2008.2, Mar., 2008 (in Japanese).
 35. Atsushi Sagata, Mitsuo Ikeda, Hiroe Iwasaki, Koyo Nitta, Takayuki Onishi, Takashi Sano, Yasuyuki Nakajima, Minoru Inamori, Takeshi Yoshitome, Hiroaki Matsuda, Ryuichi Tanida, Atsushi Shimizu, Ken Nakamura, and Jiro Naganuma, “A Professional-use Transcoder for Retransmission of Digital Terrestrial TV Broadcast over IP — Structure of MPEG-2/H.264 Transcoder —,” in *Proceedings of IEICE General Conference 2009*, D-11-10, Vol 2009.2, Mar., 2009 (in Japanese).

36. Takashi Sano, Mitsuo Ikeda, Hiroe Iwasaki, Koyo Nitta, Takayuki Onishi, Atsushi Sagata, Yasuyuki Nakajima, Minoru Inamori, Takeshi Yoshitome, Hiroaki Matsuda, Ryuichi Tanida, Atsushi Shimizu, Ken Nakamura, and Jiro Naganuma, “A Professional-use Transcoder for Retransmission of Digital Terrestrial TV Broadcast over IP — Functions for High-quality Transcoding —,” in *Proceedings of IEICE General Conference 2009*, D-11-11, Vol. 2009.2, Mar., 2009 (in Japanese).
37. Hiroto Inagaki, Takefumi Yamazaki, Koyo Nitta, and Masayuki Ito, “The intermediate service platform architecture for home networking services,” in *Proceedings of IEICE General Conference 2010*, BS-8-15, Vol. 2010.2, Mar., 2010.
38. Takayuki Onishi, Koyo Nitta, Hiroe Iwasaki, and Kazuto Kamikura, “Development of a 12-bit 4:2:2 AVC/H.264 HDTV Intra Encoder,” in *Proceedings of IEICE General Conference 2010*, D-11-9, Vol. 2010.2, Mar., 2010 (in Japanese).
39. Hiroe Iwasaki, Takayuki Onishi, Takashi Sano, Kazuya Yokohari, Yasuyuki Nakajima, Daisuke Kobayashi, Atsushi Sagata, Yasuko Takahashi, Koyo Nitta, and Kazuto Kamikura, “Full HDTV Real-time 3D Video Encoder,” *IPSJ Technical Report*, Audio Visual and Multimedia information processing (AVM), Vol. 2011-AVM-72, No. 4, Mar., 2011 (in Japanese).
40. Daisuke Kobayashi, Kazuya Yokohari, Koyo Nitta, Kazuto Kamikura, and Hirohisa Jozawa, “A Study on Ultra Low Delay Video Encoder with 2-pass Coding Method,” *IPSJ Technical Report*, Audio Visual and Multimedia information processing (AVM), Vol. 2011-AVM-74, No. 8, Sept., 2011 (in Japanese).
41. Daisuke Kobayashi, Koyo Nitta, Mitsuo Ikeda, Kazuto Kamikura, and Hirohisa Jozawa, “A Study on Complexity Reduction Methods for HEVC Intra Prediction,” in *Proceedings of The 26th Picture Coding Symposium of Japan (PCSJ2011)*, P4-05, Oct., 2011 (in Japanese).
42. Takashi Sano, Daisuke Kobayashi, Kazuya Yokohari, Koyo Nitta, Mitsuo Ikeda, Kazuto Kamikura, and Hirohisa Jozawa, “Video Encoder with Inter-channel Rate Control Function,” in *Proceedings of IEICE General Conference 2012*, D-11-59, Vol. 2012.2, Mar., 2012 (in Japanese).

-
43. Daisuke Kobayashi, Koyo Nitta, Mitsuo Ikeda, Kazuto Kamikura, and Hirohisa Jozawa, “A Study on CU Splitting Method for HEVC Intra Prediction,” in *Proceedings of IEICE General Conference 2012*, D–11–85, Vol. 2012.2, Mar., 2012 (in Japanese).
 44. Kazuya Yokohari, Koyo Nitta, Mitsuo Ikeda, and Atsushi Shimizu, “Design Methodology for Video Codec LSI with High-level Synthesis and FPGA-based Platform,” in *Proceedings of International Conference on Integrated Circuits and Devices in Vietnam (ICDV 2012)*, Aug., 2012.
 45. Takashi Sano, Kazuya Yokohari, Koyo Nitta, Mitsuo Ikeda, Kazuto Kamikura, and Atsushi Shimizu, “Inter-channel Rate Control Method for H.264/AVC Multi-channel Video Encoder,” *ITE Technical Report*, Media Engineering Research Committee (ITE-ME), Vol. 36, No. 33, pp. 127–131, Aug., 2012 (in Japanese).
 46. Takashi Sano, Kazuya Yokohari, Koyo Nitta, Mitsuo Ikeda, Kazuto Kamikura, and Atsushi Shimizu, “Inter-channel Rate Control Function for H.264/AVC 4-channel Video Encoder,” in *Proceedings of ITE Annual Conference 2012*, No. 2012, pp. 3-2-1–3-2-2, Aug., 2012 (in Japanese).
 47. Kazuya Yokohari, Takashi Sano, Koyo Nitta, Mitsuo Ikeda, Kazuto Kamikura, and Atsushi Shimizu, “2-Pass Rate Control for H.264/AVC Multi-Core Video Encoder,” *ITE Technical Report*, Media Engineering Research Committee (ITE-ME), Vol. 36, No. 39, pp. 5–8, Oct., 2012 (in Japanese).
 48. Kimiko Kawashima, Jun Okamoto, Takashi Sano, Takayuki Onishi, Atsushi Sagata, Koyo Nitta, Hiroe Iwasaki, and Takanori Hayashi, “Verification on coding method for ultra HD video services by subjective quality assessment,” in *Proceedings of IEICE Society Conference 2014*, B–11–26, Sept., 2014 (in Japanese).

All Figures in Chapter 3, Chapter 4, and Chapter 5, except for Figure 5.6, 5.7, and 5.10 are copyrighted by the Institute of Electronics, Information and Communication Engineers (IEICE).