Advances in Hardware Architectures for Image and Video Coding—A Survey

PO-CHIH TSENG, YUNG-CHI CHANG, YU-WEN HUANG, HUNG-CHI FANG,
CHAO-TSUNG HUANG, AND LIANG-GEE CHEN, FELLOW, IEEE

Invited Paper

This paper provides a survey of state-of-the-art hardware architectures for image and video coding. Fundamental design issues are discussed with particular emphasis on efficient dedicated implementation. Hardware architectures for MPEG-4 video coding and JPEG 2000 still image coding are reviewed as design examples, and special approaches exploited to improve efficiency are identified. Further perspectives are also presented to address the challenges of hardware architecture design for advanced image and video coding in the future.

Keywords**—***Hardware architecture, H.264/AVC, image coding, JPEG 2000, MPEG-4, very large scale integration (VLSI), video coding.*

I. INTRODUCTION

Due to advances in image and video coding algorithms as well as in very large scale integration (VLSI) technology, diverse and interesting visual experiences have been brought to human daily life. A number of international standards have contributed to the great success of image and video coding applications [1]. Still image compression applications, such as digital still cameras, are covered by the ISO/IEC JPEG standards, both the current JPEG [2] and the emerging JPEG 2000 [3]. The present MPEG-1 [4] and MPEG-2 [5] of ISO/IEC MPEG standards are used for video storage and playback, digital TV broadcast, and video-on-demand applications, while the emerging MPEG-4 [6] intends to cover wide-ranging multimedia communications applications. Video communications applications, such as video telephony and video conference, are regulated by the ITU-T H.26X standards, including early-stage H.261 [7], present H.263 [8], and the new-generation H.264/AVC [9].

The availability of low-cost and low-power hardware with sufficiently high performance is essential for the popularization of image and video coding applications. Thus, efficient hardware implementations in VLSI are of vital importance. However, image and video coding algorithms are characterized by very high computational complexity. Real-time processing of multidimensional image and video signal involves operating continuous data streams of huge volumes. Such critical demands cannot be fulfilled by conventional hardware architectures without specific adaptation. Therefore, special architectual approaches are indispensable for efficient hardware solutions to meet real-time constraints with desired low cost and low power.

In the literature, two effective surveys of hardware architectures for image and video coding have been presented [10], [11]. The major focuses of these previous studies are on present standards such as JPEG, MPEG-1, MPEG-2, H.261, and H.263. However, emerging MPEG-4 and JPEG 2000 are capable of offering both improved coding efficiency and additional functionalities beyond present standards. These advanced features further increase the computational complexity and consequently pose challenges for hardware architecture design. This paper provides a survey of state-of-the-art hardware architectures for image and video coding, primarily focusing on emerging MPEG-4 and JPEG 2000.

This paper is organized as follows. An overview of fundamental design issues is given in Section II. Hardware architectures for MPEG-4 video coding and JPEG 2000 still image coding are reviewed as design examples in Sections III and IV, respectively. Finally, the further perspectives in Section V are presented to address the challenges of hardware architecture design for advanced image and video coding in the future.

II. FUNDAMENTALS OF HARDWARE ARCHITECTURES

In general, image and video coding algorithms are hybrid coding schemes, which consist of several tasks to jointly

Manuscript received December 18, 2003; revised June 8, 2004. This work was supported in part by the National Science Council, Taiwan, under Grant 92-2215-E-002-015 and in part by MediaTek Inc.

The authors are with the DSP/IC Design Laboratory, Graduate Institute of Electronics Engineering, Department of Electrical Engineering, National Taiwan University, Taipei 106, Taiwan, R.O.C. (e-mail: pctseng@video.ee.ntu.edu.tw).

Digital Object Identifier 10.1109/JPROC.2004.839622

reduce the temporal, spatial, and statistical redundancies in image and video signal. In this paper, we define the hardware architectures for individual tasks as module architectures and the hardware architectures for complete coding algorithms as system architectures.

The module architectures can be classified into two categories: programmable and dedicated. Programmable architectures offer the flexibility to allow various tasks to be executed on the same hardware by only software modifications. However, the penalties of flexibility are additional hardware cost and higher power consumption. Dedicated architectures are derived by full adaptation to specific tasks, and they can achieve the highest silicon efficiency with minimum hardware overhead and lower power consumption. However, the disadvantage of dedication is the impossibility for later extensions to other tasks.

Owing to the hybrid coding schemes of image and video coding, the computational characteristics of various tasks are different. In order to efficiently process multiple individual tasks, the system architectures have generally evolved to hybrid architectures, which are flexibly composed of several different programmable and dedicated module architectures. Nevertheless, according to the category of dominating computation-intensive module architectures in each combination, hybrid system architectures can also be classified into programmable (oriented) and dedicated (oriented) categories.

The adoption of programmable or dedicated system architectures mainly depends on the target application field. Programmable system architectures are suitable for larger application fields with largely varying algorithms and computational demands, while dedicated system architectures are suitable only for well-defined applications with fixed functionality. Since the subject of this paper is specific image and video coding, in order to obtain the most efficient hardware solutions, the particular emphasis of this paper is on dedicated system architectures. The interested reader is referred to [10]–[14] for more information regarding programmable architectures.

Efficient dedicated hardware implementations for image and video coding rely on the thorough analysis of target algorithms and the exploitation of special computational characteristics inherent in the algorithms. The design goal is to achieve a dedicated system architecture with the highest degree of adaptation. The design approach is to perform the mapping of individual tasks onto different module architectures, and then to optimize each module architecture in terms of performance, area, and power constraints.

In order to exploit the special computational characteristics inherent in these algorithms, a detailed algorithm analysis can be performed. This analysis involves two aspects: one is to perform the computational complexity analysis by task profiling to capture the computational share of each task, and the other is to perform the computational characteristic analysis for each task. The computational characteristics of image and video coding algorithms can be classified into low-level, medium-level, and high-level. Low-level tasks are characterized by highly regular computation, medium-level tasks have frequently irregular computation and involve data-dependent decisions, while high-level tasks have a highly irregular and unpredictable computational flow.

Based on the algorithm analysis, an exhaustive architecture exploration can be performed to identify the suitable category of module architectures for each task. As image and video coding algorithms are typically dominated by several computation-intensive tasks, the design emphasis is, thus, primary on these tasks. However, efficient execution of other tasks is still essential for the overall system performance and cannot be neglected. For present standards such as JPEG, MPEG-1, MPEG-2, H.261, and H.263, the computational characteristics of computation-intensive tasks are mostly low-level. Consequently, dedicated architectures are always adopted for these highly regular tasks. However, for the sake of offering improved coding efficiency and additional functionalities, some computation-intensive tasks of the emerging MPEG-4 and JPEG 2000 are characterized by increased diversity and decreased predictability, in addition to greater computational demands. This presents the challenges of efficient architecture design for MPEG-4 and JPEG 2000. As for other less demanding tasks with medium-level or high-level computational characteristics, programmable architectures are suitable to provide sufficient flexibility for irregular and data-dependent computation. In order to obtain a highly efficient solution, dedicated architectures can also be adopted for these sophisticated tasks with full adaptation. However, substantial design efforts must be involved in designing such architectures.

In addition to module architectures, a complete system architecture also includes memory architecture and interconnect architecture. As image and video coding algorithms involve a large amount of data computation, during processing continuous data streams in huge volumes, a large amount of data communication is also incurred in two aspects. The first is the data access through the frame buffer, which is mainly dominated by the memory architecture; and the other is the data access between different module architectures and the memory architecture, which is mainly dominated by the interconnect architecture. Data communication has already become a bottleneck for complete system architectures and has a significant impact on the overall system performance and power consumption. In the rest of this section, special architectural approaches with regard to memory architecture and interconnect architecture are discussed.

Data access through the frame buffer is a slow and power-consuming process, and two architectural approaches can be applied to relieve this problem. The first approach is to adopt special local memory buffers. Because data access patterns of most tasks of image and video coding are predictable, and there are also many repetitive data access patterns, the adoption of special local memory buffers can, thus, effectively reduce redundant data access by using data-reuse property. However, there is a tradeoff between the size of local memory buffers and the reduced amount of data access. The other approach is to integrate the off-chip frame buffer as on-chip memory, since on-chip frame buffer can significantly improve the data access performance and reduce the I/O power consumption. However, the large size of frame buffer is a

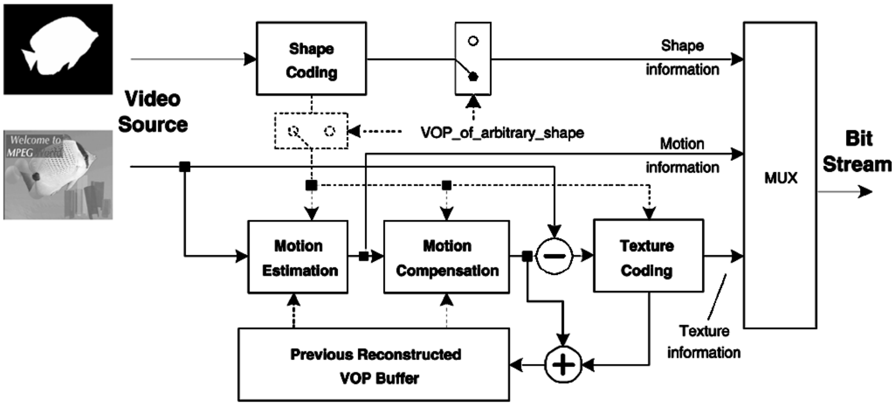

Fig. 1. Functional block diagram of MPEG-4 video coding.

major concern for integration. Due to advances in VLSI technology, various designs have already integrated embedded DRAM or SRAM as on-chip frame buffer to solve the data access problem.

In addition to memory architecture, the interconnect architecture that communicates data between different module architectures and the memory architecture is another key factor for data communication. In-depth understanding of the intermodule communication patterns is essential to design an efficient interconnect architecture, which can provide higher bandwidth and consume lower power. There is a tradeoff between flexibility and efficiency of interconnect architecture. For example, the global bus provides higher flexibility but lower efficiency, whereas the dedicated data link with full adaptation to specific algorithm provides the highest efficiency but without flexibility.

In the following two sections, state-of-the-art hardware architectures for MPEG-4 video coding and JPEG 2000 still image coding are reviewed as design examples, including algorithm analysis and architecture exploration, several module architectures for computation-intensive tasks, as well as the complete system architectures composed of module architectures, memory architecture, and interconnect architecture. Furthermore, special approaches exploited to improve efficiency are discussed.

III. HARDWARE ARCHITECTURES FOR MPEG-4

MPEG-4 is the recently most popular video coding standard. Fig. 1 shows its generic functional block diagram. For more information on MPEG-4, the reader is referred to [15] and [16].

A. *Algorithm Analysis and Architecture Exploration*

According to the computational complexity analysis for core profile reported by [17] and [18], the dominating computation-intensive tasks for encoder are ME and shape encoding, which together contribute more than 90% of the overall complexity. For simple profile without shape coding tools, ME becomes the most significant task. The other less demanding but also computation-intensive tasks are discrete cosine transform (DCT)/inverse DCT (IDCT), quantization (Q)/inverse Q (IQ), motion compensation (MC), padding, and variable-length coding (VLC). As for the decoder, the computation-intensive tasks are shape decoding, IDCT, IQ, MC, padding, and variable-length decoding (VLD). In addition to core profile, the computational complexity analysis for advanced simple profile [19] indicates that global MC (GMC) has become another computation-intensive task for the decoder.

Since the computation-intensive ME belongs to highly regular low-level task, dedicated architectures are extremely important for efficient implementations. In addition to the highly demanding computation part, a huge amount of data access through the frame buffer is also incurred during the ME computing process. Therefore, special local memory buffers are heavily relied to reduce the data access.

Shape coding is another critical task, consisting mainly of two kinds of subtasks with different computational characteristics. One is the binary motion estimation (BME)/binary motion compensation (BMC), which belong to highly regular low-level and bit-level processing tasks; and the other is the context-based arithmetic encoder (CAE)/context-based arithmetic decoder (CAD), which belong to irregular and data-dependent bit-level processing tasks. Due to the bit-level processing characteristics, dedicated architectures are essential for shape coding in order to obtain efficient hardware solutions.

VLC/VLD are less demanding but also computation-intensive tasks. Due to their irregular and data-dependent non-word-aligned processing characteristic, special architectural approaches are also required.

For the other less demanding but also computation-intensive tasks, including DCT/IDCT, Q/IQ, MC, and padding, dedicated architectures can be adopted for these highly regular low-level tasks. Although there exist a variety of architectural alternatives for DCT/IDCT, the advances in architectures for DCT/IDCT are limited. Most of them are covered by the scope of previous studies [10], [11]. The reader is referred to these two surveys for details. As for the relatively simple Q/IQ, MC, and padding, their detailed architectures are not discussed.

GME/GMC are potentially computation-intensive tasks that may even demand higher computational complexity

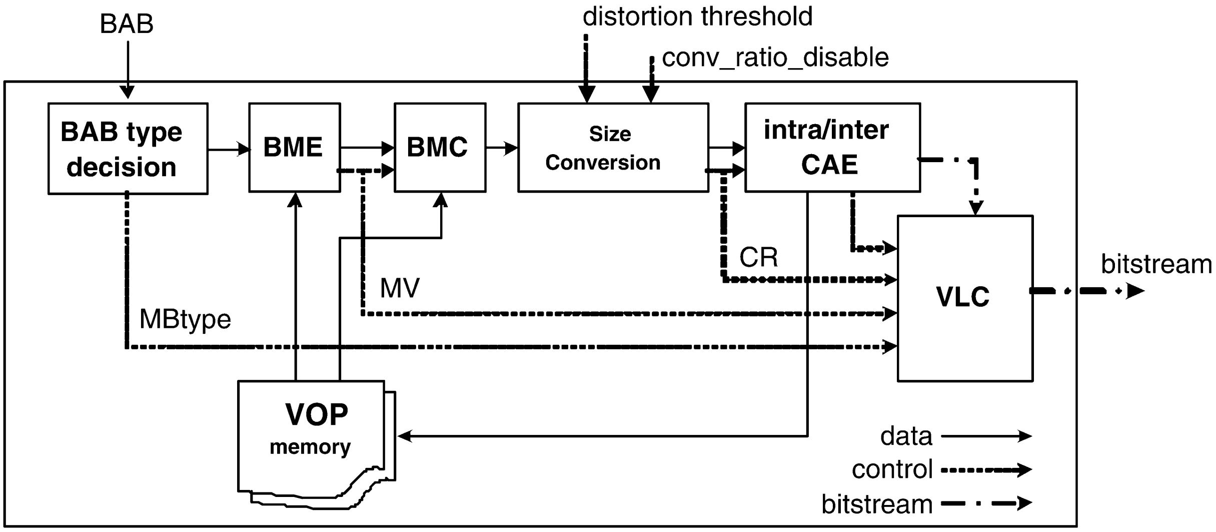

Fig. 2. Coding flow of MPEG-4 shape coding.

than ME/MC. However, the progress in GME/GMC architectures has been very limited, with only a single proposal [20] so far. Fine-granularity scalability (FGS) is another interesting task that involves bit-plane level processing. In [21], an architecture with reordered coding flow and dynamic bit-plane adaptation scheme is proposed for FGS. The other tasks of MPEG-4 video coding are less demanding but high-level tasks, such as system control and rate control. Programmable architectures are suitable for these tasks.

B. Module Architectures

In this subsection, architectures for ME, shape coding, and VLC/VLD are reviewed and discussed in detail.

1) Architectures for ME: The block matching approach is generally selected as the ME module in video codecs because of its simplicity and good performance. Among all the block matching algorithms, the full-search block matching algorithm (FSBMA) is the most typical one due to its regularity. However, FSBMA demands a huge amount of computational complexity that cannot be easily afforded by hardware architectures. Therefore, many fast search algorithms have been proposed to reduce the complexity, and many hardware architectures have been developed to meet the real-time constraints. In the following, the architectures for FSBMA are introduced first, followed by the architectures for fast BMAs as well as the discussion of on-chip memory for storing search area.

In the literature, there have been various one-dimensional (1-D) and two-dimensional (2-D) systolic/semisystolic array architectures proposed for FSBMA [22]–[28]. In addition, a completely different tree architecture is also proposed by [29]. According to our survey, PE array is the trend for FSBMA architecture design. Efficient tradeoffs can be made between area (number of PEs) and throughput (processing capability), between latency (cycles to compute a SAD) and memory bit-width/bandwidth (serial/parallel loading), as well as between PE utilization and extra data alignment circuits (shift registers/memory with circular addressing). The designers must carefully select what can be sacrificed and what should be taken into account for target applications.

Fast search algorithms can reduce the heavy computation of FSBMA with acceptable video quality degradation. The challenges of architecture design for fast BMAs include unpredictable data flow, irregular memory access, difficult mapping to systolic arrays, low hardware utilization, and sequential procedures with data dependence that cannot be parallelized. Furthermore, the silicon area of fast BMA architectures must be significantly smaller than that of FSBMA architectures for cost efficiency. The most representative designs are [30]–[34]. Based on our survey, the trend of architecture design for fast BMAs is toward algorithmic and architectural codesign. The benefits from the algorithmic level are usually larger than those from the architectural level.

Finally, as indicated in the previous subsection, a huge amount of data access through the frame buffer is incurred during the ME computing process. For this issue, special local memory buffers with data-reuse property must be adopted as the on-chip memory for storing search area, which can effectively reduce redundant data access in those well predictable data access patterns with many repetitions. In [35], the exploration of data-reuse properties of FSMBA and memory bandwidth requirements is presented. Four levels of data-reuse (A/B/C/D) are defined according to the degree of data reuse for preceding data access. With the highest level of data reuse, one access for frame pixels can be achieved with a largest size of on-chip memory. Thus, there is a tradeoff between the size of on-chip memory and the reduced amount of data access.

2) Architectures for Shape Coding: The coding flow of MPEG-4 shape coding is illustrated in Fig. 2, which is composed of BME/BMC, CAE/CAD, and so on. A complete architecture design for shape coding is presented in [36]. By applying the dataflow optimization and data-reuse techniques, efficient bit-level architectures such as data-dispatch-based BME and configurable CAE are designed to accelerate a large amount of bit-level processing.

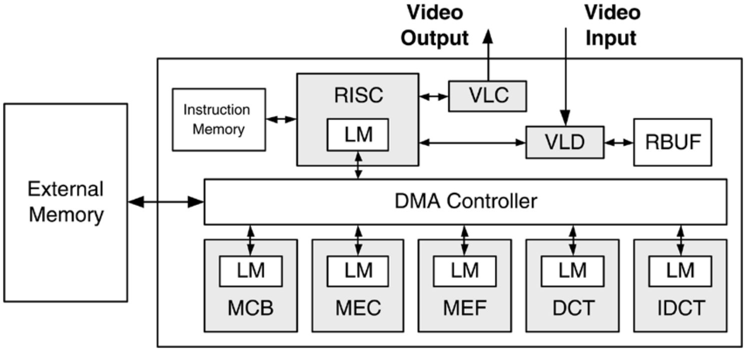

Fig. 3. Toshiba MPEG-4 video codec architecture [57].

Furthermore, both [37] and [38] discuss the CAE/CAD design, and a BME architecture is proposed in [39]. A look-ahead scheme for probability generation and shift registers in place of barrel shifters is applied in [40]. In [41], the frame memory and data transfer scheme for shape coding in an embedded system are discussed.

3) Architectures for VLC/VLD: The VLC/VLD are tasks for compressed bit stream packing/unpacking. Two categories of architectures have been proposed: tree-based and PLA-based. The tree-based architectures can be fully pipelined with short clock cycle, and are suitable for decoding multiple bit streams concurrently. However, the PLA-based architectures have better programmability [42], [43]. In [44], architectures with barrel shifters for VLC and VLD are proposed, wherein each codeword is encoded/decoded in one clock cycle. Based on these architectures, [45] proposes a solution for high data rate applications by packing the RUN amplitude and codeword together. [46] proposes an RVLC encoder/decoder design for MPEG-4 RVLC. [47] presents a complete VLC encoding system with header packing to form the final bit stream. In [48], a programmable VLC/VLD core is proposed to perform VLC and VLD within one core by sharing the VLC table memory and shifters. For low power applications, [49] proposes a fine-grained VLC table partitioning method to reduce the power consumption of table lookup.

There are many proposals for supporting multiple standards. A versatile VLD is proposed in [50], whereby both H.261 and JPEG bit stream can be decoded with the help of both PLA and CAM. For more standards, [51] proposes a VLC codeword grouping and symbol memory mapping approach to achieve a group-based VLC codec system with full VLC table programmability. Moreover, the RISC core has been adopted as an implementation platform. In [52] and [53], the instructions to enhance the RISC performance for VLD computation are proposed. The analysis of whole bit stream decoding and the dedicated bit stream parsing processor can be found in [54]–[56]. In summary, the trend of architecture design for VLC/VLD is toward intelligent and multistandard implementation.

C. System Architectures

There have already been numerous complete system architectures proposed for MPEG-4. Several of the most representative designs are selected as examples for discussion in this subsection [57]–[63]. The detailed system architectures, including module architectures, memory architecture, and interconnect architecture, are discussed below, and the comparisons of these designs are also made.

Takahashi *et al.* [57] present the first MPEG-4 video codec design, as shown in Fig. 3. Several dedicated module architectures are adopted for computation-intensive ME/MC, DCT/IDCT, and VLC/VLD, while an embedded RISC processor is included to provide flexibility for other tasks. Special local memory (LM) buffers are used to reduce the data access through off-chip frame buffer. A DMA controller that connects any two functional blocks acts as the interconnect architecture to provide higher power efficiency than a global bus architecture. The ME adopts a fast search algorithm with search range of $-31.5/+31.5$. This chip consumes 60 mW at 30 MHz for simple profile QCIF 10 frames/s encoding and decoding.

Based on [57], Nishikawa *et al.* [58] present a single-chip MPEG-4 video phone design with embedded DRAM. In addition to video codec, a speech codec, a multiplexer, and several I/O units are also included. This chip consumes 240 mW at 60 MHz for full system functionality, including simple profile QCIF 15 frames/s encoding and decoding of the video codec part. The integration of embedded DRAM significantly reduces a large amount of I/O power consumption caused by the data access through frame buffer.

Another single-chip MPEG-4 audiovisual design based on previous two designs [57], [58] is presented by Arakida *et al.* [62]. In addition to original modules of previous designs, a 5-GOPS adaptive filter engine is also included for post-processing. With more advanced VLSI technology as well as several low-power design techniques, this chip consumes 160 mW at 125 MHz for full system functionality, including simple profile CIF 15 frames/s encoding of the video codec part.

Hashimoto *et al.* [59] present the first MPEG-4 video codec design with the support of core profile. Similar to [57], several dedicated module architectures are adopted for computation-intensive ME/MC, DCT/IDCT, VLC/VLD, padding, and shape decoding, while a programmable DSP is used for other tasks. The embedded DRAM is integrated to reduce the I/O power consumption, and special local memory buffers are adopted to reduce the data access

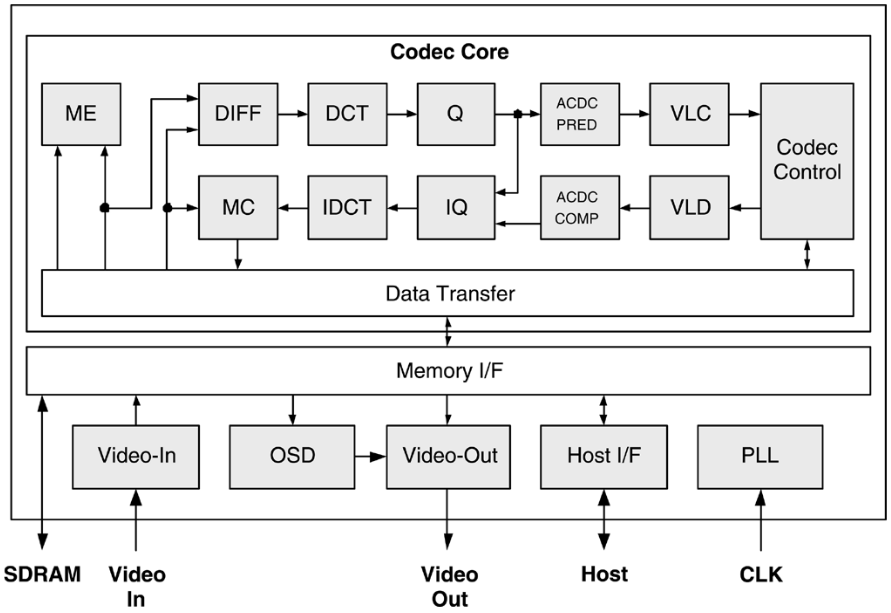

Fig. 4. Fujitsu MPEG-4 video codec architecture [61].

Table 1
Comparisons of MPEG-4 Architecture Parameters

Architecture	Category	Function	Profile Level	ME Search Algorithm	Frame Buffer
Takahashi [57]	Dedicated	V Codec	< SP@L1	Fast Search	External
Nishikawa [58]	Dedicated	AVS Codec	SP@L1	Fast Search	eDRAM
Hashimoto [59]	Dedicated	V Codec	CP@L1	N/A	eDRAM
Ohashi [60]	Dedicated	V Decoder	SP@L1	No	eSRAM
Nakayama [61]	Dedicated	V Codec	SP@L3	Fast Search	External
Arakida [62]	Dedicated	AVS Codec	SP@L2	Fast Search	eDRAM
Stolberg [63]	Programmable	V Codec	ASP@L5	Fast Search	External

Note: In Function part, V means Video and AVS means Aduio+Video+System

through frame buffer. A global bus is used as the interconnect architecture for higher flexibility. The details of ME parameters are not disclosed. This chip consumes 90 mW at 54 MHz for simple profile QCIF 15 frames/s encoding and decoding or core profile CIF 15 frames/s decoding.

Based on [59], Ohashi *et al.* [60] present a low-power MPEG-4 video decoder design. Rather than embedded DRAM, the embedded SRAM is integrated as the frame buffer for ease of future system-on-a-chip integration. Due to the elimination of unnecessary modules from previous design [59], this chip consumes only 11.1 mW at 27 MHz/54 MHz for simple profile QCIF 15 frames/s decoding.

A highly efficient MPEG-4 video codec design with full adaptation approach is proposed by Nakayama *et al.* [61], as shown in Fig. 4. Dedicated module architectures are adopted for all coding tasks including codec control. In addition, the dedicated data link is used as the interconnect architecture for maximum data communication efficiency. The data access as well as the size of local memory buffers are reduced due to the adoption of dedicated data link. The ME adopts a fast scene-adaptive search algorithm with search range of $-15.5/+15.5$. As a consequence of full adaptation, this chip consumes only 29 mW at 13.5 MHz for simple profile CIF 15 frames/s encoding and decoding.

Completely different from previous designs, Stolberg *et al.* [63] present a fully programmable multicore system-on-a-chip, comprised of a 16-way SIMD DSP core with a 2-D matrix memory, a 64-b VLIW DSP core with subword parallelism, and a 32-b RISC core. These programmable cores are connected with a global bus, and special local memory buffers are adopted to communicate data between each other. This chip consumes 3.5 W at 145 MHz for advanced simple profile D1 25 frames/s decoding or simple profile D1 25 frames/s encoding.

Table 1 shows the comparisons of several architecture parameters of aforementioned system architectures, and Table 2 shows the comparisons of several chip parameters of these architectures. From these two tables, three trends can be summarized as follows. First, except for the fully programmable solution [63], the specification of MPEG-4 video codec designs has increased year by year with rea-

Table 2
Comparisons of MPEG-4 Chip Parameters

Architecture	Technology (μm)	Frequency (MHz)	Power (mW)	Area (mm ^{2})	Video Encoding Capability
Takahashi [57]	0.3	30	60	81*	QCIF@10fps
Nishikawa [58]	0.25	60	240	117	QCIF@15fps
Hashimoto [59]	0.18	54	90	75	QCIF@15fps
Ohashi [60]	0.18	27/54	11.1	37	No
Nakayama [61]	0.18	13.5	29	28*	CIF@15fps
Arakida [62]	0.13	125	160	43	CIF@15fps
Stolberg [63]	0.18	145	3500	81*	D1@25fps

Note for *: The area does not include frame buffer

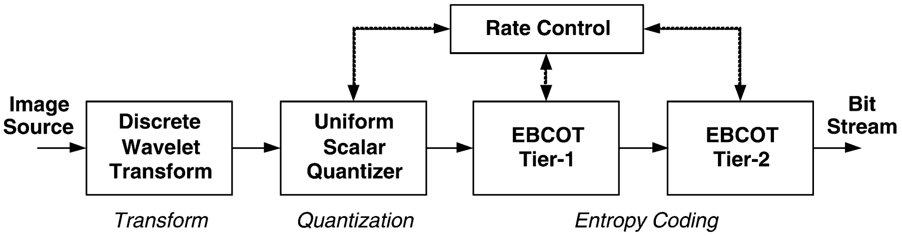

Fig. 5. Functional block diagram of JPEG 2000 still image coding.

sonable power consumption. Second, the integration of embedded DRAM or SRAM as the frame buffer has become the mainstream. Finally, fast search algorithms for ME are adopted by most designs due to stringent power constraints.

IV. HARDWARE ARCHITECTURES FOR JPEG 2000

JPEG 2000 is the latest still image coding standard. Fig. 5 shows its functional block diagram. More information on JPEG 2000 can be found in [64]–[66].

A. *Algorithm Analysis and Architecture Exploration*

According to the computational complexity analysis reported by [67] and [68], the computation-intensive tasks of JPEG 2000 are embedded block coding with optimized truncation (EBCOT) Tier-1 and discrete wavelet transform (DWT), which together contribute more than 80% of the overall complexity.

Since the computation-intensive DWT is a highly regular low-level task, dedicated architectures are suitable for efficient implementations. Due to a large amount of data access through the frame buffer incurred by multilevel DWT decomposition, the major design issue of DWT is data communication rather than computation. Special local memory buffers are generally adopted to reduce the data access, and the partition of a large-size image into small-size tiles to achieve independently processing is a typical approach to reduce the size of local memory buffers.

The most critical challenge of hardware implementations for JPEG 2000 is to design an efficient dedicated architecture for EBCOT Tier-1, which is another computation-intensive task of JPEG 2000. However, unlike DWT, the computations of EBCOT Tier-1 are characterized by irregular and data-dependent bit-level processing, which is composed of a multipass fractional bit-plane context scanning along with an adaptive binary arithmetic coding. Consequently, the major design issue of EBCOT Tier-1 is to increase the processing parallelism in order to achieve higher efficiency.

Compared with DWT and EBCOT Tier-1, the less demanding and probably unnecessary quantization task is insignificant. Due to its highly regular low-level characteristic, dedicated architectures can be adopted. The detailed architectures are not discussed.

The other task of JPEG 2000 is EBCOT Tier-2 with rate control, which is a less demanding but high-level task. Programmable architectures are, thus, suitable for this task. Although the computational complexity of EBCOT Tier-2 with rate control is not high, a large amount of data access and memory storage are still required for lossless compressed data. However, in lossy coding, most part of the data access and memory storage becomes redundant, and unnecessary computing resources are consequently wasted. Several techniques have been proposed to provide solutions for this problem [69]–[72].

B. *Module Architectures*

In this subsection, architectures for DWT and EBCOT Tier-1 are reviewed and discussed in detail.

1) Architectures for DWT: The architectures for DWT can be classified into three categories, including the convolution-based, lifting-based, and B-spline-based architectures. Since the multiresolution DWT can be viewed as cascades of several two-channel filter banks [73], in early research, the architectures for DWT are proposed based on convolution

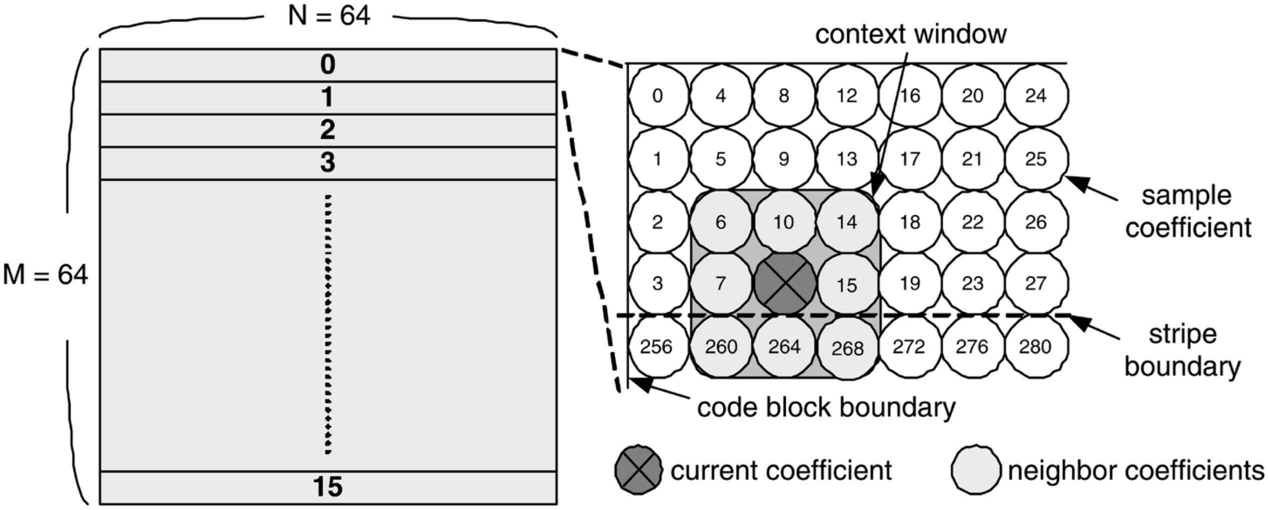

Fig. 6. Scan order and context window. Sample coefficients are scanned from top stripe to bottom stripe and column-by-column in stripe.

to construct the two-channel filter banks separately [74]. The filter bank can be implemented by using conventional techniques, such as polyphase decomposition, serial filter, and parallel filter. After the lifting scheme [75] and a novel lifting factorization method [76] were proposed by using perfect reconstruction property, lifting-based architectures became the mainstream due to the fewer multipliers and adders, as well as the regular structure. The inverse DWT can be derived easily from the lifting scheme of forward DWT, and they can even share the same architecture [77]. But the critical path of lifting-based architectures is potentially longer than that of the convolution-based architectures, and the flipping structure is proposed to address this problem without any hardware overhead [78]. However, DWT is only a subset of these two categories because any DWT filter bank must be factorized into the B-spline part and the distributed part [79]. The B-spline-based architectures have been proposed to minimize the number of multipliers based on B-spline factorization [80].

As indicated in the previous subsection, when extending a 1-D DWT module to 2-D DWT architecture, data access through the frame buffer becomes the major design issue [81]. The design tradeoff comes mainly from the size of local memory buffers and the reduced amount of data access. The 2-D DWT can be directly implemented without any local memory buffer by recursively performing 1-D DWT, but this requires the most data access. On the other hand, the 2-D DWT can be implemented with minimum data access by performing multilevel DWT decomposition totally inside [82], but a lot of local memory buffers are required. The systolic-parallel and parallel-parallel architectures [74] belong to this category. In addition, the one-level 2-D architecture [83] is proposed to balance the tradeoff between the above two extremes. Because the required size of local memory buffers is proportional to the line width of image or tile, these local memory buffers are called line buffers, and the architectures that adopt line buffers are called line-based architectures. In addition to line-based architectures, block-based architectures are also proposed to address the data access issue by implementing 2-D DWT block-by-block [84].

2) Architectures for EBCOT Tier-1: EBCOT Tier-1 is composed of two parts: the context formation (CF) for multipass fractional bit-plane context scanning and the arithmetic encoder (AE) for adaptive binary arithmetic coding. The CF scans the code block in a specific order and generates one context-decision (CXD) pair for each bit, and the adaptive binary AE encodes the CXD pairs into embedded bit stream. Coefficients in code blocks are represented as sign-magnitude bit-planes. Each code block is scanned in a bit-plane by bit-plane order from MSB to LSB. To improve embedding, each bit-plane is further divided into three fractional bit-planes: significance propagational pass (Pass1), magnitude refinement pass (Pass2), and cleanup pass (Pass3). Each pass involves a stripe-based scan of size $4 \times N$, as shown in Fig. 6. When scanning a pass, CXD pairs are formed by the CF according to the coding states of the sample coefficient itself and its neighbors in context window, which is also as shown in Fig. 6. More details of EBCOT Tier-1 are presented in [85] and [86].

As indicated in the previous subsection, the major design issue of EBCOT Tier-1 is to increase the processing parallelism for irregular and data-dependent bit-level processing. The most straightforward approach to increase processing parallelism is to exploit code block parallelism due to the independence between code blocks, by which several code blocks can be parallel processed by individual processing elements. However, there is still a great deal of redundant processing within a code block. There are two coding modes of EBCOT Tier-1: default serial mode and parallel mode. The parallel mode enables more parallelism than the default serial mode, but it comes with a little bit of performance degradation. Several techniques have been proposed to reduce the processing cycles in either default serial mode or parallel mode, including column parallelism, pass parallelism, and bit-plane parallelism. In addition to processing parallelism, the memory usage for state variables of CF is another design issue of EBCOT Tier-1. Below, the details of these techniques and corresponding architectures are discussed.

Lian *et al.* [68] propose the first architecture for EBCOT Tier-1 in default serial mode. In the conventional approach [87], each sample coefficient must be scanned three times for the three coding passes. However, only a single scan on each sample coefficient is effective and the other two scans are redundant. In [68], two techniques are proposed to speed up the coding process: sampling skipping (SS) and group-of-column skipping (GOCS). Through parallel checking a column, SS skips no-operation samples and scans only samples that need to be coded. GOCS skips one group of columns at a time if all the columns are no-operation columns. By applying both skipping techniques, the required processing cycles of the architecture can be reduced to 40% of that of the conventional approach. In addition, a four-stage pipelined AE architecture is also presented in [68]. A probability look-ahead technique is used to break the feedback loop, and pipelined architecture is, thus, feasible. Chen *et al.* [88] adopt the skipping techniques of [68] and further propose a multiple column skipping (MCOLS) technique to improve the skipping ratio. Hsiao *et al.* [89] propose a memory-saving architecture for CF. Correlations between the three state variables of CF are exploited to reduce the memory size to 80% of that of [68].

A pass-parallel architecture is proposed by Chiang *et al.* [90] to process three coding passes by scanning each bit-plane once. In order to achieve pass-parallel coding, the EBCOT Tier-1 must be coded in parallel mode. This pass-parallel architecture can further improve the processing speed by 25% over the SS technique in [68]. Furthermore, the size of state variable memory can be reduced due to an unnecessary state variable in parallel mode. In order to decrease the hardware cost, a pass-switching AE architecture is also proposed. This AE architecture can process three coding passes in a bit-plane in parallel because each sample coefficient must belong to one and only one of the three coding passes. Therefore, the AE processes only one of the three coding passes at a time, switching between them if needed. This AE architecture requires only one processing element and three suits of coding status registers for the three coding passes.

Yamauchi *et al.* [91] propose to increase the processing speed by two-bit-plane with three-pass parallel processing. A four-symbol parallel AE architecture is also proposed to together achieve a high degree of parallelism for high performance and low power consumption.

A novel context formation algorithm is proposed by Fang *et al.* [92] to exploit the maximum parallelism of EBCOT Tier-1. Instead of using state variables, the word-level algorithm generates contexts by examining the value and the coding pass of the first nonzero bit of each coefficient in the context window. Based on this algorithm, a parallel EBCOT Tier-1 architecture is proposed in [93]. In addition to maximum processing parallelism, this parallel architecture only needs to buffer one line of coefficients of a code block.

Table 3
Comparisons of EBCOT Tier-1 Architectures

Architecture	Speed (cycles)	Logic Gate Count	Memory Size (bits)	Memory Access
Conventional [87]	$(3n-2) \times N^2$	N/A	N/A	N/A
Lian [68]	$1.3n \times N^2$	19000	$5 \times N^2$	$n \times N^2$
Hsiao [89]	$1.3n \times N^2$	21589	$4 \times N^2$	$n \times N^2$
Chiang [90]	$n \times N^2$	23927	$4 \times N^2$	$n \times N^2$
Yamauchi [91]	$0.5n \times N^2$	N/A	$4 \times N^2$	$0.5n \times N^2$
Fang [93]	N^2	91758	$12 \times N$	N^2

Note: The Memory Size is the required memory size of CF, and the Memory Access is the amount of memory access through the code block memory of CF

Table 3 shows the comparisons of aforementioned EBCOT Tier-1 architectures. The code block size is set to be $N \times N$ with n nonzero bit-planes to generalize the comparisons. As can be seen, the parallel architectures [90], [91], [93] achieve better processing speed than the default serial architectures [68], [89]. Among all the architectures, Fang's architecture [93] achieves the best processing speed, as well as minimum memory size and memory access, although it has the largest logic gate count.

C. *System Architectures*

There have been several system architectures proposed for JPEG 2000 so far. Some of them are complete system architectures, including DSPworx [94], Yamauchi [91], Andra [95], ADI [96], and Fang [97], while some of them are hardware accelerators only and need to cooperate with a host processor, such as ALMA [98] and AMPHION [99].

In order to obtain highly efficient hardware solutions, all of the system architectures adopt several dedicated module architectures for computation-intensive tasks such as DWT, EBCOT Tier-1, and quantization. The complete system architectures [91], [94]–[97] even include dedicated module architectures for EBCOT Tier-2 with rate control. As for memory architecture, all system architectures heavily rely on various special local memory buffers in order to effectively reduce the large amount of data access through off-chip frame buffer. In addition, the dedicated data link is adopted as the interconnect architecture for all system architectures to achieve highly efficient data communication. Below, three selected designs are discussed in detail, and the comparisons of all designs are also made.

Amphion [99] reports a JPEG 2000 codec accelerator design, as shown in Fig. 7. In order to increase the throughput, three entropy codecs are adopted to encode or decode three code blocks in parallel. This design can achieve 60/20 M samples/s encoding/decoding rate at 180 MHz.

A complete JPEG 2000 image processor system-on-a-chip is proposed by Yamauchi *et al.* [91]. It uses block-based 2-D DWT to minimize the size of local memory buffers for DWT, but comes with increased data access. For EBCOT Tier-1, it encodes two bit-planes, three coding passes, and four symbols in parallel to increase the throughput by a factor of 24 compared with the conventional approach. As a result of these techniques, this chip can encode 20.7 mega (M) samples/s at 27 MHz. In addition, a one-pass code size controlling method is used to predict the code size for rate control.

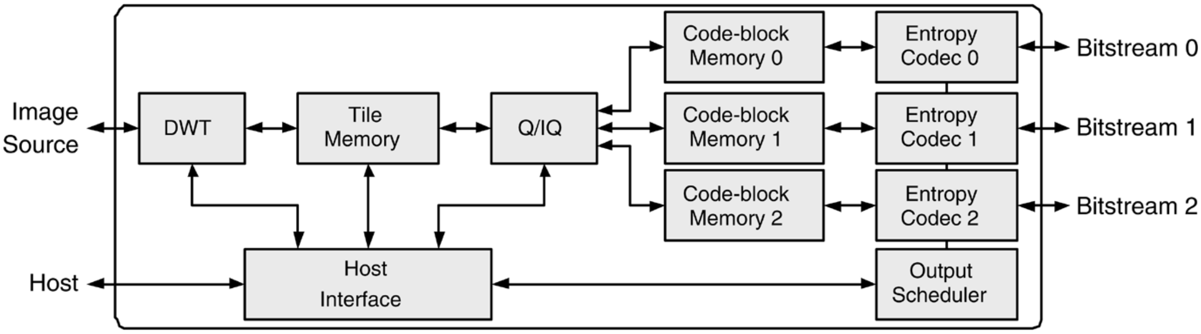

Fig. 7. Amphion JPEG 2000 codec hardware accelerator [99].

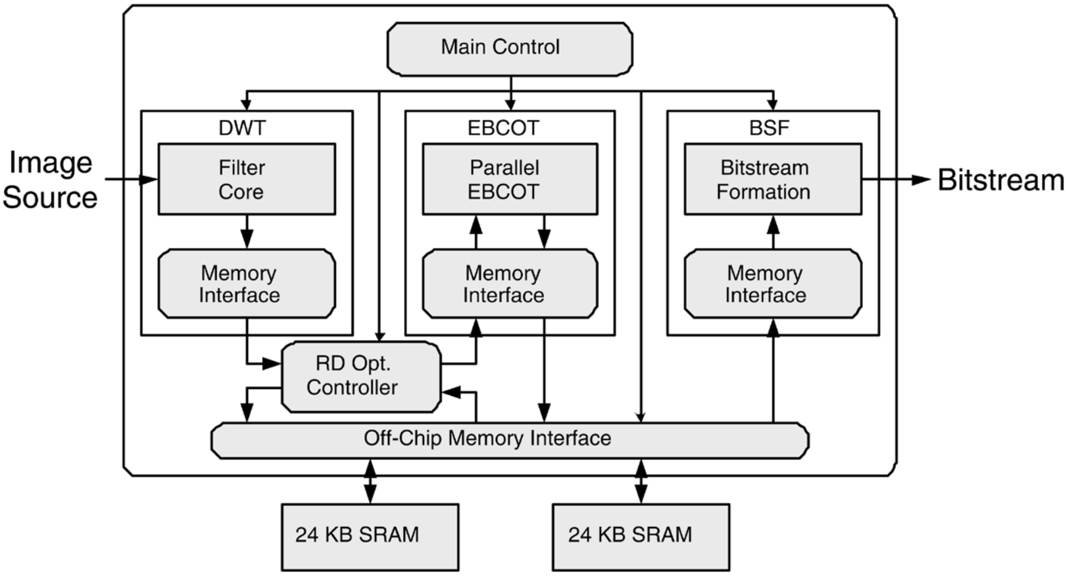

Fig. 8. Fang's JPEG 2000 encoder architecture [97].

Table 4
Comparison of JPEG 2000 Architecture Parameters

Architecture	Function	DWT Filter	DWT Level	Tile Size	CB Size
DSPworx [94]	Codec	5/3 + 9/7	5	256x256	N/A
ALMA [98]	Encoder*	5/3 + 9/7	N/A	256x256	64x64
AMPHION [99]	Codec*	5/3 + 9/7	5	128x128	N/A
Yamauchi [91]	Encoder	5/3 + 9/7	2	1024x512	32x32
Andra [95]	Encoder	5/3 + 9/7	5	128x128	32x32
ADI [96]	Codec	5/3 + 9/7	6	4096x2048	N/A
Fang [97]	Encoder	5/3	2	128x128	64x64

Note for *: The design is a hardware accelerator only but not a complete system architecture

Based on parallel EBCOT Tier-1 architecture [93], Fang *et al.* [97] propose a JPEG 2000 encoder architecture, as shown in Fig. 8. Unlike conventional architectures, this architecture does not require any code block memory due to the use of parallel EBCOT Tier-1 architecture. This design is fully pipelined, and the throughput rate is equal to the operating frequency. Another important feature of this design is the rate-distortion optimized (RD Opt.) controller, which realizes the precompression rate-distortion optimization algorithm [72]. By using this algorithm, redundant computation and data access, as well as the bit stream buffer are eliminated, which leads to low power and small area.

Table 4 shows the comparison of several architecture parameters of all system architectures. Among these parameters, the tile size may be the most important, since it affects the architectures for DWT. Line-based 2-D DWT can minimize the off-chip data access, but it requires a large-size on-chip memory, which is proportional to the tile size. On the other hand, the memory size is irrelevant to tile size for block-based 2-D DWT. Thus, block-based 2-D DWT is preferred for designs with large tile size. The DWT filter type and decomposition level as well as EBCOT Tier-1 code block size have some effects on coding efficiency. The 9/7 filter requires more processing elements than the 5/3 filter, while

Table 5
Comparison of JPEG 2000 Chip Parameters

Architecture	Technology (μm)	Area (mm ^{2})	Frequency (MHz)	Throughput (MS/s)
DSPworx [94]	0.18	289	200	50
AMPHION [99]	0.18	5.4	120	60
Yamauchi [91]	0.25	13.7*	27.4	21
ADI [96]	0.18	144	N/A	65
Fang [97]	0.25	5.5*	81	81

Note for *: The size is the core size but not die size

higher decomposition level and larger code block size lead to larger memory size.

Table 5 compares several chip parameters of actually implemented system architectures. It is readily seen that the throughput of Fang's architecture is equal to the operating frequency. Moreover, the area of Fang's architecture is also the smallest, since it uses parallel EBCOT Tier-1 architecture. In other architectures, parallel processing of code blocks, which leads to large area, is used to increase the throughput. Another way to increase the throughput is to raise the operating frequency. However, this would result in higher power consumption.

V. FURTHER PERSPECTIVES

Efficient system architectures for image and video coding are composed of efficient module architectures for data computation, together with efficient memory architecture and interconnect architecture for data communication. In this paper, state-of-the-art hardware architectures for emerging MPEG-4 video coding and JPEG 2000 still image coding have been reviewed as design examples, and special approaches exploited to improve efficiency have also been discussed.

Although the emerging MPEG-4 and JPEG 2000 are capable of providing improved coding efficiency and additional functionalities, the demands on much better coding efficiency and much richer functionalities for image and video coding applications are ever-increasing. Therefore, advanced coding algorithms are continuing to be developed vigorously, and the standardization of new-generation coding standards will keep emerging. In order to provide better coding efficiency and richer functionalities, more complicated coding tools must be adopted by advanced coding algorithms, which will further inevitably increase the computational complexity. Even if the advances in VLSI technology continue to provide more processing capability without increased hardware cost and power consumption, there is still a strong need to explore highly efficient hardware architectures for advanced image and video coding.

For example, the H.264/AVC recently developed by the Joint Video Team (JVT) of ITU-T VCEG and ISO/IEC MPEG is an advanced new-generation video coding standard, which provides a significant improvement in coding efficiency compared with various preceding standards. It can achieve essentially the same reproduction quality as previous standards, while typically requiring 60% or less of the bit-rate [100], [101]. However, the improvement in coding efficiency by H.264/AVC comes, not surprisingly, with a high degree of computational complexity. An estimation reports that the complexity of H.264/AVC encoder grows an order of magnitude higher than that of the MPEG-4 encoder [102]. Even with highly optimized encoder implementation, the complexity of H.264/AVC is still about 3.4 times more than that of H.263 [103]. Therefore, this advanced video coding standard has a significant impact on computational complexity, and it will, consequently, present new challenges for hardware architecture design. Special architectural approaches introduced by this paper are also applicable to advanced image and video coding in the future. Exploiting these approaches in a more efficient way is a critical factor to cope with the new challenges.

REFERENCES

[1] B. G. Haskell, P. G. Howard, Y. A. LeCun, A. Puri, J. Ostermann, M. R. Civanlar, L. Rabiner, L. Bottou, and P. Haffner, "Image and video coding—Emerging standards and beyond," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 7, pp. 814–837, Nov. 1998.

[2] "Information technology—Digital compression and coding of continuous-tone still images," Int. Standards Org./Int. Electrotech. Comm. (ISO/IEC) and Int. Telecommun. Union-Telecommun. (ITU-T), ISO/IEC 10918-1 and ITU-T Recommendation T.81, 1994.

[3] "JPEG 2000 part 1 final draft international standard," Int. Standards Org./Int. Electrotech. Comm. (ISO/IEC), ISO/IEC FDIS15444-1, Dec. 2000.

[4] "Information technology—Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s: Video," Int. Standards Org./Int. Electrotech. Comm. (ISO/IEC), ISO/IEC 11172-2, 1994.

[5] "Information technology—Generic coding of moving pictures and associated audio information: Video," Int. Standards Org./Int. Electrotech. Comm. (ISO/IEC) and Int. Telecommun. Union-Telecommun. (ITU-T), 13818-2 and ITU-T Recommendation H.262, 1996.

[6] "Information technology—Coding of audio-visual objects—Part 2: Visual," Int. Standards Org./Int. Electrotech. Comm. (ISO/IEC), ISO/IEC 14496-2, 1999.

[7] "Video codec for audiovisual services at $p \times 64$ kbit/s," Int. Telecommun. Union-Telecommun. (ITU-T), Geneva, Switzerland, Recommendation H.261, 1993.

[8] "Video coding for low bit rate communication," Int. Telecommun. Union-Telecommun. (ITU-T), Geneva, Switzerland, Recommendation H.263, 1998.

[9] "Draft ITU-T Recommendation and final draft International Standard of Joint Video Specification," Joint Video Team, Int. Telecommun. Union-Telecommun. (ITU-T) and Int. Standards Org./Int. Electrotech. Comm. (ISO/IEC), ITU-T Recommendation H.264 and ISO/IEC 14496-10 AVC, May 2003.

[10] P. Pirsch, N. Demassieux, and W. Gehrke, "VLSI architectures for video compression—A survey," *Proc. IEEE*, vol. 83, no. 2, pp. 220–246, Feb. 1995.

[11] P. Pirsch and H.-J. Stolberg, "VLSI implementations of image and video multimedia processing systems," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 7, pp. 878–891, Nov. 1998.

[12] I. Kuroda and T. Nishitani, "Multimedia processors," *Proc. IEEE*, vol. 86, no. 6, pp. 1203–1221, Jun. 1998.

[13] A. Dasu and S. Panchanathan, "A survey of media processing approaches," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 8, pp. 633–645, Aug. 2002.

[14] V. Lappalainen, T. D. Hamalainen, and P. Liuha, "Overview of research efforts on media ISA extensions and their usage in video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 8, pp. 660–670, Aug. 2002.

[15] T. Sikora, "The MPEG-4 video standard verification model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 1, pp. 19–31, Feb. 1997.

[16] T. Ebrahimi and C. Horne, "MPEG-4 natural video coding—An overview," *Signal Process. Image Commun.*, vol. 15, no. 4–5, pp. 365–385, Jan. 2000.

[17] P. M. Kuhn and W. Stechele, "Complexity analysis of the emerging MPEG-4 standard as a basis for VLSI implementation," in *Proc. Int. Conf. Visual Communications and Image Processing*, 1998, pp. 498–509.

[18] H. C. Chang, L. G. Chen, M. Y. Hsu, and Y. C. Chang, "Performance analysis and architecture evaluation of MPEG-4 video codec system," in *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 2, 2000, pp. 449–452.

[19] H.-J. Stolberg, M. Berekovic, P. Pirsch, and H. Runge, "The MPEG-4 advanced simple profile—A complexity study," in *Proc. Workshop and Exhibition on MPEG-4*, 2001, pp. 33–36.

[20] S. Y. Chien, C. Y. Chen, W. M. Chao, Y. W. Huang, and L. G. Chen, "Analysis and hardware architecture for global motion estimation in MPEG-4 advanced simple profile," in *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 2, 2003, pp. 720–723.

[21] C. W. Hsu, Y. C. Chang, W. M. Chao, and L. G. Chen, "Hardware-oriented optimization and block-level architecture design for MPEG-4 FGS encoder," in *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 2, 2003, pp. 784–787.

[22] T. Komarek and P. Pirsch, "Array architectures for block matching algorithms," *IEEE Trans. Circuits Syst.*, vol. 36, no. 2, pp. 1301–1308, Oct. 1989.

[23] L. D. Vos and M. Stegherr, "Parameterizable VLSI architectures for the full-search block-matching algorithm," *IEEE Trans. Circuits Syst.*, vol. 36, no. 2, pp. 1309–1316, Oct. 1989.

[24] K. M. Yang, M. T. Sun, and L. Wu, "A family of VLSI designs for the motion compensation block-matching algorithm," *IEEE Trans. Circuits Syst.*, vol. 36, no. 2, pp. 1317–1325, Oct. 1989.

[25] C. H. Hsieh and T. P. Lin, "VLSI architecture for block-matching motion estimation algorithm," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 2, no. 2, pp. 169–175, Jun. 1992.

[26] H. Yeo and Y. H. Hu, "A novel modular systolic array architecture for full-search block matching motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, no. 5, pp. 407–416, Oct. 1995.

[27] Y. K. Lai and L. G. Chen, "A data-interlacing architecture with two-dimensional data-reuse for full-search block-matching algorithm," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 2, pp. 124–127, Apr. 1998.

[28] Y. H. Yeh and C. Y. Lee, "Cost-effective VLSI architectures and buffer size optimization for full-search block matching algorithms," *IEEE Trans. Very Large Scale (VLSI) Syst.*, vol. 7, no. 3, pp. 345–358, Sep. 1999.

[29] Y. S. Jehng, L. G. Chen, and T. D. Chiueh, "An efficient and simple VLSI tree architecture for motion estimation algorithms," *IEEE Trans. Signal Process.*, vol. 41, no. 2, pp. 889–900, Feb. 1993.

[30] H. M. Jong, L. G. Chen, and T. D. Chiueh, "Parallel architectures for 3-step hierarchical search block-matching algorithm," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, no. 4, pp. 407–416, Aug. 1994.

[31] S. Dutta and W. Wolf, "A flexible parallel architecture adopted to block-matching motion estimation algorithms," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 1, pp. 74–86, Feb. 1996.

[32] 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 J. Solid-State Circuits*, vol. 32, no. 11, pp. 1807–1816, Nov. 1997.

[33] M. Takahashi, T. Nishikawa, M. Hamada, T. Takayanagi, H. Arakida, N. Machida, H. Yamamoto, T. Fujiyoshi, Y. Ohashi, O. Yamagishi, T. Samata, A. Asano, T. Terazawa, K. Ohmori, Y. Watanabe, H. Nakamura, S. Minami, T. Kuroda, and T. Furuyama, "A 60-mW MPEG4 video codec using clustered voltage scaling with variable supply-voltage scheme," *IEEE J. Solid-State Circuits*, vol. 33, no. 11, pp. 1772–1780, Nov. 1998.

[34] J. H. Lee, K. W. Lim, B. C. Song, and J. B. Ra, "A fast multi-resolution block matching algorithm and its VLSI architecture for low bit-rate video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 12, pp. 1289–1301, Dec. 2001.

[35] J. C. Tuan, T. S. Chang, and C. W. Jen, "On the data reuse and memory bandwidth analysis for full-search block-matching VLSI architecture," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 1, pp. 61–72, Jan. 2002.

[36] H. C. Chang, Y. C. Chang, Y. C. Wang, W. M. Chao, and L. G. Chen, "VLSI architecture design of MPEG-4 shape coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 9, pp. 741–751, Sep. 2002.

[37] M. Berekovic, K. Jacob, and P. Pirsch, "Architecture of a hardware module for MPEG-4 shape decoding," in *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 1, 1999, pp. 157–160.

[38] D. Gong and Y. He, "An efficient architecture for real-time content-based arithmetic coding," in *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 3, 2000, pp. 515–518.

[39] T. H. Tsai and C. P. Chen, "An efficient binary motion estimation algorithm and its architecture for MPEG-4 shape coding," in *Proc. Int. Symp. Circuits and Systems*, vol. 2, 2003, pp. 496–499.

[40] J. Thinakaran, D. J. Ho, and N. Ling, "Fast shape decoding for MPEG-4 video," in *Proc. IEEE Workshop Signal Processing Systems*, 2000, pp. 110–119.

[41] K. B. Lee, N. Y. C. Chang, H. Y. Chin, H. J. Hsu, and C. W. Jen, "Optimal frame memory and data transfer scheme for MPEG-4 shape coding," in *Proc. IEEE Int. Conf. Consumer Electronics*, 2003, pp. 164–165.

[42] S. F. Chang and D. G. Messerschmitt, "Designing high-throughput VLC decoder, Part I—Concurrent VLSI architectures," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 2, no. 2, pp. 187–196, Jun. 1992.

[43] H. D. Lin and D. G. Messerschmitt, "Designing a high-throughput VLC decoder, Part II—Parallel decoding methods," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 2, no. 2, pp. 197–206, Jun. 1992.

[44] S. M. Lei and M. T. Sun, "An entropy coding system for digital HDTV applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 1, no. 1, pp. 147–155, Mar. 1991.

[45] H. C. Chang, L. G. Chen, Y. C. Chang, and S. C. Huang, "A VLSI architecture design of VLC encoder for high data rate video/image coding," in *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 4, 1999, pp. 398–401.

[46] M. Novell and S. Molloy, "VLSI implementation of a reversible variable length encoder/decoder," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 4, 1999, pp. 1969–1972.

[47] J. Y. Yang, Y. Lee, H. Lee, and J. Kim, "A variable length coding ASIC chip for HDTV video encoders," *IEEE Trans. Consum. Electron.*, vol. 43, no. 3, pp. 633–638, Aug. 1997.

[48] Y. Fukuzawa, K. Hasegawa, H. Hanaki, E. Iwata, and T. Yamazaki, "A programmable VLC core architecture for video compression DSP," in *Proc. IEEE Workshop Signal Processing Systems*, 1997, pp. 469–478.

[49] S. H. Cho, T. Xanthopoulos, and A. P. Chandrakasan, "A low power variable length decoder for MPEG-2 based on nonuniform fine-grain table partitioning," *IEEE Trans. Very Large Scale (VLSI) Syst.*, vol. 7, no. 2, pp. 249–257, Jun. 1999.

[50] K. M. Yang, H. Fujiwara, T. Sakaguchi, and A. Shimazu, "VLSI architecture design of a versatile variable length decoding chip for real-time video codecs," in *Proc. IEEE Conf. Computer and Communication Systems*, vol. 2, 1990, pp. 551–554.

[51] B. J. Shieh, Y. S. Lee, and C. Y. Lee, "A new approach of group-based VLC codec system with full table programmability," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 2, pp. 210–221, Feb. 2001.

[52] Y. Zhang and K. K. Ma, "An embedded RISC core design for variable length coding," in *Proc. IEEE Int. Conf./Exhibition High Performance Computing in the Asia–Pacific Region*, vol. 1, 2000, pp. 361–365.

[53] M. Berekovic, H.-J. Stolberg, M. B. Kulaczewski, P. Pirsch, H. Moller, H. Runge, J. Kneip, and B. Stabernack, "Instruction set extensions for MPEG-4 video," *J. VLSI Signal Process.*, vol. 23, no. 1, pp. 27–49, Oct. 1999.

[54] H. C. Chang, Y. C. Chang, Y. B. Tsai, C. P. Fan, and L. G. Chen, "MPEG-4 video bitstream structure analysis and its parsing architecture design," in *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 2, 2000, pp. 184–187.

[55] Y. C. Chang, H. C. Chang, and L. G. Chen, "Design and implementation of a bitstream parsing coprocessor for MPEG-4 video system-on-chip solution," in *Proc. IEEE Int. Symp. VLSI Technology, Systems, and Applications*, 2001, pp. 188–191.

[56] Y. C. Chang, C. C. Huang, W. M. Chao, and L. G. Chen, "An efficient embedded bitstream parsing processor for MPEG-4 video decoding system," in *Proc. Int. Symp. VLSI Technology, Systems, and Applications*, 2003, pp. 168–171.

[57] M. Takahashi, M. Hamada, T. Nishikawa, H. Arakida, Y. Tsuboi, T. Fujita, F. Hatori, S. Mita, K. Suzuki, A. Chiba, T. Terazawa, F. Sano, Y. Watanabe, H. Momose, K. Usami, M. Igarashi, T. Ishikawa, M. Kanazawa, T. Kuroda, and T. Furuyama, "A 60 mW MPEG4 video codec using clustered voltage scaling with variable supply-voltage scheme," in *Dig. Tech. Papers IEEE Int. Solid-State Circuits Conf.*, 1998, pp. 36–37.

[58] T. Nishikawa, M. Takahashi, M. Hamada, T. Takayanagi, H. Arakida, N. Machida, H. Yamamoto, T. Fujiyoshi, Y. Maisumoto, O. Yamagishi, T. Samata, A. Asano, T. Terazawa, K. Ohmori, J. Shirakura, Y. Watanabe, H. Nakamura, S. Minami, T. Kuroda, and T. Furuyama, "A 60 MHz 240 mW MPEG-4 video-phone LSI with 16 Mb embedded DRAM," in *Dig. Tech. Papers IEEE Int. Solid-State Circuits Conf.*, 2000, pp. 230–231.

[59] T. Hashimoto, S. Kuromaru, M. Matsuo, K. Yasuo, T. Mori-iwa, K. Ishida, S. Kajita, M. Ohashi, M. Toujima, T. Nakamura, M. Hamada, T. Yonezawa, T. Kondo, K. Hashimoto, Y. Sugisawa, H. Otsuki, M. Arita, H. Nakajima, H. Fujimoto, J. Michiyama, Y. Lizuka, H. Komori, S. Nakatani, H. Toida, T. Takahashi, H. Ito, and T. Yukitake, "A 90 mW MPEG4 video codec LSI with the capability for core profile," in *Dig. Tech. Papers IEEE Int. Solid-State Circuits Conf.*, 2001, pp. 140–141.

[60] M. Ohashi, T. Hashimoto, S. Kuromaru, M. Matsuo, T. Mori-iwa, M. Hamada, Y. Sugisawa, M. Arita, H. Tomita, M. Hoshino, H. Miyajima, T. Nakamura, K. Ishida, T. Kimura, Y. Kohashi, T. Kondo, A. Inoue, H. Fujimoto, K. Watada, T. Fukunaga, T. Nishi, H. Ito, and J. Michiyama, "A 27 MHz 11.1 mW MPEG-4 video decoder LSI for mobile application," in *Dig. Tech. Papers IEEE Int. Solid-State Circuits Conf.*, 2002, pp. 366–367.

[61] H. Nakayama, T. Yoshitake, H. Komazaki, Y. Watanabe, H. Araki, K. Morioka, J. Li, L. Peilin, S. Lee, H. Kubosawa, and Y. Otobe, "An MPEG-4 video LSI with an error-resilient codec core based on a fast motion estimation algorithm," in *Dig. Tech. Papers IEEE Int. Solid-State Circuits Conf.*, 2002, pp. 368–369.

[62] H. Arakida, M. Takahashi, Y. Tsuboi, T. Nishikawa, H. Yamamoto, T. Fujiyoshi, Y. Kitasho, Y. Ueda, M. Watanabe, T. Fujita, T. Terazawa, K. Ohmori, M. Koana, H. Nakamura, E. Watanabe, H. Ando, T. Aikawa, and T. Furuyama, "A 160 mW, 80 nA standby, MPEG-4 audiovisual LSI with 16 mb embedded DRAM and a 5 GOPS adaptive post filter," in *Dig. Tech. Papers IEEE Int. Solid-State Circuits Conf.*, 2003, pp. 1–11.

[63] H.-J. Stolberg, S. Moch, L. Friebe, A. Dehnhardt, M. Kulaczewski, M. Berekovic, and P. Pirsch, "An SoC with two multimedia DSP's and a RISC core for video compression and surveillance," in *Dig. Tech. Papers IEEE Int. Solid-State Circuits Conf.*, 2004, pp. 330–331.

[64] C. Christopoulos, A. Skodras, and T. Ebrahimi, "The JPEG2000 still image coding system: An overview," *IEEE Trans. Consum. Electron.*, vol. 46, no. 4, pp. 1103–1127, Nov. 2000.

[65] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The JPEG 2000 still image compression standard," *IEEE Signal Process. Mag.*, vol. 18, no. 5, pp. 36–58, Sep. 2001.

[66] D. S. Taubman and M. W. Marcellin, "JPEG2000: Standard for interactive imaging," *Proc. IEEE*, vol. 90, no. 8, pp. 1336–1357, Aug. 2002.

[67] M. D. Adams and F. Kossentini, "JasPer: A software-based JPEG-2000 codec implementation," in *Proc. IEEE Int. Conf. Image Processing*, vol. 2, 2000, pp. 53–56.

[68] C. J. Lian, K. F. Chen, H. H. Chen, and L. G. Chen, "Analysis and architecture design of block-coding engine for EBCOT in JPEG 2000," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 3, pp. 219–230, Mar. 2003.

[69] T. H. Chang, L. L. Chen, C. J. Lian, H. H. Chen, and L. G. Chen, "Computation reduction technique for lossy JPEG2000 encoding through EBCOT tier-2 feedback processing," in *Proc. IEEE Int. Conf. Image Processing*, 2002, pp. 85–88.

[70] T. H. Chang, C. J. Lian, H. H. Chen, J. Y. Chang, and L. G. Chen, "Effective hardware-oriented technique for the rate control of JPEG2000 encoding," in *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 2, 2003, pp. 684–687.

[71] Y. M. Yeung, O. C. Au, and A. Chang, "Efficient rate control technique for JPEG2000 image coding using priority scanning," in *Proc. IEEE Int. Conf. Multimedia and Expo*, vol. 3, 2003, pp. 277–280.

[72] Y. W. Chang, H. C. Fang, C. J. Lian, and L. G. Chen, "Novel pre-compression rate-distortion optimization algorithm for JPEG 2000," in *Proc. Int. Conf. Visual Communications and Image Processing*, 2004.

[73] S. G. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 7, pp. 674–693, Jul. 1989.

[74] C. Chakrabarti, M. Vishwanath, and R. M. Owens, "Architectures for wavelet transforms: A survey," *J. VLSI Signal Process.*, vol. 14, pp. 171–192, 1996.

[75] W. Sweldens, "The lifting scheme: A custom-design construction of biorthogonal wavelets," *Appl. Comput. Harmonic Anal.*, vol. 3, no. 15, pp. 186–200, 1996.

[76] I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting steps," *J. Fourier Anal. Appl.*, vol. 4, pp. 247–269, 1998.

[77] K. Andra, C. Chakrabarti, and T. Acharya, "A VLSI architecture for lifting-based forward and inverse wavelet transform," *IEEE Trans. Signal Process.*, vol. 50, no. 4, pp. 966–977, Apr. 2002.

[78] C. T. Huang, P. C. Tseng, and L. G. Chen, "Flipping structure: An efficient VLSI architecture for lifting-based discrete wavelet transform," *IEEE Trans. Signal Process.*, to be published.

[79] M. Unser and T. Blu, "Wavelet theory demystified," *IEEE Trans. Signal Process.*, vol. 51, no. 2, pp. 470–483, Feb. 2003.

[80] C. T. Huang, P. C. Tseng, and L. G. Chen, "VLSI architecture for forward discrete wavelet transform based on B-spline factorization," *J. VLSI Signal Process.*, to be published.

[81] N. D. Zervas, G. P. Anagnostopoulos, V. Spiliotopoulos, Y. Andreopoulos, and C. E. Goutis, "Evaluation of design alternatives for the 2-D-discrete wavelet transform," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 12, pp. 1246–1262, Dec. 2001.

[82] C. Chrysafis and A. Ortega, "Line-based, reduced memory, wavelet image compression," *IEEE Trans. Image Process.*, vol. 9, no. 3, pp. 378–389, Mar. 2000.

[83] P. C. Wu and L. G. Chen, "An efficient architecture for two-dimensional discrete wavelet transform," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 4, pp. 536–545, Apr. 2001.

[84] W. Jiang and A. Ortega, "Lifting factorization-based discrete wavelet transform architecture design," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 5, pp. 651–657, May 2001.

[85] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Trans. Image Process.*, vol. 9, no. 7, pp. 1158–1170, Jul. 2000.

[86] D. Taubman, E. Ordentlich, M. Weinberger, and G. Seroussi, "Embedded block coding in JPEG 2000," *Signal Process. Image Commun.*, vol. 17, no. 1, pp. 49–72, Jan. 2002.

[87] "JPEG 2000 verification model 7.0 (technical description) ," Int. Standards Org./Int. Electrotech. Comm. (ISO/IEC), ISO/IEC JTC1/SC29/WG1 N1684, Apr. 2000.

[88] H. H. Chen, C. J. Lian, T. H. Chang, and L. G. Chen, "Analysis of EBCOT decoding algorithm and its VLSI implementation for JPEG 2000," in *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 4, 2002, pp. 329–332.

[89] Y. T. Hsiao, H. D. Lin, K. B. Lee, and C. W. Jen, "High-speed memory saving architecture for the embedded block coding in JPEG2000," in *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 5, 2002, pp. 133–136.

[90] J. S. Chiang, Y. S. Lin, and C. Y. Hsieh, "Efficient pass-parallel architecture for EBCOT in JPEG2000," in *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 1, 2002, pp. 773–776.

[91] H. Yamauchi, S. Okada, K. Taketa, T. Ohyama, Y. Matsuda, T. Mori, T. Watanabe, Y. Matsuo, Y. Yamada, T. Ichikawa, and Y. Matsushita, "Image processor capable of block-noise-free JPEG2000 compression with 30 frames/s for digital camera applications," in *Dig. Tech. Papers IEEE Int. Solid-State Circuits Conf.*, 2003, pp. 46–47.

[92] H. C. Fang, T. C. Wang, Y. W. Chang, Y. Y. Shih, and L. G. Chen, "Novel word-level algorithm of embedded block coding in JPEG 2000," in *Proc. IEEE Int. Conf. Multimedia and Expo*, vol. 1, 2003, pp. 137–140.

[93] H. C. Fang, T. C. Wang, C. J. Lian, T. H. Chang, and L. G. Chen, "High speed memory efficient EBCOT architecture for JPEG2000," in *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 2, 2003, pp. 736–739.

[94] DSPworx Cheetah [Online]. Available: http://www.dspworx.com/cheetah.htm

[95] K. Andra, C. Chakrabarti, and T. Acharya, "A high-performance JPEG2000 architecture," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 3, pp. 209–218, Mar. 2003.

[96] Analog Devices Inc. ADV202 [Online]. Available: http://www.analog.com/

[97] H. C. Fang, C. T. Huang, Y. W. Chang, T. C. Wang, P. C. Tseng, C. J. Lian, and L. G. Chen, "81 MS/s JPEG 2000 single-chip encoder with rate-distortion optimization," in *Dig. Tech. Papers IEEE Int. Solid-State Circuits Conf.*, 2004, p. 328.

[98] ALMA Technologies JPEG2KE [Online]. Available: http://www.alma-tech.com/

[99] AMPHION CS6590 [Online]. Available: http://www.amphion.com/cs6590.html

[100] T. Wiegand, G. J. Sullivan, G. Bjntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[101] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G. J. Sullivan, "Rate-constrained coder control and comparison of video coding standards," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 688–703, Jul. 2003.

[102] K. Denolf, C. Blanch, G. Lafruit, and A. Bormans, "Initial memory complexity analysis of the AVC codec," in *Proc. IEEE Workshop Signal Processing Systems*, 2002, pp. 222–227.

[103] V. Lappalainen, A. Hallapuro, and T. D. Hamalainen, "Performance of H.26L video encoder on general-purpose processor," *J. VLSI Signal Process.*, vol. 34, no. 3, pp. 239–249, Jul. 2003.

Po-Chih Tseng was born in Tao-Yuan, Taiwan, R.O.C., in 1977. He received the B.S. degree in electrical and control engineering from National Chiao Tung University, Hsinchu, Taiwan, R.O.C., in 1999 and the M.S. degree in electrical engineering from National Taiwan University, Taipei, in 2001. He is currently working toward the Ph.D. degree at the Graduate Institute of Electronics Engineering, National Taiwan University.

His research interests include energy-efficient reconfigurable computing for multimedia systems and power-aware image and video coding systems.

Yung-Chi Chang was born in Kaohsiung, Taiwan, R.O.C., in 1975. He received the B.S. and M.S. degrees in electrical engineering from National Taiwan University, Taipei, in 1998 and 2000, respectively. He is currently working toward the Ph.D. degree at the Graduate Institute of Electrical Engineering, National Taiwan University.

His research interests include video coding algorithms and very large scale integration (VLSI) architectures for image/video processing.

Yu-Wen Huang was born in Kaohsiung, Taiwan, R.O.C., in 1978. He received the B.S. degree in electrical engineering from National Taiwan University, Taipei, in 2000. He is currently working toward the Ph.D. degree at National Taiwan University.

His research interests include video segmentation, moving object detection and tracking, intelligent video coding, motion estimation, face detection and recognition, and associated very large scale integration (VLSI) architectures.

Hung-Chi Fang was born in I-Lan, Taiwan, R.O.C., in 1979. He received the B.S. degree in electrical engineering from National Taiwan University, Taipei, in 2001. He is currently working toward the Ph.D. degree at National Taiwan University.

His research interests are very large scale integration (VLSI) design and implementation for signal processing systems, image processing systems, and video compression systems.

Chao-Tsung Huang was born in Kaohsiung, Taiwan, R.O.C., in 1979. He received the B.S. degree from the Department of Electrical Engineering, National Taiwan University, Taipei, in 2001. He is currently working toward the Ph.D. degree at the Graduate Institute of Electronics Engineering, National Taiwan University.

His major research interests include very large scale integration (VLSI) design and implementation for the one-, two-, and three-dimensional discrete wavelet transform.

Liang-Gee Chen (Fellow, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical engineering from National Cheng Kung University, Tainan, Taiwan, R.O.C., in 1979, 1981, and 1986, respectively.

In 1988, he joined the Department of Electrical Engineering, National Taiwan University, Taipei. During 1993–1994, he was a Visiting Consultant in the DSP Research Department, AT&T Bell Labs., Murray Hill, NJ. In 1997, he was a Visiting Scholar of the Department of Electrical Engineering, University of Washington, Seattle. Currently, he is a Professor at National Taiwan University. His current research interests are digital signal processing (DSP) architecture design, video processor design, and video coding systems.