A Low-Power Dual-Mode Video Decoder for Mobile Applications

Tsu-Ming Liu, Ting-An Lin, Sheng-Zen Wang, and Chen-Yi Lee, National Chiao-Tung University

ABSTRACT

The objective of this article is to highlight design challenges for low-power and dual-video standard requirements, especially in mobile applications. Due to the advent of the newly announced H.264, a generic problem of standard incompatibility has appeared between H.264 and prevalent MPEG-x video standards, which must be resolved on both algorithmic and architectural levels. Furthermore, several low-power techniques targeted at achieving lower memory requirements and processing cycles are also described and discussed.

INTRODUCTION

Current video compression standards including MPEG-1/2/4, H.261/2/3/4 [1], AVS [2], and VC-1 (an acronym for Video Codec 1 and the name of the standardized version of WMV-9) [3] have played an important role in the world of mobile communication systems where bandwidth is still a valuable commodity. Hence, video compression techniques are of prime concern for reducing the amount of information without sacrificing much of its visual quality. Specifically, these techniques are based on a hybrid coding infrastructure. They perform a block-based discrete cosine transform to take advantage of the spatial correlation property and exploit the motion-compensated prediction to improve the coding efficiency. In general, transform coding is based on dividing a frame into small blocks, taking the transform of each block, discarding high-frequency coefficients, and quantizing low-frequency coefficients. Afterward, quantized coefficients are coded using variable-length coding techniques. These coded streams can be stored as a digital content for video playback or sent into a wireless/broadcast channel environment for portable multimedia services.

Although a wide range of video coding standards have been developed, these algorithms are quite diverse, thus leading to a lack of compatibility. For example, prevalent MPEG standards are backward compatible; however, the advent of H.264 and VC-1 cannot be backward compatible to the former H.26x and MPEG-x family of video coding standards. Therefore, a development of the combined video coding standard is indispensable for meeting different standard requirements. Moreover, from an application point of view, the digital video broadcasting (DVB) project has paved the way for the introduction of MPEG-2-based digital TV service, known as DVB-T in many countries. Recently, DVB-H, a spin-off of the DVB-T standard, has adopted the transmission of H.264 for handheld digital TV due to its bandwidth efficiency. DVB-H is totally backward compatible to DVB-T but is transmitted with different video contents (i.e., MPEG-2 versus H.264). In other words, a generic problem of standard incompatibility has emerged, resulting in the design challenge for the multistandard integration. In this article we choose the well-known MPEG-2 and the newly announced H.264 for our decoding platform in order to support dual-mode video standards.

One of the primary purposes in the design of mobile communication systems is power reduction. Although many compression techniques greatly reduce the transmission bandwidth, the introduced coding/decoding complexity and computing power become the key design challenges for real-time and battery-operated systems. We consider a newly standardized H.264 video standard [1] as an extreme case. There are two major factors to adversely impact the power dissipation. The first is the large memory storage including internal registers, SRAM, and external SDRAM. Specifically, H.264 utilizes the neighboring pixels to create a reliable predictor, leading to a dependency on a long past history of data. This problem can be solved by allocating large memory storage but introducing large memory power dissipation. The second is the computationally intensive processing unit, for example, motion compensation and deblocking filter. Motion compensation requires a great deal of memory access, which degrades decoding throughput. The deblocking filter involves a pixelwise process and adaptively performs interpolation processes. In summary, they require high-speed or high-throughput hardware solutions to improve the coding/decoding performance.

The organization of this article is as follows. First, we present a cost-efficient design integrating both MPEG-2 and H.264 video standards. Next, several low-power techniques are described and discussed in this integrated circuit. Implementation results are also summarized. Finally, a brief conclusion is made.

This work was supported by the National Science Council of Taiwan, R.O.C. under grant no. NSC94-2215-E-009-046, and by the NCTU-MTK Research Program.

0163-6804/06/$20.00 © 2006 IEEE

In the video decoder, H.264 aims at providing functionality similar to prevalent MPEG-2, but with significantly better coding performance. The improved performance comes from some new techniques. To integrate these new techniques of H.264 into MPEG-2, a combined data flow is designed.

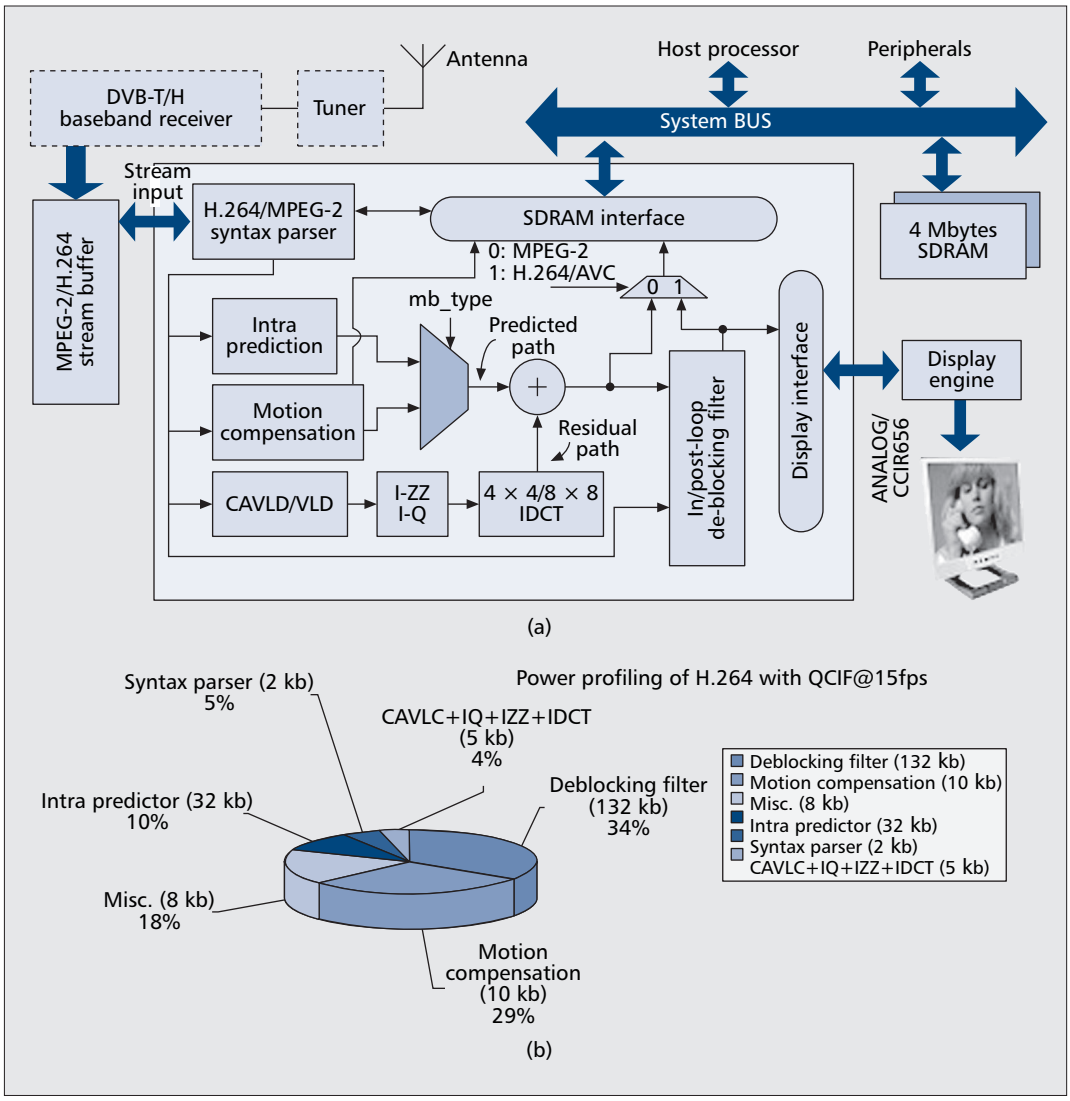

Figure 1. *a) Block diagram; b) power profiling for video decoding systems [8].*

System Overviews of MPEG-2 and H.264

This section provides system overviews of general mobile communication systems, especially with regard to the video decoding side. For a video broadcasting environment, one specific example is the COFDM-based transmission of MPEG-2/H.264 [4] in DVB-T/H [5] communication systems. This article emphasizes the dual-mode video decoder due to the design challenge of backward incompatibility between MPEG-2 and H.264.

Basic Configurations

Figure 1a shows the system block diagram, including the DVB-T/H baseband receiver and MPEG-2/H.264 video decoder. The received signal is first processed by the RF front-end. Afterward, the analog signal is fed into the baseband receiver to create the video bit-stream. The video decoder retrieves the stream in order to parse the header information and decode into the pixel data for subsequent display devices.

In the video decoder, H.264 aims at providing functionality similar to prevalent MPEG-2, but with significantly better coding performance. The improved performance comes from some new techniques such as spatial prediction in intra coding, adaptive block-size motion compensation, 4×4 integer transformation, context-adaptive entropy decoding, and adaptive deblocking filtering. To integrate these new techniques of H.264 into MPEG-2, a combined data flow is designed: it is composed of the residual path and predicted path shown in Fig. 1a. In the residual path, a context-adaptive variable length decoder (CAVLD) or VLD translates the received streams into symbols through a table-lookup method. The followup processes first reorder the symbols into a 2D block using inverse zig-zag (I-ZZ) scan, rescale (inverse quantization, or I-Q) the frequency-domain coefficients of a block, and then perform inverse discrete cosine transform (IDCT) to produce residual pixels. On the other hand, the macroblock type (`mb_type`) can be decoded by an MPEG-2/H.264 syntax parser and is defined to select the source of predicted pixels. These pixels come from either spatially predicted (intra prediction) or temporally predicted (motion

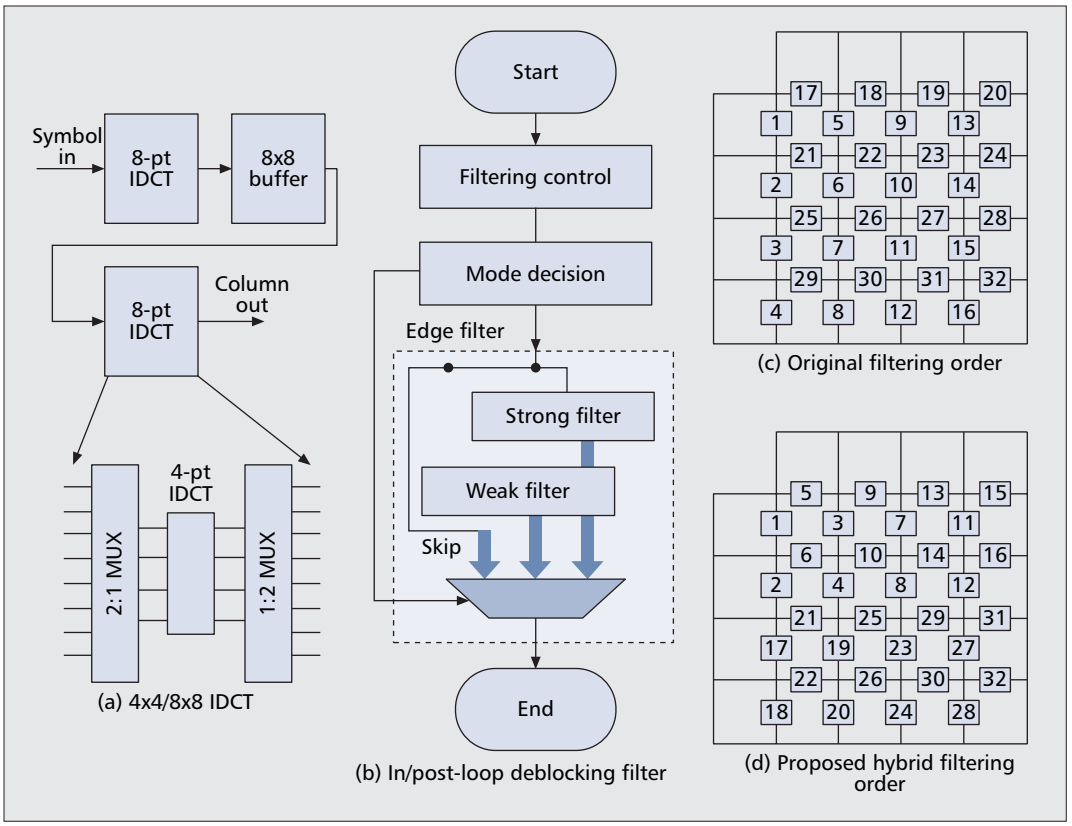

Figure 2. *Block diagram of a) 4×4/8×8 IDCT; b–d) in/post-loop deblocking filter.*

The deblocking filter is standardized by H.264 and brought within the motion-compensated prediction loop (i.e., in-loop filter), whereas the deblocking filter in the MPEG-x family is user-defined without standard-limitation and performs outside the coding loop (i.e., a post-loop filter).

compensation) blocks. The addition of predicted and residual blocks will be performed in a pixel-wise manner. Afterward, the filtered results are sent into external memory through synchronous DRAM (SDRAM) interface (I/F) for decoding subsequent frames. Moreover, a separate data path is utilized for onscreen display (OSD) through the display I/F. The results of the display engine are sent into the display monitor, in either digital (CC1R656) or analog form.

Dual-Mode Video Decoder

This section is an overview of dual-mode techniques from a cost perspective. A dual-mode video decoder integrating MPEG-2 and H.264 under the low-bit-rate mobile communication systems is introduced. At first, we tabulate the similarity of the key modules given in Table 1. From an algorithmic point of view, motion compensation and intra prediction in MPEG-2 are just a subset of that in H.264. That is, they can be totally merged into H.264 by sharing the hardware resources. With regard to the entropy decoder, H.264 features adaptive tables, but it is still similar to variable length codes defined in MPEG-2. So far, H.264 can be partially compatible to the MPEG-2 video coding standard. Nevertheless, the IDCT algorithms between MPEG-2 and H.264 are so diverse that they are difficult to combine. The integration of deblocking filters also has the same problem.

A major focus while combining MPEG-2 and H.264 is IDCT. As listed in Table 1, the IDCT kernel of H.264 is a 4×4 integer transform kernel, but that of MPEG-2 is an 8×8 cosine transform kernel. Due to an algorithmic difference in terms of transform size and kernel characteristics, a shared IDCT structure presents a great challenge. In our design, we exploit two 8-point IDCTs for row and column transforms, respectively, and an 8×8 buffer for matrix transposition (Fig. 2a). The 8-point IDCT can be computed using 4-point IDCT recursively. In other words, N-point IDCT can be decomposed into an $N/2$-point IDCT by partitioning even and odd coefficients. This allows us to generate 8-point IDCT from lower-order 4-point IDCTs [6]. Therefore, this 4-point IDCT can be simultaneously shared both in MPEG-2 and H.264, and the problem of different transform size can be resolved. As for kernel characteristics, both standards require addition and the input bit-width of adders in H.264 is smaller than those in MPEG-2. Thus, the common terms of addition can be shared between MPEG-2 and H.264.

The deblocking filter is standardized by H.264 and brought within the motion-compensated prediction loop (i.e., in-loop filter), whereas the deblocking filter in the MPEG-x family is user-defined without standard-limitation and performs outside the coding loop (i.e., a post-loop filter). Although both filters intend to remove block discontinuities and improve visual quality, there is a main drawback when we replace an in-loop filter with a post-loop one. The experimental results reveal that the performance improvement is very small (0.04 dB) if we put the in-loop filter of H.264 into MPEG-2 video decoding flows. To alleviate this problem, we derive an integration-oriented algorithm that can be reconfigured as either an in-loop or a post-loop filter without sacrificing considerable per-

There are two issues to be noted. First, reducing the memory size is key to achieving low-power consumption. Second, most of computations require memory accesses in order to accomplish a data exchange between logic and memory.

Key module		MPEG-2	H.264	Similarity
Entropy decoder		VLD	Context-adaptive VLD	Variable length table
Inverse DCT		8 × 8 cosine kernel	4 × 4 integer kernel	
Motion compensation	Luma	Bilinear	Half: 6-tap FIR, Quarter: 6-tap FIR/bilinear	Bilinear interpolation
	Chroma	Bilinear	Bilinear	
Intra prediction		DC prediction	Directional spatial prediction	Spatial prediction
Deblocking filter		N/A (user-defined)	In-loop adaptive filter	

Table 1. *The similarity analysis of each key module.*

formance [7]. Figure 2b depicts the block diagram, composed of filtering control, mode decision, and edge filter. First, the filtering control decides the filtering order where the vertical edges will be filtered first, followed likewise by the horizontal edges. Second, the mode decision partitions filtering strengths into strong, weak, and skip modes. Third, an edge filter follows the filtering mode to adaptively perform interpolation processes. Altogether, this algorithm is totally compatible with the standardized in-loop deblocking filter but improves visual quality in the post-loop operation. In addition, it shares edge filters between MPEG-2 and H.264 and integrates other distinct blocks into a simple one.

Low-Power Techniques for Mobile Applications

Figure 1b shows the power profiling of H.264 video decoding for mobile applications [8]. Only power profiling in H.264 is shown here because its power requirements are much higher than those in MPEG-2. In this figure, the numbers in brackets represent the memory size used in that module. Motion compensation and the deblocking filter occupy large portions of this pie chart. This is because the motion compensation is the most computationally intensive process and the deblocking filter uses large internal memory to remove long data dependencies. Therefore, there are two issues to be noted. First, reducing the memory size is key to achieving low-power consumption. Second, most of computations require memory accesses in order to accomplish a data exchange between logic and memory. It is obvious that reducing the working frequency cuts data switching activity, resulting in less dynamic power consumption. In the following, we introduce a collection of low-power techniques, including the improved memory system and several low-power building blocks.

Memory System

To improve the memory system of the video decoder, we introduce two techniques. First, we propose a decoding ordering method to reduce the processing cycles on the intra prediction and inter prediction (i.e., motion compensation) modules. Second, we use a new prediction method to make a better compromise between internal and external memory power consumption.

According to the decoding flow of the specification in H.264, a 4 × 4 subblock is the smallest processing unit. To efficiently transfer 4 × 4 pixel data among modules, we discuss two strategies to organize the word lengths for data transactions. One is a 4 × 1 row-by-row decoding ordering, and the other is a 1 × 4 column-by-column decoding ordering. Although the 1 × 4 is similar to the 4 × 1 within a 4 × 4 subblock, the decoding orders in a 16 × 16 macroblock (MB) are widely different. Figures 3a and 3b show the 4 × 1 and 1 × 4 decoding orderings in one MB, respectively. Compared to the 4 × 1 row-by-row decoding ordering, the 1 × 4 column-by-column decoding ordering provides a better data structure, reducing the processing cycles on intra prediction and interprediction modules. For example, an intra prediction module requires left neighboring pixels to spatially create the predicted pixels. These neighboring pixels can be easily fetched through the 1 × 4 ordering since they are organized in a column-wise manner. As for the intra prediction module, extra initialization cycles are required when the thread of Figs. 3a and 3b makes a turn. Therefore, the access times are related to the frequency of turning events. The results prove that the 1 × 4 column-by-column decoding ordering reuses the neighboring pixels and reduces the turning events, yielding cycle reductions of 17 percent and 28 percent in the intra prediction and interprediction modules, respectively [9].

Using a memory hierarchy is advantageous to a large memory system. To reduce memory power consumption, we aim to have a memory hierarchy where copies of data from larger memories that exhibit high spatial locality are cached in smaller memories. Figure 3c shows a three-level memory hierarchy for an H.264 video decoding system [4]. The first level of memory hierarchy includes several pipelined registers and data buffers. The second and third levels of memory hierarchy are slice pixel SRAM and SDRAM. Because intra prediction and deblocking filter modules require neighboring pixels to construct the predicted and filtered pixels, the slice pixel SRAM, storing upper rows of pixels,

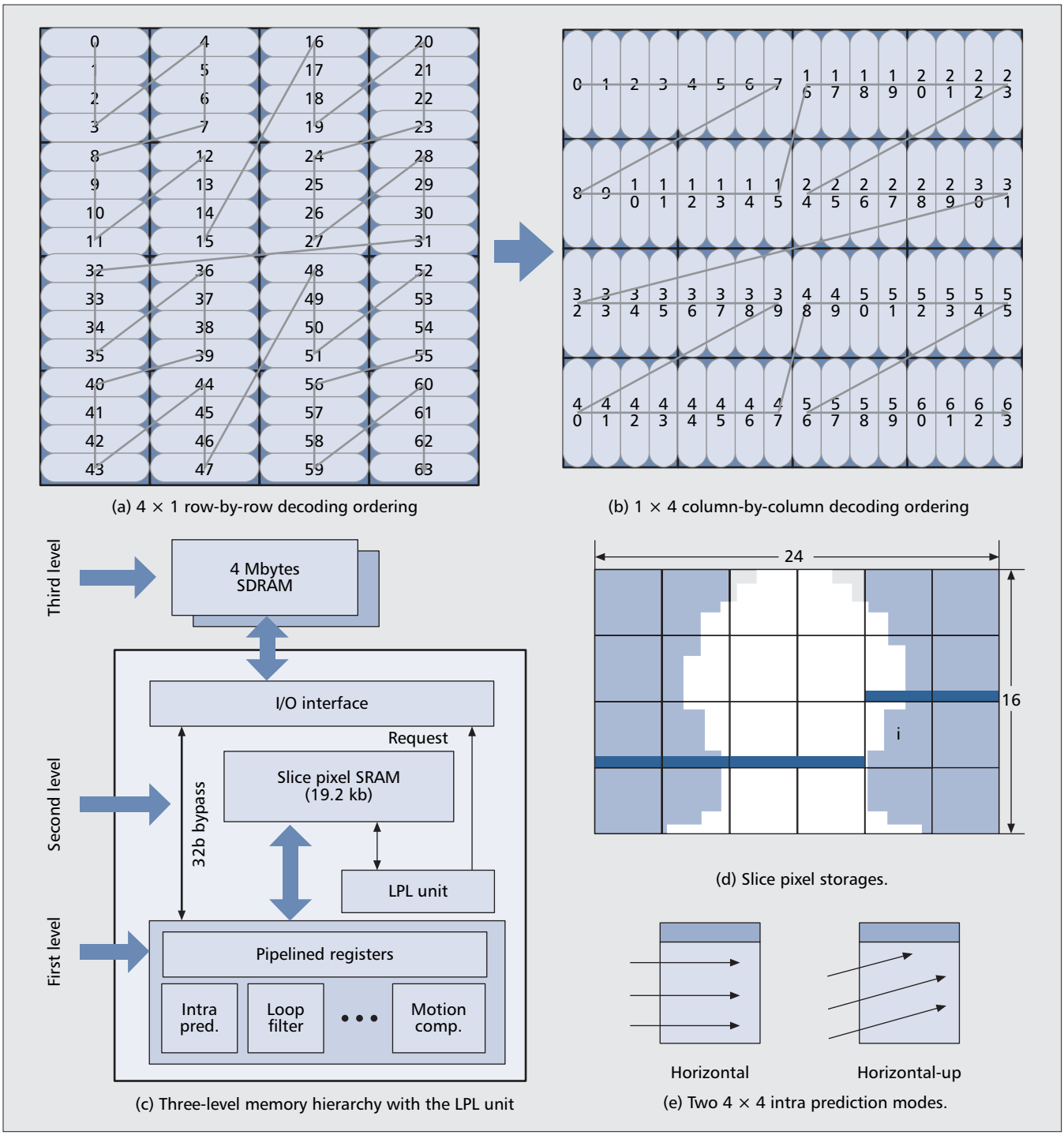

Figure 3. *A memory system in terms of a), b) data organization; c)-e) memory hierarchy.*

is exploited to remove the data dependency. For instance, considering a general decoding flow of a 24 × 16 frame as in Fig. 3d, the black-region indicates the required pixel when the decoding index moves to *i*, and these pixels will be stored in slice pixel SRAM. However, storing all pixels in rows of vertical pixels is unnecessary when the following decoding process is unrelated to the upper neighboring pixels. Hence, we propose a line-pixel-lookahead (LPL) unit [12] to eliminate the unused pixels. In particular, we find that not all upper neighboring pixels need to be prestored when they are determined as a "horizontal prediction mode" in intra prediction or a "SKIP mode" in a deblocking filter. To illustrate how this works, we use intra prediction as an example. In H.264, there are nine prediction modes used to spatially create the predictor as follows: vertical, horizontal, DC, diagonal down-left, diagonal down-right, vertical-right, horizontal-down, vertical-left, and horizontal-up modes. In Fig. 3e, because both the horizontal and horizontal-up modes are only related to the left neighboring pixels, there is no need to keep the upper pixels, leading to a reduction of memory space and access times. Finally, the implementa-

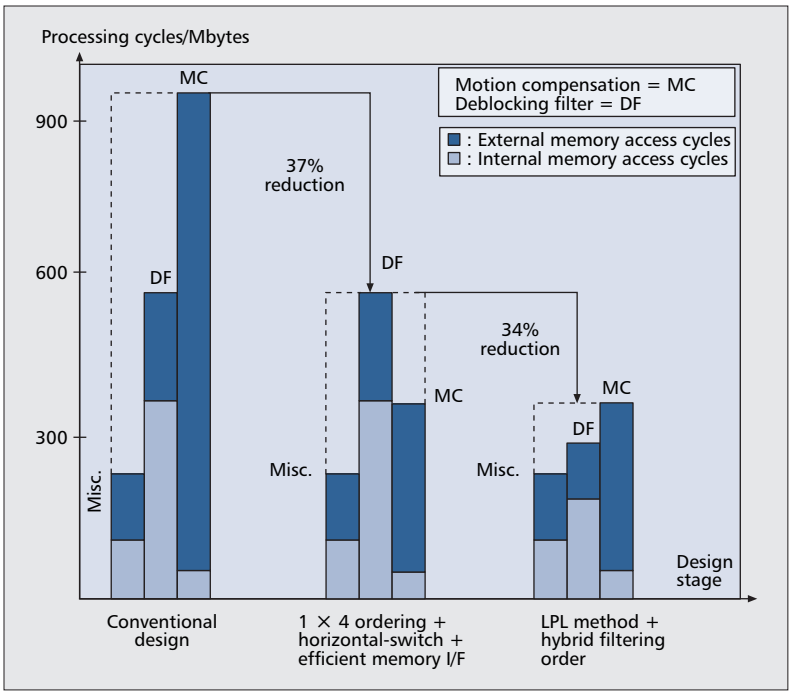

Figure 4. *Processing cycle breakdown in each architectural design phase.*

tion results show that 50 percent of memory power consumption can be saved as compared to the conventional design without exploiting the memory hierarchy [12].

Under an identical design specification, reducing the processing cycles yields operations with lower system clock rate and voltage and thus lower power consumption. Figure 4 shows a processing cycle breakdown in different architectural design stages. The white and gray regions denote the internal- and external-memory access cycles, respectively. Moreover, each stage can be partitioned into three components: motion compensation (MC), deblocking filter (DF), and miscellany (misc.). According to the left bar in Fig. 4, MC and DF are the most computationally intensive operations in H.264. In the following, we propose several techniques to reduce the processing cycles in these modules.

Motion Compensation

It is obvious that reducing the processing cycles in the MC module can improve the system performance, since this presents a system bottleneck as compared to other modules. The interpolation unit is always the most time-consuming module. In [10], we propose a horizontal-switch approach to reuse the neighboring pixels for the interpolator design. To reduce the external-memory access times, the MC module has to increase the reuse probability for overlapped regions of neighboring interpolation windows. This can be achieved by applying a data buffer attached to each shift register for luma interpolators. Furthermore, a chroma interpolator behaves similarly to a luma interpolator, and both of them can be simplified into a multiplication-free design approach. The simulation results exhibit that the proposed horizontal-switch approach can save 30 percent of memory accesses as compared to conventional approaches.

Although we increase the reuse probability in order to reduce the access frequency to the external memory, the external-memory interface has to cooperate with the MC so as to improve the overall access efficiency. In this design, two external frame memories are allowed for writing decoded data and reading reference data reciprocally at the same time. Compared with SRAM, SDRAM is adopted due to the cost and power issues. However, SDRAM also induces the longer access latency and degrades the decoding throughput because of the internal pipeline architectures and 3D structure of the bank, row, and column characteristics. In [8], to solve the above problems, an efficient memory interface is introduced to overlap and reschedule each access command in order to improve the bandwidth utilization. This interface is composed of the bank controller, memory scheduler, and several read/write buffers. Each bank controller generates suitable commands for read/write processes. The memory scheduler collects these commands and then sends rescheduled commands to external SDRAM. Read and write data buffers store burst data, and read/write command queues are designed to hold successive commands.

Deblocking Filter

In general, deblocking filter contributes about one-third of the computational complexity at the H.264 decoder. It operates each filtering process on the 4×4 boundaries instead of the 8x8 boundaries in filters of H.263 or MPEG-4 video standards. Therefore, a large number of memory accesses are its penalty for the low-power mobile communication applications. Figure 2c illustrates the filtering order on 4×4 pixel boundaries where the vertical edges are filtered first, followed by the horizontal edges. To accomplish the filtering processes, a direct approach reloads the redundant pixels when the filtering edges are switched from vertical to horizontal directions. These redundant accesses will increase processing cycles in the DF. To alleviate this problem, we propose a hybrid filtering order [11] to reschedule the processing orders without sacrificing the system performance. In Fig. 2d, we deduce a hybrid order so as to reuse the filtered pixels and thereby reduce the memory access times. The main idea is that we use four pixel buffers to keep the intermediate pixel value and perform the vertical and horizontal filtering process successively. Therefore, the proposed method not only prevents the data reaccess for different directions, but also improves the access efficiency without degrading the filtering performance. Compared to existing solutions, the proposed design can save one-half of processing cycles per macroblock on average.

Let us make a brief summary in terms of processing cycles in Fig. 4. First, we propose several techniques to reduce the processing cycles in the MC as follows: the 1×4 decoding ordering, the horizontal-switch, and the efficient memory-interface methods. Therefore, the processing cycles are reduced by 37 percent as compared to conventional designs. Next, the processing cycles on the MC module decrease, whereas the deblocking filter becomes the cycle bottleneck from the system point of view. To further reduce the process-

ing cycles, the hybrid filtering order and LPL prediction methods are proposed to improve the memory access efficiency and a 34 percent of cycle reduction can be further obtained as compared to the previous design stage.

Implementation Results

Low power dissipation is always a critical issue in the design of mobile or handheld devices. To obtain the real metric of power, a chip using the aforementioned techniques has been fabricated in a 0.18 μm single-poly six-metal CMOS process with an area of 3.9 × 3.9 and 300,000 logic gates (excluding memory). Figure 5 shows a chip micrograph that combines MPEG-2 with H.264 video standards. Particularly, the 4 × 4/8 × 8 IDCT and in/post-loop deblocking filter are designed to reduce the silicon area. The slice pixel SRAM is used to store neighboring pixels, thus reducing the extensive accesses of external memory as well as I/O power dissipation. The LPL unit interfaced to the slice pixel SRAM is exploited to further improve the access efficiency. On the other hand, this chip can be further integrated into mobile communication systems (see the general block diagram in Fig. 1a). The maximum working frequency of this chip is 100 MHz and achieves 101 Mpixels/s of maximum throughput rates that meet the decoding requirements of high-resolution video sequences (1080 HD, 1920 × 1088 pixels/frame at 30 frames/s as well as 4:2:0 chrominance formats). The associated core power dissipation is 89 and 102 mW in MPEG-2 and H.264 video standards, respectively. Because low-resolution video formats are also supported through changing the working frequency, the required frequencies of standard definition (SD), common intermediate format (CIF), and quarter CIF (QCIF) are 16.6, 4.6, and 1.15 MHz, respectively. This fairly low operating frequency is an indication of the improved memory hierarchy and processing cycle reduction. In a mobile communication environment, a set of well-known QCIFs, which correspond to a spatial resolution of 176 pixels × 144 lines, are used. The power dissipation on MPEG-2 and H.264 is only submilliwatt and requires 108 and 125 μW at 1 V supply voltage, respectively. Hence, the proposed design offers a low-power VLSI solution and is applicable to mobile communication systems.

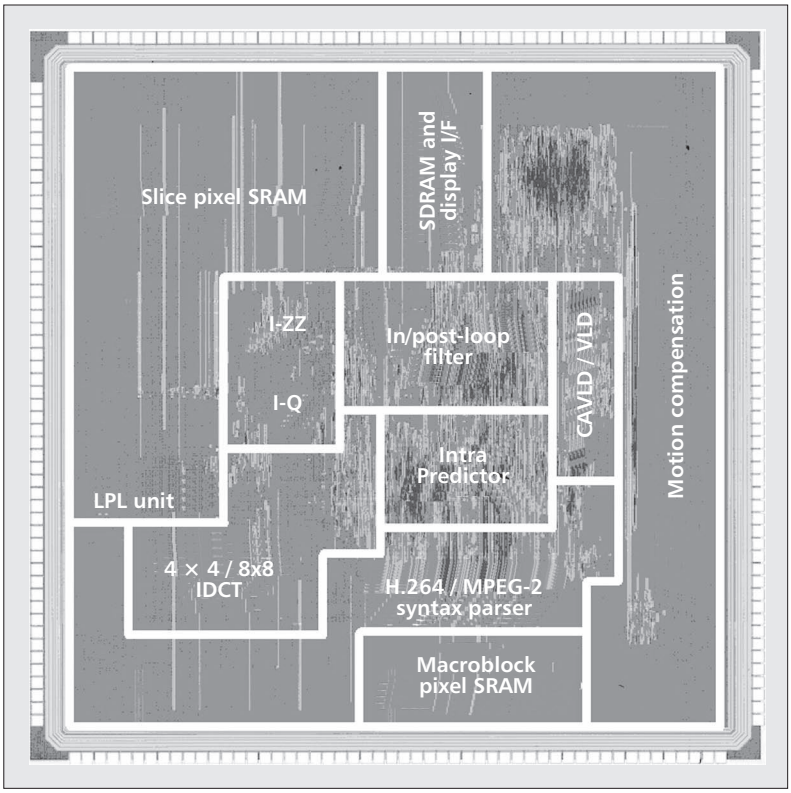

Figure 5. *Chip micrograph.*

Conclusion

In this article, we have introduced a dual-mode video decoder that efficiently combines MPEG-2 with the H.264 video coding standard. Specifically, we derived an algorithm in IDCT and a deblocking filter in order to achieve integration-efficiency. Moreover, we introduce several low power techniques to prolong battery life. First, we chose the 1 × 4 column-by-column decoding ordering to reduce the memory access times. In addition, we introduced an LPL prediction unit to reduce the memory size and access frequency. Second, we reduced the processing cycles in both MC and DF modules, since they are the most critical factors from a system point of view. Finally, we found that the measured result exhibits that MPEG-2 and H.264 video decoding of QCIF sequences at 15 frames per second can be accomplished at a clock frequency of 1.15 MHz and requires 108 and 125 μW, respectively. This low-power and integration-efficient design also reveals its strong suitability for mobile communication systems where low power requirements are essential.

Acknowledgment

The authors would like to thank Kang-Cheng Hou, Jiun-Yan Yang, Wen-Ping Lee, Wei-Chin Lee, and colleagues of the Si2 Group of National Chiao Tung University for insightful discussions about this work. They also thank the Chip Implementation Center (CIC) for testing services.

References

[1] ITU-T Rec. H.2641 ISO/IEC 14496-10 AVC, "Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification," May 2003.

[2] Audio Video Coding Standard Workgroup of China (AVS), http://www.avs.org.cn/enl

[3] S. Srinivasan *et al.*, "Windows Media Video 9: Overview and Applications," *Sig. Processing: Image Commun.*, vol. 19, 2004, pp. 851–75.

[4] T.-M. Liu *et al.*, "A 125 μW, Fully Scalable MPEG-2 and H.264/AVC Video Decoder for Mobile Applications," ISSCC Dig. Tech. Papers, Feb. 2006, pp. 402–3.

[5] L.-F. Chen *et al.*, "A 1.8 V 250 mW COFDM Baseband Receiver for DVB-T/H Applications," *ISSCC Dig. Tech. Papers*, Feb. 2006, pp. 262–63.

[6] Hsieh S. Hou, "A Fast Recursive Algorithm for Computing the Discrete Cosine Transform," *IEEE Trans. Acoustics, Speech, and Sig. Processing*, vol. 35, no. 10, Oct. 1987, pp. 1455–61.

[7] T.-M. Liu, W.-P. Lee, and C.-Y. Lee, "An Area-Efficient and High-Throughput De-blocking Filter for Multi-Standard Video Applications," *IEEE Int'l. Conf. Image Processing*, Sept. 2005, pp. III-1044–47.

[8] T.-M. Liu *et al.*, "An 865-μW H.264/AVC Video Decoder for Mobile Applications," *IEEE Asian Solid-State Circuit Conf.*, Nov. 2005, pp. 301–4.
[9] T.-A. Lin *et al.*, "An H.264/AVC Decoder with 4 × 4-Block Level Pipeline," *IEEE Int'l. Symp. Circuit and Sys.*, May 2005, pp. 1810–13.
[10] S.-Z. Wang *et al.*, "A New Motion Compensation Design for H.264/AVC Decoder," *IEEE Int'l. Symp. Circuit and Sys.*, May 2005, pp. 4558–61.
[11] T.-M. Liu *et al.*, "A Memory-Efficient Deblocking Filter for H.264/AVC Video Coding," *IEEE Int'l. Symp. Circuit and Sys.*, May 2005, pp. 2140–43.
[12] T.-M. Liu and C.-Y. Lee, "Memory-Hierarchy-Based Power Reduction for H.264/AVC Video Decoder," *IEEE Int'l. Symp. VLSI Design, Automation, and Test*, Apr. 2006, pp. 247–50.

BIOGRAPHIES

TSU-MING LIU (mingle@si2lab.org) was born in I-Lan, Taiwan, R.O.C., in 1980. He received B.S. and M.S. degrees in electronics engineering from National Chiao-Tung University, Taiwan, in 2002 and 2004, respectively. During 2004.7-10, he was an intern with Sunplus Technology Co., Hsinchu, Taiwan. In 2004, he joined the Institute of Electronics Engineering of National Chiao-Tung University, where he is currently a Ph.D. candidate. His major research interests include binary shape coding, joint source and channel design, H.264/AVC video decoding, and associated VLSI architectures.

TING-AN LIN received B.S. and M.S. degrees in electronics engineering from National Chiao-Tung University, Taiwan, in 2003 and 2005, respectively. In 2006, he joined MediaTek, Inc., Hsinchu, Taiwan, where he develops video decoding chip design. His major research interests include channel equalizers, H.264/AVC intrapredictor design, and associated VLSI architecture.

SHENG-ZEN WANG received B.S. and M.S. degrees in electronics engineering from National Chiao-Tung University, Taiwan, in 2003 and 2005, respectively. In 2006, he joined MediaTek, Inc., Hsinchu, Taiwan, where he develops digital TV backend-related systems. His major research interests include memory controller design, H.264/AVC video decoding, and associated VLSI architecture.

CHEN-YI LEE [M'01] received a B.S. degree from National Chiao Tung University, Hsinchu, Taiwan, R.O.C., in 1982, and M.S. and Ph.D. degrees from Katholieke University Leuven (KUL), Leuven, Belgium, in 1986 and 1990, respectively, all in electrical engineering. From 1986 to 1990, he was with IMEC/VSDM, working in the area of architecture synthesis for digital signal processing (DSP). In February 1991 he joined the faculty of the Electronics Engineering Department, National Chiao Tung University, where he is currently a professor and Department Chair. His research interests mainly include VLSI algorithms and architectures for high-throughput DSP applications. He is also active in various aspects of high-speed networking, system-on-chip design technology, very-low-power designs, and multimedia signal processing. He served as the director of the Chip Implementation Center (CIC), an organization for IC design promotion in Taiwan. He is now Microelectronics Program Coordinator of the Engineering Division under National Science Council of Taiwan. He was the former IEEE Circuits and Systems Society Taipei Chapter Chair.