Hardware Architecture Design of Video Compression for Multimedia Communication Systems

Shao-Yi Chien, Yu-Wen Huang, Ching-Yeh Chen, Homer H. Chen, and Liang-Gee Chen
National Taiwan University

ABSTRACT

VLSI realization of video compression is the key to many real-time multimedia communications systems. Among the video compression algorithms, the newly established MPEG-4 and, in particular, H.264 standards have become increasingly popular. However, the high coding efficiency of such video coding algorithms comes at the cost of a dramatic increase in complexity. Effective and efficient hardware solutions to this problem are necessary. In this article we present an overview of the hardware design issues of MPEG-4 and H.264. Both module and system architectures of these two coding standards are discussed. Based on these architectures, the design of a single-chip encoder complying with the H.264 baseline profile and capable of encoding the D1 resolution (720 × 480) video at 30 Hz is presented as an example. In addition, the system integration issues of video compression engines with multimedia communication systems and a general hardware platform for various applications are discussed.

INTRODUCTION

We are now in the era of multimedia communications. In this new era, we not only communicate with others by voice, but also through video, audio, graphics, text, and other data over networks. There are several driving technologies to make multimedia communications feasible. The first is broadband and network technology. With this new technology, the bandwidth of transmission increases dramatically, and the quality of transmission is also enhanced for various types of data over networks. The second driving technology is the compression technique. The bit rate of multimedia data is quite large. Without compression, it is impossible to transmit the raw data directly. The third is very larg-scale integration (VLSI) technology. The technology development of semiconductor technology and processors both follow Moore's Law, in which chip capacity and processing power are doubled every 18 months. Advanced VLSI technology provides a powerful platform for realization of more complicated compression and communication techniques, and makes multimedia communication systems affordable for everyone.

Among various types of multimedia data, video data takes the largest bit rate; thus, video compression usually plays an important role in a multimedia communications system. There are two series of video compression standards: Instnational Standards Organization (ISO) MPEG-x standards and International Telecommunication Union — Telecommunication Standardization Sector (ITU-T) H.26x standards. Among these standards, the newly established MPEG-4 standard [1] and, in particular, H.264 standards [2], which can also be referred as Joint Video Team (JVT) or MPEG-4 Advanced Video Coding (AVC), have become increasingly popular. Both of these standards are designed for multiple purposes, such as video broadcasting, video streaming, video surveillance, and storage. However, from an industrial point of view, MPEG-4 is often used for video streaming, videoconference, and video recording in digital still cameras and feature phones; H.264, which can save more than 60 percent of the bit rate of MPEG-2, will possibly be used for digital video broadcasting (DVB) and high-definition DVD (HD-DVD).

The latter coding standards can achieve much higher coding efficiency. However, the complexity of the newly established standards become tremendous. Therefore, for real-time applications, VLSI implementation of these coding standards is necessary, and integrating moderate hardware accelerators for video compression in a system-on-chip (SoC) is usually the key to achieving the low power consumption, high performance, and low cost requirements of real-time multimedia communication products. In this article we briefly introduce the hardware architecture design issues of video compression for multimedia communications systems. In the next section the newly established video coding standards are described, including

0163-6804/05/$20.00 © 2005 IEEE

The H.264 project started in 1999 as H.26L, a long-term project in ITU-T. It was finalized as MPEG-4 Part 10 AVC, or H.264, in 2003. It is currently the most powerful coding standard and can save about 40 percent of the bitrate of MPEG-4 ASP for storage applications.

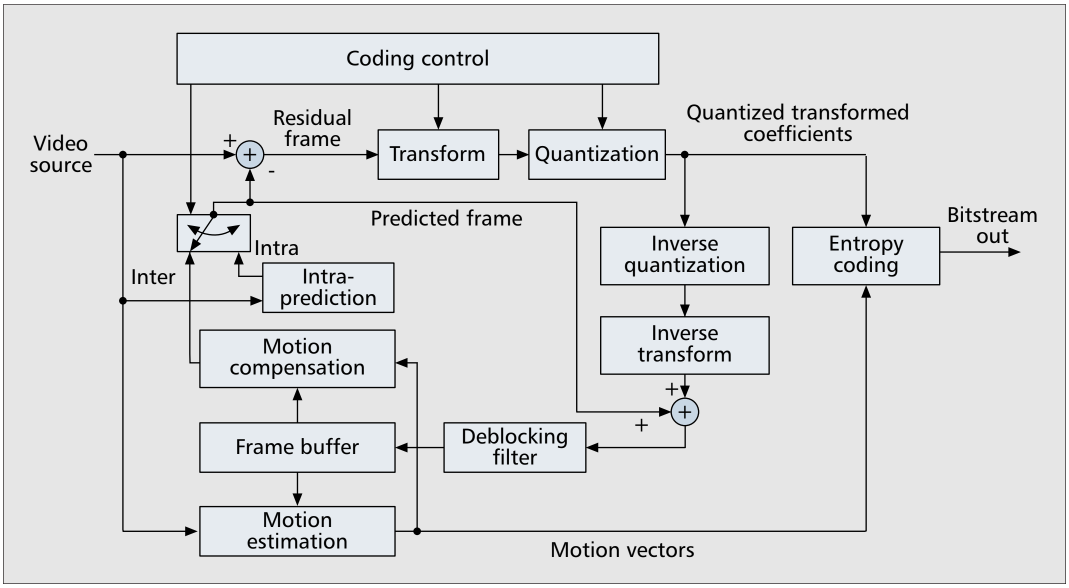

Figure 1. *Block diagram of a video encoder based on hybrid coding scheme.*

MPEG-4 and H.264. The associated hardware architectures of the key modules are then shown. Next, the system integration issues will be discussed. Finally, a short conclusion is given.

Newly Established Video Compression Standards: MPEG-4 and H.264

In this section the newly established video compression standards MPEG-4 and H.264 are introduced. The standardizing process of MPEG-4 started in 1993, and it became an International Standard in 1999. There are thee important goals of MPEG-4: interactivity, high compression efficiency, and universal access. For interactivity, MPEG-4 is the first coding standard supporting object-based coding, which allows users to interact with video contents object by object. MPEG-4 also integrates several new coding tools to reach better coding efficiency. For the goal of universal access, MPEG-4 considers scalability of video contents. Transmission of video over mobile networks and error-prone environments is considered as well, which makes MPEG-4 popular for network-related applications. Note that in MPEG-4 many profiles and levels are defined for conformance consideration. A profile is defined by a set of coding tools, and a level is defined for the capability of the encoder and decoder (codec). Among various profiles, simple profile (SP) and advanced simple profile (ASP) are paid more attention by the industry.

The H.264 project started in 1999 as H.26L, a long-term project in ITU-T. It was finalized as MPEG-4 Part 10 AVC, or H.264, in 2003. It is currently the most powerful coding standard and can save about 40 percent of the bit rate of MPEG-4 ASP for storage applications. Note that H.264 has four profiles: baseline, extended, main, and high profiles. The baseline profile is the simplest one, designed for video telephony and mobile video applications. The extended profile is designed for Internet video streaming. The main profile is designed for video broadcasting and entertainment applications. Finally, the high profile is designed for HDTV applications.

Before comparing between different video compression standards, we first introduce the general hybrid video coding scheme, which is adopted by most standards. The block diagram of the hybrid coding scheme is shown in Fig. 1. The frame buffer stores one or several reference frames that have been transmitted. Within these reference frames, motion estimation (ME) finds the corresponding position for each block in the current frame with a block matching algorithm and records the displacements as motion vectors. Motion compensation (MC) can then reconstruct the frame by motion vectors as the predicted frame in the temporal domain, which is called *interprediction*. On the other hand, *intraprediction* can make another prediction only considering the information in the same frame. The difference between the original frame and the inter- or intrapredicted frame is the residual frame data. Next, transform and quantization are used to further reduce the spatial redundancy. Transform can transform the data from the spatial domain to the frequency domain, where the data representation becomes more compact and easier to compress. The quantization then quantizes the coefficients in the frequency domain considering the human vision system. The quantized coefficients have better statistic distribution for compression. After that, entropy coding, which can remove statistical redundancy, can encode the quantized transformed coefficients and motion vectors as the final bitstream. On the other hand, the quantized transformed coefficients are also used to reconstruct frames to be stored in the frame buffer. Sometimes, there is a deblocking filter before the frame buffer. It can decrease the block artifacts and improve the subjective quality.

Referring to Fig. 1, the comparison results

Modules	Standard		
	MPEG-2	MPEG-4 ASP	H.264 baseline profile
Motion estimation/compensation			
Block size	16 × 16	16 × 16 and 8 × 8	16 × 16, 16 × 8, 8 × 16, 8 × 8, 8 × 4, 4 × 8, and 4 × 4
Quarter-pel precision	No	Yes	Yes
Multiple reference frame	Up to 2	Up to 2	Yes (5 reference frames)
Intraprediction	DC prediction	AC/DC prediction	Yes (9 modes for 4 × 4 blocks and 4 modes for 16 × 16 blocks)
Transform	8 × 8 DCT	8 × 8 DCT	4 × 4 integer transform
Entropy coding	VLC	VLC	VLC and CAVLC
In-loop deblocking filter	No	No	Yes

Table 1. *Comparison of different video coding standards.*

between MPEG-2, MPEG-4 ASP, and the H.264 baseline profile are shown in Table 1. Comparing MPEG-4 ASP to MPEG-2, several new prediction methods are included, such as supporting 8 × 8 blocks, quarter-pel precision motion compensation, AC/DC prediction, and global motion compensation. On the other hand, comparing H.264 and MPEG-4 ASP, it is obvious that the H.264 encoder is much more complicated. Variable block size (VBS), multiple reference frame (MRF), and complicated intraprediction can provide more accurate prediction for current frames. Furthermore, context-adaptive variable-length coding (CAVLC) is included as a new tool in entropy coding, where one of many VLC tables can be selected according to the context of each block. Note that the H.264 baseline profile supports 4 × 4 integer transform instead of 8 × 8 discrete cosine transform (DCT). It can ensure that the transformation of the encoder and decoder are matched in H.264.

VLSI Hardware Architecture of MPEG-4 and H.264 Systems

In this section the VLSI hardware architectures of the key modules in a video compression system are described. We start from instruction profiling of a video encoder to evaluate the importance of each module, and then describe and propose hardware architectures for some important modules. Note that since H.264 is much more complicated than MPEG-4, we put more emphasis on the hardware architecture of H.264.

Instruction Profiling

Before introducing the hardware architecture, we first do instruction profiling of an encoder system. Instruction profiling is based on a reduced instruction set computing (RISC) platform. It can be viewed as a hardware architecture with only one processing element (PE), or an "extremely folded" architecture. This data is valuable for software implementation, hardware implementation, and software/hardware partitioning. For software implementation, it can be used to find the critical module to be optimized. For hardware implementation, it can be used to find the parallelism requirement for each module in order to achieve the given specification. As for software/hardware partitioning, it can be used to roughly find some critical modules that need to be implemented in hardware and some modules whose complexity is small enough to be handled by software executed on a given processor.

Here, we take H.264 as an example. The instruction profiling results are shown in Table 2. The simulation model is standard-compatible software developed by us, and the platform is a SunBlade 2000 workstation with a 1.015 GHz Ultra Sparc II CPU and 8 Gbytes RAM. Arithmetic, controlling, and data transfer instructions are separated in this table. It can be observed that motion estimation, including integer-pel motion estimation, fractional-pel motion estimation, and fractional-pel interpolation in the table, takes up more than 95 percent of the computation in the whole encoder, which is a common characteristic in all video encoders. Among motion estimation, integer-pel motion estimation, which is implemented with a full search algorithm, plays the most important role. The total required computing power for a H.264 encoder is more than 300 giga instructions per second (GIPS), which cannot be achieved by existing processors. Therefore, for H.264 hardware implementation is necessary. Furthermore, the amount of data transfer is more than 460 Gbytes/s, which cannot be achieved by existing memory systems. Most of the data transfer comes from motion estimation. Therefore, the memory architecture of motion estimation must be carefully designed. Note that the simulation model used for this profiling is not optimized software. Hence, some data above may be larger than those of commercial products. However, this can still be valuable information for hardware architecture design.

It is an interesting fact that from MPEG-4, standardized in 1999, to H.264, standardized in 2003, the computational complexity increases

Functions	Arithmetic		Controlling		Data transfer		
	MIPS	%	MIPS	%	MIPS	Mbytes/s	%
Integer-pel motion estimation	95,491.9	78.31	21,915.1	55.37	116,830.8	365,380.7	77.53
Fractional-pel motion estimation	21,396.6	17.55	14,093.2	35.61	30,084.9	85,045.7	18.04
Fractional-pel interpolation	558.0	0.46	586.6	1.48	729.7	1067.6	0.23
Lagrangian mode decision	674.6	0.55	431.4	1.09	880.7	2642.6	0.56
Intra prediction	538.0	0.44	288.2	0.73	585.8	2141.8	0.45
Variable length coding	35.4	0.03	36.8	0.09	44.2	154.9	0.03
Transform and quantization	3223.9	2.64	2178.6	5.50	4269.0	14,753.4	3.13
Deblocking	29.5	0.02	47.4	0.12	44.2	112.6	0.02
Total	121,948.1	100.00	39,577.3	100.00	153,469.3	471,299.3	100.0
(Baseline profile, 30 CIF frames/s, 5 reference frames, ±16-Pel search range, and QP = 20)							

■ **Table 2.** *Instruction profiling results of H.264.*

more than 10 times. The growth speed is higher than Moore's Law, which can only increase computing power about 6.36 times in four years. It implies that it is impossible to achieve real-time requirements with only software. There must be some hardware accelerators integrated in the system to meet the specification.

Motion Estimation

There is no doubt that ME is the most important module in video encoding systems. Because of having the largest complexity and memory bandwidth requirements, it is the first module that has to be accelerated with hardware. For smaller frame sizes on a PC, ME can be accelerated with special instructions, such as Intel MMX. However, for larger frame sizes or real-time embedded devices, hardware implementation of motion estimation is necessary.

For smaller frame sizes and early video coding standards, most conventional architectures are based on array architectures [3], where full search block matching algorithms are implemented. Pixel data of the current frame and reference frame are loaded and propagated in the array, which is composed of several PEs. Two different types of ME parallelism are often considered. The first is interlevel parallelism, where each PE takes charge of one candidate search position (candidate motion vector) in the search range in the reference frame. The other is intralevel parallelism, where each PE takes charge of the distortion calculation of one pixel in the current frame to different pixels with different candidate search positions.

However, when the required frame size becomes larger and larger, the complexity of ME increases dramatically. For example, in the H.264 baseline profile, the motion vector of each block needs to be searched at quarter-pel precision with different block sizes in five reference frames! In this situation an array architecture may be not suitable; the required working frequency and hardware cost may be too high. A tree-based architecture [4], which can be viewed as an intralevel parallel processing architecture, can solve this problem. In a tree-based architecture, the pixel data of the current block are stored in a register file, and the pixel data of the search range are stored in on-chip memory. All the data are fed into the adder tree to calculate one sum of absolute difference (SAD) between the current block and the corresponding block in the search range for one candidate search position. With careful design of the memory architecture and memory access scheme, a tree-based architecture can generate one SAD for one candidate in each cycle. Moreover, fast ME algorithms (e.g., diamond and three-step searches, and successive elimination and partial distortion elimination algorithms) can also be supported in the tree-based hardware architecture.

Variable block size motion estimation (VBSME) can also be supported by the tree-based architecture. For example, to support block sizes of 16 × 16, 16 × 8, 8 × 16, 8 × 8, 8 × 4, 4 × 8, and 4 × 4 for H.264, for each candidate search position we need to calculate the SAD values of the one 16 × 16 block, two 16 × 8 blocks, two 8 × 16 blocks, four 8 × 8 blocks, eight 8 × 4 blocks, eight 4 × 8 blocks, and 16 4 × 4 blocks. That is, we need to calculate 41 SAD values simultaneously. With a tree-based architecture, we can only change the architecture of the adder tree to a multilevel adder tree to support VBSME. The fist level consists of the adder trees to generate SAD values of 4 × 4 blocks (4 × 4 SADs). The second level consists of the adder trees, which accumulate the output of the first level, and 4 × 4 SADs to generate SAD values of 4 × 8 and 8 × 4 blocks. Similarly, the third level can generate SAD values of 8 × 8 blocks from 4 × 8 SADs or 8 × 4 SADs, and the 16 × 8

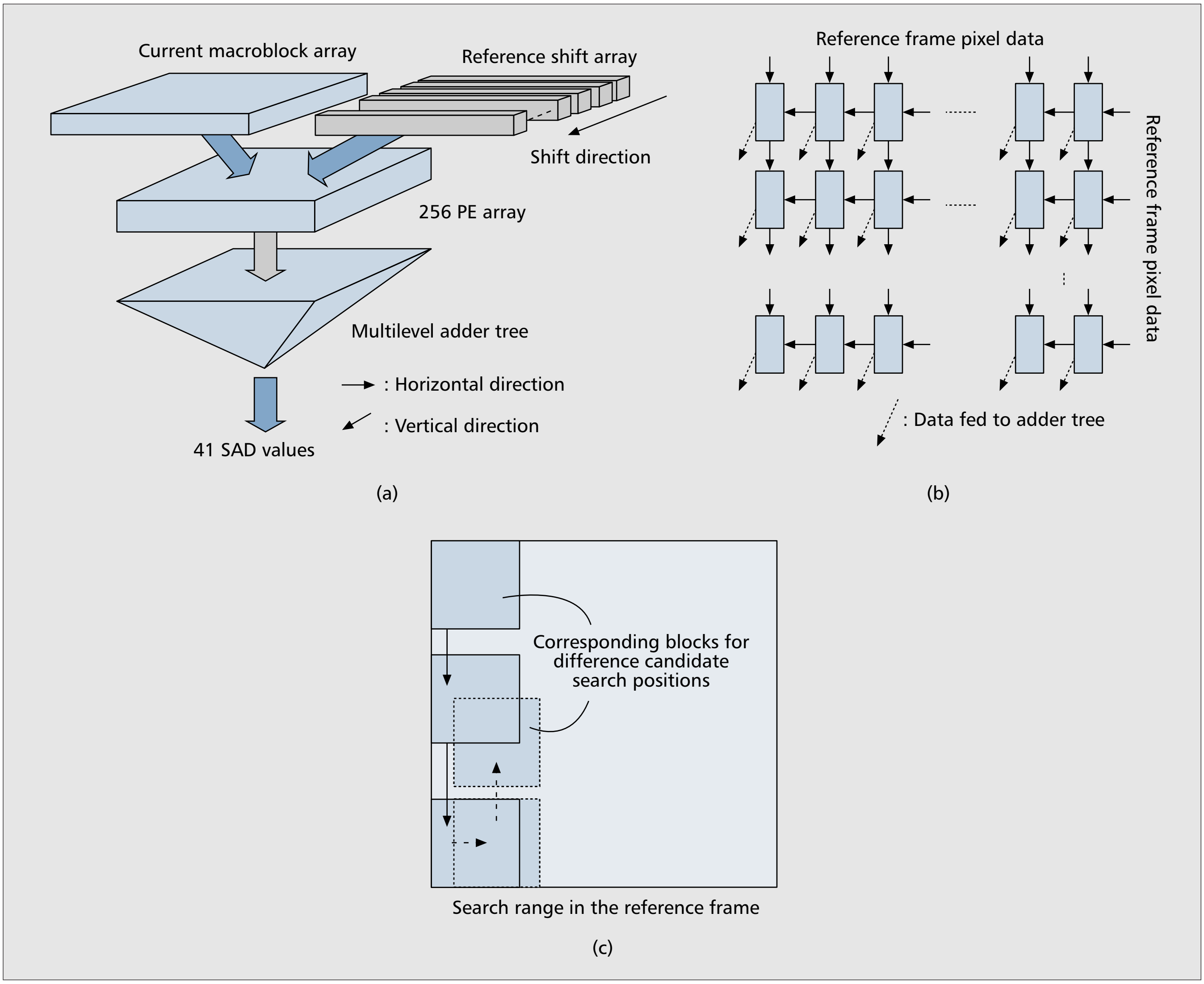

Figure 2. *The proposed hardware architecture of a variable block size motion estimator for H.264: a) proposed hardware architecture; b) hardware architecture of the reference shift array; c) candidate search positions are visited in snake-scan order in the search range.*

SADs, 8 × 16 SADs, and 16 × 16 SADs can also be calculated in the fourth, fourth, and fifth levels, respectively.

Based on this concept, we propose a hardware architecture for integer-pel VBSME for H.264, which is shown in Fig. 2a. The PE array is composed of 256 PEs, and each PE can calculate the absolute difference between one pixel in the current block and one in the search range. By summing up the 256 absolute difference values in the 2D multilevel adder tree, the 41 SAD values are generated at the same time. The reference pixels are loaded into the reference shift array before they are fed into the adder tree. The detailed architecture is shown in Fig. 2b. With this 2D shift register file, the reference pixels can be loaded by following the snake scan order shown in Fig. 2c. That is, for each new search candidate position, only one row or one column needs to be loaded into the reference shift array. Pixel data are reused efficiently with this scheme, which can decrease the memory bandwidth requirement of on-chip memory.

Motion Compensation

Motion compensation is needed for both the video encoder and decoder. The function of MC is to load the corresponding best matched block in the reference frame according to the motion vectors derived by ME. The computational complexity is small and can easily be implemented as a special address generator. However, since fractional motion vectors in quarter-pel and half-pel precision are supported by MPEG-4 and H.264, the fractional pixels need to be generated in MC by interpolating with a 2D FIR filter. With a pixel buffer, the pixel data and some partial results during interpolation can be reused to reduce the memory bandwidth requirement.

Transformation

Transformation, which is typically DCT in video compression systems, can transform pixel data in the spatial domain to the frequency domain for a more compact representation. It is a regular filter operation, matrix multiplication operation, or butterfly operation. If dgital signal processing

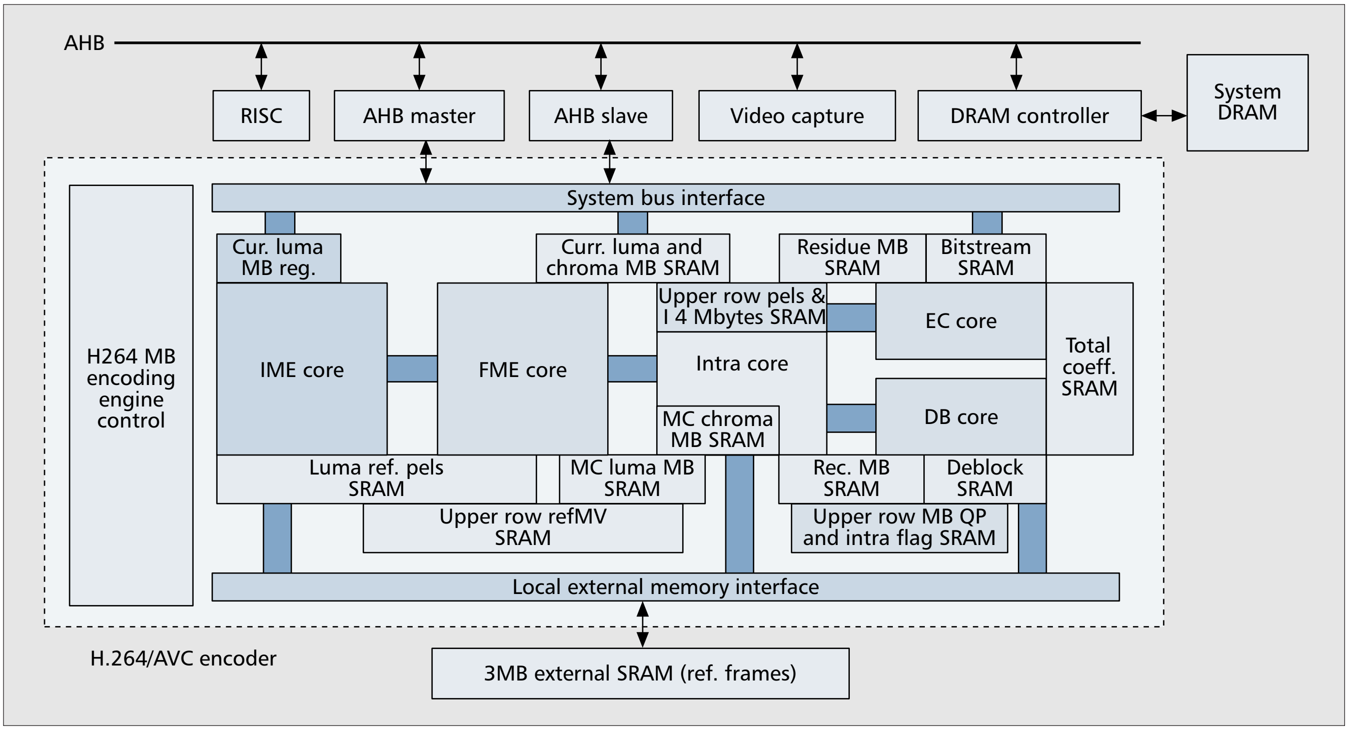

Figure 3. *Hardware architecture of our H.264 encoder.*

(DSP) is available in a system, it is usually able to take charge of transformation.

For hardware implementation of transformation, a 2D DCT can be separated into two 1D DCTs and implemented with a matrix multiplier and transpose memory. With a word-serial-in word-serial-out interface and word-parallel-in word-parallel-out data path, such architecture can achieve high throughput of almost 1 pixel/cycle. The transformation in H.264 is a 4 × 4 integer transform. We have proposed a hardware architecture for such an operation [5]. With 2D shift registers, the proposed architecture can achieve the throughput of 4 pixels/cycle.

Intraprediction

Intraprediction is a new coding tool introduced in H.264. Although the computational complexity of intraprediction is not large, it is still a very complicated operation. For each block, we have to test nine intraprediction modes for each 4 × 4 block and test four intraprediction modes for each 16 × 16 block. Implementing different prediction modes with different hardware architecture is inefficient. The hardware cost would be too large. In [6] we have proposed a hardware architecture for intraprediction in H.264. The hardware for different prediction modes are folded together as a reconfigurable data path. With different configurations, different prediction modes can be achieved on the same hardware.

Deblocking Filter

An in-loop deblocking filter is first introduced in H.261, and H.264 also adopts this coding tool. With a deblocking filter, the block artifacts can be reduced, which can enhance subjective quality. Although the deblocking filter is a related simple operation in an H.264 encoder, it takes up 33 percent of the computation time in the H.264 decoder. The hardware architecture for the deblocking filter was proposed in [7]. It consists of a local memory to store a large window in one frame and a 2D shift register to store a small window for the deblocking filter. Pixel data of the 2D shift register are shifted out to a reconfigurable 1D filter to do horizontal and vertical filtering.

System

As video coding systems become more and more complex, integration is achieved with a platform-based design and a hierarchical bus. The modules of video coding, such as ME, transformation, and entropy coding, are connected to each other with a local bus or local interconnection and can be controlled locally. A global bus is used to access off-chip system memory or on-chip embedded DRAM [8].

There is another important issue for system integration of video compression systems: macroblock pipelining. For MPEG-4 encoders, since the required number of cycles for ME for one macroblock is larger than the summation of those of other modules, two-stage pipelining is usually used. When ME calculates the motion vector for one macroblock, the other modules — MC, DCT, quantization (Q), inverse DCT (IDCT), inverse quantization (IQ), and entropy coding — handle the previous macroblock. Two macroblocks are processed simultaneously in a coding system, which can make controlling this system much easier and make the scheduling more compact to reduce the latency. However, it is not the case for H.264 encoders. Many components, including integer-pel ME (IME), fractional-pel ME (FME), intraprediction (IP), deblocking (DB), and entropy coding (EC), are complicated and require large amounts of cycles to process one macroblock. If two-stage pipelining is used, the throughput and hardware utilization is low. Therefore, four-stage pipelining is proposed [9]. The prediction stage is partitioned into three stages: IME, FME, and IP. Note that the DPCM loop of the video encoder, including forward and inverse transformation and quantization, are also integrated in IP. The reconstructed upper row pixels are stored in a local memory and used by intraprediction. The other operations, DB and EC, belong to the fourth stage. They encode the residual and reconstruct reference frames. With this scheme, four MBs are processed simultaneously, and the processing cycles of the four stages are balanced to achieve high utilization.

Based on the proposed techniques, an H.264 encoding system is integrated as an example (Fig. 3) [10]. The die photo of the chip is shown in Fig. 4. With the UMC 0.18 µm process, the gate count is 922.8K, the size of on-chip memory is 34.72 kbytes, and the chip area is 7.68 × 4.13 mm^2. When the chip operates at 81 MHz, it can process 30 D1 (720 × 480) frames/s with four reference frames. The search range of the first reference frame is [–64,+63] horizontal and [–32,+31] vertical, while the search ranges of the rest of the reference frames are [–32,+32] horizontal and [–16,+15] vertical. It should be noted that the proposed architecture is designed for the H.264 baseline profile. For the main and high profiles, since the operations become even more complex, some modifications of the architecture are required.

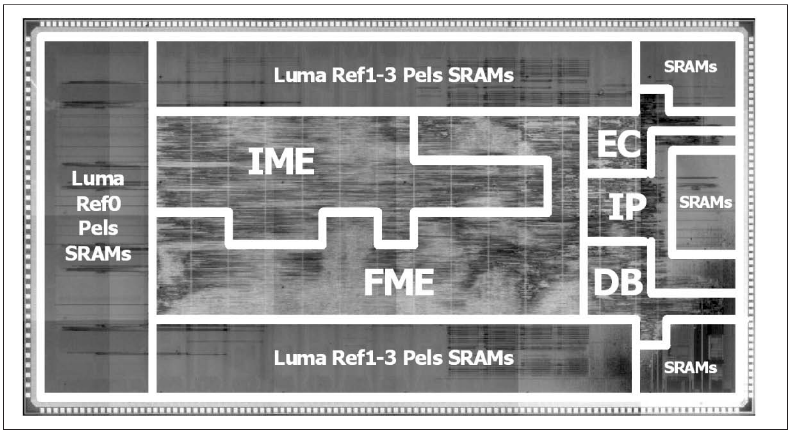

Figure 4. *Die photo of our H.264 encoder.*

Integrating Video Compression Modules into Communication Systems

For real-time multimedia communication systems, video compression hardware, such as the H.264 encoder proposed in the last section, has to be integrated into a system platform as a hardware accelerator. A general architecture of a multimedia communication system is shown in Fig. 5. The host CPU plays an important role in the whole system. It can control all the modules in this system and allocate the memory for different modules. For more complex applications, an operation system is executed on the host CPU as the bridge from application software to hardware. The DSP is usually integrated in the system for audio coding. Some video compression modules, such as transformation and MC can also be implemented with DSP. The audio interface (I/F) can input/output audio data. It is connected with ADC and DAC to microphones and speakers. The display I/F is also a very important module in the system. It connects to the display devices to show the video data. Sometimes, the display I/F is complicated since some 2D and 3D graphics functions may also be integrated into this module, such as scaling the frame for displaying. The network I/F can be viewed as the communication channel of this system. Various kinds of networks (e.g., wired and wireless) can be considered for different applications. The VLC parser and demultiplexer (demux) can decode the transport stream (TS) and separate the streams into video, audio, and system streams. The hardware introduced in the previous section can be integrated into this system as a hardware accelerator for video compression, which is composed of many smaller

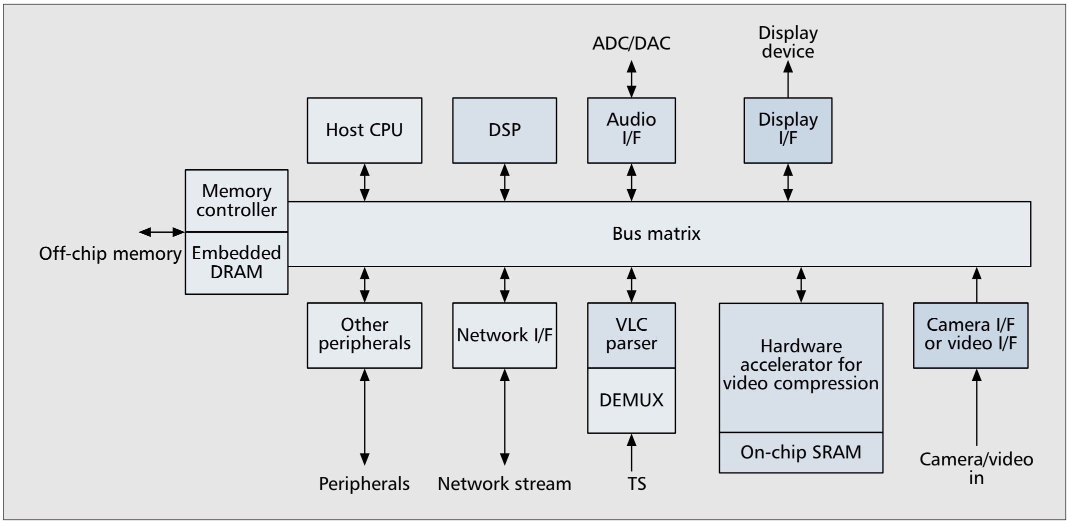

Figure 5. *A general hardware architecture of a multimedia communication system.*

If the system clock frequency is set as 50 MHz, the multiple bus architecture should be more suitable for this system. The system bus is a 32-bit bus, and another 32-bit bus is required as the local bus of the video decoder.

modules and on-chip memory. These modules can be connected to each other via a local bus and controlled by a local controller. Then the controller can communicate with the host CPU to get the instructions. The other alternative is to connect these modules directly to the system bus. In this case, the host CPU can control each module directly. The selection of these architectures depends on the target applications. The camera or video I/F can acquire video frames from an image sensor or input video signal. It sometimes becomes complex if the image signal processing functions need to be implemented in this module. Other than the above modules in the system, the backbone of the system is composed of the memory ccontroller and bus mMatrix. All the modules connect to each other through the bus matrix, and all the modules transfer large amounts of data between each other with the off-chip memory or eembedded DRAM controlled by the memory controller. Note that the whole architecture can be integrated into a single chip as an SoC or partitioned into several blocks integrated on a printed circuit board.

For different applications, the hardware accelerator for video compression and bus matrix can be further modified. The hardware accelerator, which consists of a set of hardware modules, is designed according to the target applications and platforms. For example, if the host CPU has high computing power, such as a CPU with processing speed of 3 GIPS in a PC, only ME related modules need be integrated in the hardware accelerator. The bus matrix can also be modified according to the target applications. For example, for low-end applications (e.g., video decoding with QCIF 30 frames/s), the loading of the bus is light. A single bus connecting all the modules together is enough. However, for high-end applications, the bus matrix can be further partitioned. For example, we can partition it into five buses according to the relationship between different modules: the first one connects to the host CPU, other peripherals, network I/F, and demux; the second one connects to the VLC parser and hardware accelerator for video compression; the third one connects to DSP and the audio I/F for audio processing; the fourth one connects to the camera I/F; the fifth one connects to the display I/F. Modules connected to different buses will not interfere with each other, which can improve the performance of the system. Here are two examples for two applications: IP camera and DVB.

IP Camera

The function of IP cameras is to capture and compress video data acquired from a camera and then stream the data over network. For such applications, the camera I/F is required as the input channel of this system. The DSP and audio I/F are used for audio recording. Since the host CPU needs to take care of video streaming over the network, the computing power of the host CPU may be not enough for video encoding, and all the hardware modules of video coding are required. The network I/F is used as the output channel of this system. The display I/F, VLC parser, and demux can be removed. Because the bit rate and frame size may be limited in this system, the bus loading should be light. For example, for CIF 30 frames/s video with bit rate of 128 kb/s and single-channel 16-bit 44.1 samples/s audio with bit rate of 32 kb/s, the bus bandwidth requirement for data transmission is about 20 Mbytes/s. Considering memory bandwidth for instruction loading and data exchange of program execution, a 16-bit single bus operating at 20 MHz can provide enough bandwidth for this system.

DVB

On the other hand, for DVB applications targeted for HDTV, the VLC parser and demux should be viewed as the input channel of this system. The DSP and audio I/F are used for audio bitstream decoding and playing. For the hardware accelerator, only decoder-related hardware modules are included; that is, ME related modules are removed. The display I/F, which connects to a TV, now becomes the output channel of this system. It should be able to control large screen and support some 2D graphics operations. In this system, the network I/F and camera I/F can be removed. Sincethe bit rate and frame size are both large for this application, the bus loading should be heavy. For example, for 1920 × 1080 30 frames/s video with bit rate of 10 Mb/s and 5.1-channel 16-bit 48 samples/s audio with bit rate of 384 kb/s, the bus bandwidth requirement for data transmission is more than 300 Mbytes/s. If the system clock frequency is set as 50 MHz, the multiple bus architecture should be more suitable for this system. The system bus is a 32-bit bus, and another 32-bit bus is required as the local bus of the video decoder.

Conclusion

In this article we discuss some VLSI hardware implementation issues for multimedia communication systems. First of all, three important video coding standards, MPEG-2, MPEG-4 ASP, and H.264, are compared to show that video coding standards become more and more complicated, and require more and more computing power in order to achieve better coding performance. For H.264 baseline profile with 30 CIF frames/s, the required computing power is more than 300 GIPS. The growth speed is even larger than Moore's Law. Therefore, hardware implementation with special architecture design is required. Next, for MPEG-4 and H.264, we introduce the hardware architectures for their modules and propose hardware architectures for H.264. Motion estimation is the most important module in a video coding system. Using a tree-based architecture with a multilevel adder tree, variable block size motion estimation of H.264 can be realized efficiently. We also propose some architectures for transformation, intraprediction, and deblocking filter. For video coding system integration, a two-stage macroblock pipeline is proposed for MPEG-4, while a four-stage macroblock pipeline is proposed for H.264. With the macroblock pipelining technique, low latency and high throughput can be achieved. We also propose a general hardware architecture for multimedia communication systems. Two applications, IP camera and DVB, are mapped into the proposed architecture as examples.

We believe that VLSI implementation of video compression is the key to making real-time multimedia communications products feasible and affordable for consumers. With well designed video coding modules and system platform, new applications and devices can be developed quickly.

> *We believe that VLSI implementation of video compression is the key to make real-time multimedia communication products feasible and affordable for consumers.*

ACKNOWLEDGMENTS

The authors would like to thank members of the H.264 group at the DSP/IC Design Laboratory for their great help, including Yu-Wen Huang, Tung-Chien Chen, Ching-Yeh Chen, Chen-Han Tsai, and To-Wei Chen.

REFERENCES

[1] Information Technology — Coding of Audio-Visual Object — Part 2: Visual, ISO/IEC Std. 14 496-2, 1999.

[2] ITU-T Rec. H.264 and ISO/IEC 14496-10 Std., "Joint Video Specification," 2003.

[3] P.-C. Tseng *et al.*, "Advances in Hardware Architectures for Image and Video Coding — A Survey," *Proc. IEEE*, vol. 93, no. 1, Jan. 2005, pp. 184–97.

[4] Y.-S. Jehng, L.-G. Chen, and T.-D. Chiueh, "An Efficient and Simple VLSI Tree Architecture for Motion Estimation Algorithms," *IEEE Trans. Sig. Proc.*, vol. 41, no. 2, Feb. 1993, pp. 889–900.

[5] T.-C. Wang *et al.*, "Parallel 4×4 2D Transform and Inverse Transform Architecture for MPEG-4 AVC/H.264," *Proc. IEEE Int'l. Symp. Circuits and Systems (ISCAS '03)*, May 2003, pp. II-800–803.

[6] Y.-W. Huang *et al.*, "Analysis, Fast Algorithm, and VLSI Architecture Design for H.264/AVC Intra Frame Coder," *IEEE Trans. Circuits Sys. Video Tech.*, vol. 15, no. 3, Mar. 2005, pp. 378–401.

[7] Y.-W. Huang *et al.*, "Architecture Design for Deblocking Filter in H.264/JVT/AVC," *Proc. IEEE Int'l. Conf. Multimedia and Expo*, July 2003, pp. I-693–96.

[8] T. Fujiyoshi *et al.*, "An H.264/MPEG-4 Audio/Visual Codec LSI with Module-wise Dynamic Voltage/Frequency Scaling," *IEEE Int'l. Solid-State Circuits Conf. Dig. Tech. Papers*, Feb. 2005.

[9] T.-C. Chen, Y.-W. Huang, and L.-G. Chen, "Analysis and Design of Macroblock Pipelining for H.264/AVC VLSI Architecture," *Proc. IEEE Int'l. Symp. Circuits and Sys.*, May 2004, pp. II-273–76.

[10] Y.-W. Huang *et al.*, "A 1.3 TOPS H.264/AVC Single-chip Encoder for HDTV Applications," *IEEE Int'l. Solid-State Circuits Conf. Dig. Tech. Papers*, Feb. 2005.

BIOGRAPHIES

SHAO-YI CHIEN (sychien@cc.ee.ntu.edu.tw) received B.S. and Ph.D. degrees from the Department of Electrical Engineering, National Taiwan University (NTU), Taipei, in 1999 and 2003, respectively. From 2003 to 2004 he was a research staff member at the Quanta Research Institute, Tao Yuan Shien, Taiwan. In 2004 he joined the Graduate Institute of Electronics Engineering and Department of Electrical Engineering, NTU, as an assistant professor. His research interests include video segmentation algorithms, intelligent video coding technology, image processing, computer graphics, and associated VLSI architectures.

YU-WEN HUANG received a B.S. degree in electrical engineering and a Ph. D. degree from the Graduate Institute of Electronics Engineering at NTU in 2000 and 2004, respectively. He joined MediaTek, Inc., Hsinchu, Taiwan, in 2004, where he develops integrated circuits related to video coding systems. His research interests include video segmentation, moving object detection and tracking, intelligent video coding technology, motion estimation, face detection and recognition, H.264/AVC video coding, and associated VLSI architectures.

CHING-YEH CHEN received a B.S. degree from the Department of Electrical Engineering, NTU in 2002. He is currently pursuing a Ph.D. degree at the Graduate Institute of Electronics Engineering, NTU. His research interests include intelligent video signal processing, motion estimation, scalable video coding, and associated VLSI architectures.

HOMER H. CHEN [S'83, M'86, SM'01, F'03] received a Ph.D. degree from the University of Illinois at Urbana-Champaign in electrical and computer engineering. Since August 2003 he has been with the College of Electrical Engineering and Computer Science, NTU, where he is Irving T. Ho Chair Professor. Prior to that, he held various R&D management and engineering positions in leading U.S. companies such as AT&T Bell Labs, Rockwell Science Center, iVast, Digital Island, and Cable & Wireless over a period of 17 years. He was a U.S. delegate to ISO and ITU standards committees, and contributed to the development of many new interactive multimedia technologies that are now part of the MPEG-4 and JPEG-2000 standards. His professional interests lie in the broad area of multimedia signal processing and communications. He is an Associate Editor of *IEEE Transactions on Circuits and Systems for Video Technology*. He served as Associate Editor for *IEEE Transactions on Image Processing* from 1992 to 1994, Guest Editor for *IEEE Transactions on Circuits and Systems for Video Technology* in 1999, and Editorial Board Member for *Pattern Recognition* from 1989 to 1999.

LIANG-GEE CHEN [F] received B.S., M.S., and Ph.D. degrees in electrical engineering from National Cheng Kung University in 1979, 1981, and 1986, respectively. He was an instructor (1981–1986) and associate professor (1986–1988) in the Department of Electrical Engineering, National Cheng Kung University. During his military service during 1987 and 1988, he was an associate professor at the Institute of Resource Management, Defense Management College. In 1988 he joined the Department of Electrical Engineering, NTU. From 1993 to 1994 he was a visiting consultant at the DSP Research Department, AT&T Bell Labs, Murray Hill. In 1997 he was d visiting scholar du the Department of Electrical Engineering, University of Washington, Seattle. Currently, he is a professor at NTU. Since 2004 he is also executive vice president and general director of the Electronics Research and Service Organization of the Industrial Technology Research Institute. His current research interests are DSP architecture design, video processor design, and video coding systems. He is also a member of the honor society Phi Tan Phi. He was general chairman of the 7th VLSI Design CAD Symposium. He was also general chairman of the 1999 IEEE Workshop on Signal Processing Systems: Design and Implementation. He has served as Associate Editor of *IEEE Transactions on Circuits and Systems for Video Technology* from June 1996 to the present and Associate Editor of *IEEE Transactions on VLSI Systems* from January 1999 to the present. He has been Associate Editor of the *Journal of Circuits, Systems, and Signal Processing* from 1999 to the presnt. He served as Guest Editor of the *Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology*, November 2001. He is also Associate Editor of *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*. Since 2002 he is also Associate Editor of *Proceedings of the IEEE*. He received the Best Paper Award from ROC Computer Society in 1990 and 1994. From 1991 to 1999 he received Long-Term (Acer) Paper Awards annually. In 1992 he received the Best Paper Award of the 1992 Asia-Pacific Conference on Circuits and Systems in the VLSI design track. In 1993 he received the Annual Paper Award of Chinese Engineer Society. In 1996 he received the Out-standing Research Award from NSC and the Dragon Excellence Award for Acer. He was elected as an IEEE Circuits and Systems Distinguished Lecturer, 2001–2002.