# Algorithm and Architecture Design of Power-Oriented H.264/AVC Baseline Profile Encoder for Portable Devices

Yu-Han Chen, Tung-Chien Chen, Chuan-Yung Tsai, Sung-Fang Tsai, and Liang-Gee Chen, *Fellow, IEEE*

***Abstract*— Because video services are becoming popular on portable devices, power becomes the primary design issue for video coders nowadays. H.264/AVC is an emerging video coding standard which can provide outstanding coding performance and thus is suitable for mobile applications. In this paper, we target a power-efficient H.264/AVC encoder. The main power consumption in an H.264/AVC encoding system is induced by data access of motion estimation (ME). At first, we propose hardware-oriented algorithms and corresponding parallel architectures of integer ME (IME) and fractional ME (FME) to achieve memory access power reduction. Then, a parameterized encoding system and flexible system architecture are proposed to provide power scalability and hardware efficiency, respectively. Finally, our design is implemented under TSMC 0.18 $\mu$m CMOS technology with 12.84 mm$^2$ core area. The required hardware resources are 452.8 K logic gates and 16.95 KB SRAMs. The power consumption ranges from 67.2 to 43.5 mW under D1 ($720 \times 480$) 30 frames/s video encoding, and more than 128 operating configurations are provided.**

***Index Terms*— H.264/AVC, low-power designs, parallel architectures, video codecs.**

## I. Introduction

TRADITIONALLY, architectural design focuses more on hardware costs and throughput. However, power has become the primary design issue nowadays [1]. "If media coding power dissipation increases beyond a modest 100 mW, it will be hard to implement the media application in portable devices [2]." Therefore, low power consumption is the first goal for power-oriented video encoders in mobile systems. A low-power video encoder targets optimization of coding performance under the specific power constraint. Beyond low power, power-aware [3], [4] is another emerging issue recently. A power-aware encoder can adjust power consumption in response to different conditions, like users' preferences and battery states. In a power-rich condition, a high-quality service is preferred in spite of higher power consumption. On the contrary, users may tolerate poorer quality services with lower power consumption to extend the service time. Fig. 1 shows the battery discharging effects of three kinds of video encoders. A battery has two important properties: the rate capacity effect and the recovery effect [5]. The capacity of a battery is dependent on the discharging rate. The low-power encoder **B** consuming 50% of power of the general encoder **A** can have more than double battery lifetime. We can further extend the battery lifetime with the power-aware encoder **C** by gradually stepping down the power dissipation because the battery capacity can be recovered with a lower loading.

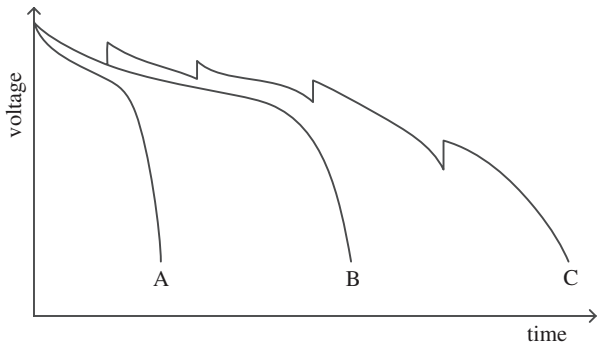


Fig. 1. Battery discharging effects of three kinds of video encoders. **A** is a general encoder. **B** is a low-power encoder with 50% power consumption as compared to **A**. **C** is a low-power and power-aware encoder.

H.264/AVC [6] is one of the most important video coding standard nowadays. Compared to the prior standards, H.264/AVC can save approximately 50% of bit rate with equivalent visual quality [7] and thus is practical for bandwidth-limited mobile applications. However, the outstanding coding performance comes from the adopted complex coding tools. The considerable computational complexity makes H.264/AVC starve for power and is difficult to be realized in battery-powered portable devices. For this reason, we undertake the implementation of a power-efficient video encoder to make H.264/AVC possible for future mobile applications.

There have been several designs of H.264/AVC encoders proposed in the literature. However, most of them, such as [8]–[10], are targeted at HDTV applications and not optimized for low-power portable devices. For example, [11] and [12] are both power-efficient designs. However, few power modes are supported under a fixed frame resolution like D1 ($720 \times 480$). For mobiles applications, there are various devices with different video contents, battery capacity, environmental

Manuscript received January 28, 2008; revised August 9, 2008 and October 27, 2008. First version published April 7, 2009; current version published August 14, 2009. This work was supported in part by the National Science Council of Taiwan, under Grant NSC95-2752-E-002-008-PAE. This paper was recommend by Associate Editor M. Comer.

The authors are with the Graduate Institute of Electronics Engineering and Department of Electrical Engineering, National Taiwan University, Taipei, 10617 Taiwan (e-mail: doliamo@video.ee.ntu.edu.tw; djchen@video.ee.ntu.edu.tw; cytsai@video.ee.ntu.edu.tw; bigmac@video.ee.ntu.edu.tw; lgchen@video.ee.ntu.edu.tw).

Digital Object Identifier 10.1109/TCSVT.2009.2020323

conditions, and users' preferences. There are many coding tools in H.264/AVC. If we can adaptively enable/disable some of the coding tools according to the incoming video contents, we can support more usable power modes to improve the system flexibility and meet the demands.

For a video coding system, low power can be achieved with optimized algorithms and architectures in ASIC designs. Motion estimation (ME) is the most power-consuming part of a video encoder. To realize a low-power video encoder, we have to realize a low-power ME design first. We first apply efficient data reuse techniques at both the algorithm and architecture levels to save memory access power. Then, we propose fast ME algorithms which are aware of video contents to reduce computation and maintain coding performance. On the other hand, to achieve a power-aware video encoding system, multiple power modes are provided to fulfill different power constraints. A parameterized video encoder is first presented to provide more than 128 operating configurations under a fixed frame resolution. Then, we propose a three-layer system architecture which can provide a flexible system schedule and fine-grained module-wise clock gating. In conclusion, with the integration of low-power techniques at the algorithm, architecture, and circuit levels, we can achieve a low-power design. In addition, with parameter-controlled reconfigurable design and flexible system architecture, we can further achieve a power-aware video encoder.

The rest of this paper is organized as follows. In Section II, we will introduce the fundamental knowledge for design of a power-efficient video encoder. The proposed integer motion estimation (IME) and fractional motion estimation (FME) designs for a low-power H.264/AVC encoder are introduced in Sections III and IV, respectively. In Section V, a parameterized power-scalable H.264/AVC encoding system is provided. In Section VI, we propose a three-layer system architecture to improve flexibility and power efficiency of the whole encoding system. Finally, the implementation results will be shown in Section VII, and we will draw a conclusion in Section VIII. Some ideas of the algorithm and architecture design of the proposed encoding system have been published in the previous works [13]–[18]. The main contributions of this paper are the novel one-pass FME algorithm and architecture, the parameterized power-scalable system, the three-layer system architecture, and the integration of the whole encoding system.

## II. Fundamental Knowledge

In this section, we first introduce ME, which is the most critical coding tool of a video encoding system. Then, we will show the design techniques to realize a power-efficient video encoding system. Finally, the power–performance curve is illustrated to show the role of a power-aware video encoder.

### A. Motion Estimation

ME is a powerful coding tool used to remove temporal redundancy in a video sequence. For each macroblock (MB) in the current frame, a best matching block inside the search window of the reference frames is searched for as the predictors of the current MB. The coding performance is better if the predicted block is more accurate. In H.264/AVC, multiple reference frame, variable block size (VBS), and quarter-pixel precision ME algorithms are adopted to improve coding performance, but they lead to heavy computational load. According to our software profiling, ME with the full search algorithm [19] spends more than 90% of computation. Therefore, a low-power ME design is vital for a mobile video application.

### B. Power Reduction Techniques

The full search algorithm for ME is brute force and wasteful. At the algorithm level, computation-economical algorithms are necessary to reduce data processing power. In a video frame, some objects are still but some are moving fast. We can spend more searching efforts in the region with fast moving objects to find better matching blocks. On the contrary, much computation can be saved in the region with still objects with less quality drop. With the content-aware algorithm, we can not only save processing power but also maintain coding performance.

Data access is usually a significant part of power consumption (50–80%) in a signal processing system [20]. For this reason, reduction of data access power is a critical issue. In the ME algorithm, some data read from memory for one searching candidate may also be required by other candidates in the future. Data reuse, i.e., recycling previous accessed data, is an important technique to reduce data access power. At the architecture level, we usually design suitable parallel architecture and data processing flows for data reuse.

In a video encoding system, processing engines (PEs) are sometimes inactive. However, the idle engines still consume power because of the propagated switching activity from the clock. For this reason, a power management system is utilized to monitor the status of each PE and gate the relative clock network once a module enters the idle state. The module-wise clock gating technique is beneficial for a power-aware design. When a power-aware system is set to an ultra low-power configuration, fast algorithms usually make PEs idle in most of the processing cycles. Under this condition, module-wise clock gating can save much idle power.

### C. Power–Performance Curve

A power–performance curve of different video encoding systems is shown in Fig. 2. Point **A** is a real-time video encoder without power optimization. All coding tools are supported, and thus the best coding performance (or visual quality) is provided. However, the power consumption is ultrahigh. Point **B** is an encoder with architectural optimization like in [8]. Parallel processing and data reuse techniques are originally adopted to improve throughput for real-time applications but also contribute to power reduction. The coding performance of **B** is maintained as good as that of **A**. However, the encoder is not suitable for portable devices because the brute-force algorithms still consume too much power. Point **C** is another encoder with algorithmic and architectural co-optimization. Hardware-oriented and content-aware fast algorithms are developed to reduce redundant computation with the consideration of hardware efficiency. The coding performance of **C** is optimized under the specific power constraint and is close to that of **B**. Point **D** is an ultra low-power encoder. Most

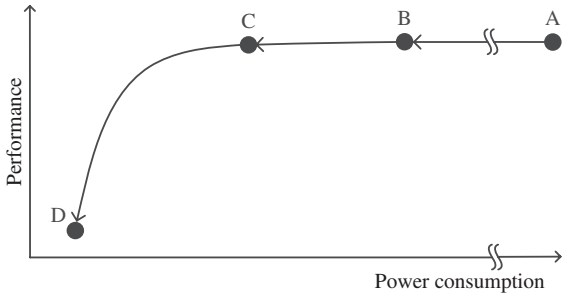


Fig. 2. Power–performance curve of different real-time video encoding systems. Point **A** is a general encoder without optimization. Point **B** is an architecture-optimized encoder. Point **C** is an encoder with algorithmic and architectural co-optimization. Point **D** is an ultra low-power encoder with minimized computation but the worst coding performance.

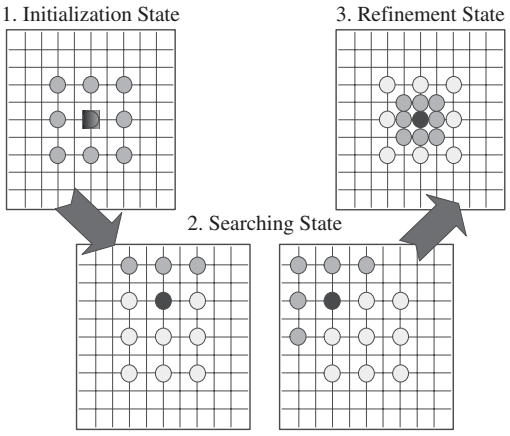


Fig. 3. Searching flow of four-step search. In the initialization state, $3 \times 3$ candidates with the step of two pixels are searched. In the searching state, the searching pattern moves according to the best matching candidate in the previous iteration. Finally, if the best matched candidate is the central point, the refinement is performed around the neighboring eight candidates.

of the coding tools are off and computation is minimized. In this case, the coding performance is the worst and not acceptable in the normal condition. However, when the battery is running out, the ultra low-power encoder **D** is useful to extend the battery lifetime. In this paper, we target a power-aware video encoder that can provide multiple operating configurations between point **C** and point **D** and thus can adapt to different environmental conditions for mobile applications.

## III. Integer Motion Estimation

In this section, we will introduce the proposed IME design which integrates the low-power design techniques at the algorithm level and the architecture level. At first, a hardware-oriented fast algorithm is adopted to improve data reuse capability. Then, a content-aware algorithm is proposed to achieve good tradeoff between coding performance and computation complexity [13]. Finally, a parallel architecture and a memory organization technique are designed to support the proposed algorithm effectively [14].

### A. Hardware-Oriented Algorithm

To implement a power-efficient hardware design, we should take the data reuse issues into consideration at the algorithm level. Two data reuse techniques are adopted in our IME algorithm and will be introduced in the following.

For H.264/AVC, predictive (P) MBs can be partitioned using seven different block types ($16 \times 16$, $16 \times 8$, $8 \times 16$, $8 \times 8$, $8 \times 4$, $4 \times 8$, and $4 \times 4$) which results in 41 subblocks for each P MB. A sequential flow is adopted to find the best matching candidates of the subblocks in the reference software [19], and it cannot achieve efficient data reuse for hardware implementation. To solve this problem, a modified parallel-VBS-IME algorithm has been adopted in many designs [15], [21]–[23]. The parallel-VBS-IME algorithm computes all matching costs of different block-sizes with the same motion vectors (MVs) simultaneously. For a searching candidate, the costs of $4 \times 4$ blocks are computed first, and all other cost of larger block sizes are calculated by summing up the corresponding $4 \times 4$ costs immediately. As a result, the matching costs of the smaller block sizes are reused by the larger block sizes. This technique is called intra-candidate data reuse.

For a searching candidate, a block of reference pixels are loaded from memory for block matching. However, a large portion of reference pixels is overlapped between the neighboring candidates. For two horizontally neighboring candidates of a $16 \times 16$ block, $16 \times 15$ reference pixels are overlapped and can be shared. The technique in which the overlapped reference pixels of neighboring searching candidates are reused is called as inter-candidate data reuse.

Four-step search (FSS) is chosen as the base of our fast algorithm. The search flow of FSS is shown in Fig. 3. Its regular flow is good for inter-candidate data reuse. Because parallel-VBS-IME is also adopted, our algorithm is called parallel-VBS-FSS. At the start, an initial candidate is set as the center of the $3 \times 3$ searching pattern in the initialization state. Then, the searching pattern moves according to the locally best candidate of the $16 \times 16$ block in the searching state. Finally, once the locally best candidate is at the center of the searching pattern, the refinement state is started to find the final best integer MVs.

### B. Content-Aware Strategy

In a video frame, there are regions with complex motion where multiple objects move to different directions in an MB. Because our parallel-VBS-FSS algorithm moves the searching pattern according to the locally best candidate of the $16 \times 16$ block, MVs of some smaller blocks may be not accurate. In this way, coding performance will be degraded. In order to provide robust coding efficiency, more initial candidates should be involved. However, more computation is required. The neighboring motion activity, i.e., the MV difference in the neighboring region, can be exploited to achieve good tradeoff between performance and computation. If motion activity is high (i.e., neighboring MVs are quite different), we should set more initial candidates to find the accurate MVs, and vice versa.

The proposed multi-iteration parallel-VBS-FSS algorithm is shown in Fig. 4. There are six initial searching candidates. The origin (0, 0) is a must because there are usually blocks with

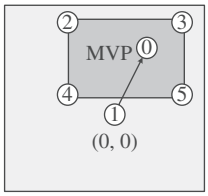


Fig. 4. Multi-iteration algorithm. The light gray region is the search window, and the dark gray region is the predicted motion window generated from the neighboring motion information. For example, the right boundary of the moving window can be set as the largest horizontal motion vector of the neighboring blocks. The number in the circle represents the processing order of the initial candidates.

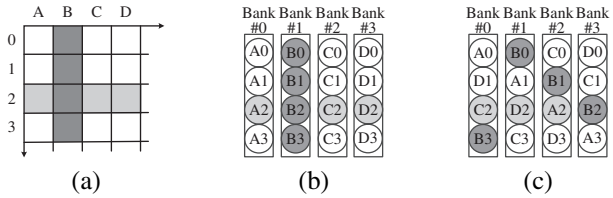


Fig. 5. Data organization of the search window memory. In the example, the number of the memory bank is 4. (a) Physical location of reference pixels in the search window. (b) Conventional data organization with one-directional random access. A row of four reference pixels can be accessed in parallel. (c) Ladder-shaped data organization with two-directional random access. A row or a column of four reference pixels can be parallelly accessed.

less motion in the background region. The second candidate is the MV predictor (MVP), which is defined as the median MV of left, up, and up-right blocks. In the regions with regular motion, the best MV is usually near MVP. The rest of the four initial searching candidates are used to find good matching in a complex motion region where smaller blocks move in different directions. They are located at the four corners of a predicted motion window (PMW) for which boundaries are defined in (1). In the equation, $MV_1$, $MV_2$, and $MV_3$ are MVs of the left, up, and up-right neighboring blocks

$$\begin{aligned} Bound_{\text{up}} &= \max(MV_{y1}, MV_{y2}, MV_{y3}) \\ Bound_{\text{down}} &= \min(MV_{y1}, MV_{y2}, MV_{y3}) \\ Bound_{\text{left}} &= \min(MV_{x1}, MV_{x2}, MV_{x3}) \\ Bound_{\text{right}} &= \max(MV_{x1}, MV_{x2}, MV_{x3}). \end{aligned} \tag{1}$$

To further reduce computation but maintain coding performance, we adopt a content-adaptive searching strategy. The PMW will be adaptively shrunk according to the neighboring motion activity to reduce the number of initial searching candidates. The criteria are shown in (2). If one criterion is met, two initial searching candidates are reduced. If two criteria are both met, only (0, 0) and MVP are set as the initial searching candidates. The possible number of initial candidates is two, four, or six

$$\begin{aligned} &IF\ \ |MV_{x1} - MV_{x2}| \le T\ \ AND\ |MV_{x2} - MV_{x3}| \le T\ \ AND \\ &\quad |MV_{x1} - MV_{x3}| \le T\ \ THEN \\ &\quad Bound_{\text{left}} = Bound_{\text{right}} = median(MV_{x1}, MV_{x2}, MV_{x3}) \\ &IF\ \ |MV_{y1} - MV_{y2}| \le T\ \ AND\ |MV_{y2} - MV_{y3}| \le T\ \ AND \\ &\quad |MV_{y1} - MV_{y3}| \le T\ \ THEN \\ &\quad Bound_{\text{up}} = Bound_{\text{down}} = median(MV_{y1}, MV_{y2}, MV_{y3}). \end{aligned} \tag{2}$$

Threshold $T$ could be adjusted to achieve tradeoff of computation and coding performance. A higher $T$ value makes PMW shrink more easily. As a result, computation is lower but coding performance is poorer. In the current design, $T$ is set as 0 to maintain coding performance.

Finally, about 98% searching candidates of a $\pm 32$ search window for CIF (352 × 288) 30 frames/s video encoding can be reduced with at most 0.05 dB quality drop compared with the full search algorithm [13].

### C. Architecture

Unlike the full search algorithm, the adopted fast algorithm will lead to irregular searching paths. The searching candidate will conditionally move vertically or horizontally. We need flexible memory access to support efficient data reuse. In other words, reference pixels in the search window memory should be organized to support flexible random access. The physical location of the reference pixels in the search window is shown in Fig. 5(a). In order to achieve parallel memory access, reference pixels are conventionally organized in the pattern as shown in Fig. 5(b). The first column of reference pixels are placed in the memory bank #0. The second column of pixels are placed in the memory bank #1, and so on. In this way, a row of reference pixels, like "A2–D2," can be read in parallel. However, a column of reference pixels, like "B0–B3," cannot be accessed in parallel because they are in the same memory bank. To solve the problem, we adopt a ladder-shaped pattern for memory arrangement as shown in Fig. 5(c). The reference pixels of the second, third, and fourth rows are rotated rightward by one, two, and three pixels. In this way, the reference pixels of "A2–D2" or "B0–B3" are arranged in different memory banks and can be accessed in parallel. As a result, two-directional random access for a row or a column of reference pixels is supported. In fact, 16 memory banks are used in our design to support a row/column of 16-pixel memory access in parallel.

The proposed IME architecture is illustrated in Fig. 6. The reference and the current frames are originally stored in the external memory. Before the process of ME, the required data of the current and reference MBs are preloaded from external memory and respectively stored into the "16 × 16 Curr-pel Buffer" and "Ladder-shaped Search Window SRAMs." In order to support two-directional random access, reference pixels are ladder-shaped arranged in the search window SRAMs. In the process of ME, a row or a column of reference pixels are read from search window memory and stored into "16×16 Ref-pel Shift Register Array." Four configurations are supported in the shift register array for upward, downward, leftward, and rightward movement of the current searching candidate. The 16×16 current and reference pixels are simultaneously fetched into "256 PE Array" to compute the absolute difference values. Then, the intermediate results are input to "2-D Adder Tree" to compute the sum of absolute difference (SAD) values from 4 × 4 to 16 × 16 blocks. Finally, the minimal SAD values of all block sizes are generated.

Intra-candidate data reuse can be achieved with the parallel-VBS-IME algorithm under the 2-D adder tree architecture.

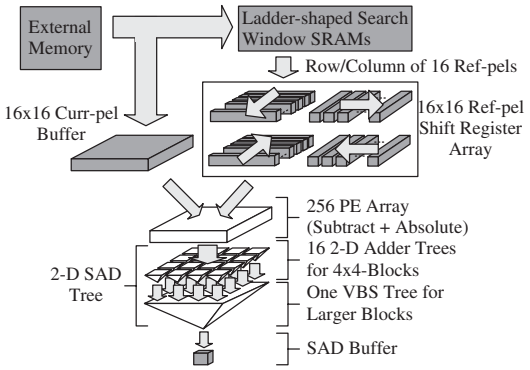


Fig. 6. Architecture of integer motion estimation. The dark gray parts are the memory units. The light gray parts are the data flows. The white parts are the processing units.

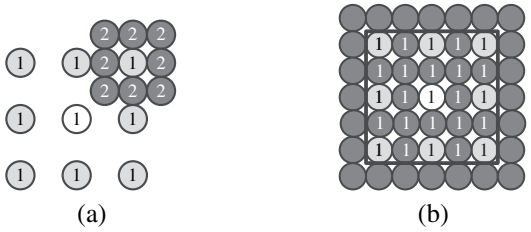


Fig. 7. Illustration of fractional motion estimation algorithm. The white circles are the best integer-pixel candidates. The light gray circles are the half-pixel candidates. The dark gray circles are the quarter-pixel candidates. The circles labeled "**1**" and "**2**" are the candidates refined in the first and second passes, respectively. (a) Conventional two-step algorithm and (b) proposed one-pass algorithm. The 25 candidates insides the dark square are processed in parallel.

The SAD cost of seven block sizes is computed in parallel and thus the amount of memory access can be reduced to 1/7. With ladder-shaped search window data organization and the shift register array with four configurations, we can support inter-candidate data reuse in upward, downward, leftward, and rightward directions. According to our analysis, about 78% of memory access power of IME can be further saved [14].

## IV. Fractional Motion Estimation

After optimization of IME, FME becomes the most power-consuming part of an H.264/AVC encoder. Therefore, a power-efficient FME design is vital. At first, the advanced mode pre-decision algorithm [17] is presented to reduce computation. Then, the hardware-oriented one-pass algorithm is proposed to reduce 50% memory access with simultaneous half-pixel and quarter-pixel refinement. Finally, the corresponding parallel architecture with good data reuse capability is shown.

### A. Advanced Mode Pre-Decision

In the reference software [19], all the required reference pixels with half-pixel and quarter-pixel precision are interpolated in advance and stored in the system memory. Therefore, the interpolated data can be reused for FME refinement of seven kinds of block sizes. However, the required memory is 16 times the frame size, which is a considerable area overhead for hardware implementation. In addition, heavy system bandwidth is induced in order to load the reference pixels during the process of FME. As a result, the online interpolation architecture is generally adopted [16]. Under this architecture, the interpolated reference pixels are repeatedly generated online for different block sizes. In this case, more refined inter-modes (block sizes) lead to more power consumption.

An advanced mode pre-decision algorithm [17] is adopted to save power here. In the full search FME algorithm of the reference software [19], the best inter-mode is decided after all the modes are refined to the quarter-pixel precision. In the proposed advanced mode pre-decision algorithm, $N$ best modes ($N = 0 - 7$) are pre-decided after IME with integer-pixel precision. Then, only the $N$ best modes are refined to quarter-pixel precision. "$N = 3$" is chosen in our highest quality configuration for FME to reduce about half the computation. If more power reduction is required, $N$ could be set to 2, 1, or 0. When $N$ is set to 0, MVs with only integer-pixel accuracy are supported.

### B. One-Pass Algorithm

In the reference software, a two-step algorithm is adopted as illustrated in Fig. 7(a). Half-pixel candidates around the best integer-pixel candidate are first refined. Then, the quarter-pixel candidates around the best half-pixel candidate are further refined to find the best MV. As a result, 17 candidates are searched to find the best matching candidates. In H.264/AVC, a six-tap interpolation filter is adopted to generate the half-pixel reference data from integer-pixel reference data. The quarter-pixel reference data are generated from the standard defined neighboring half-pixel reference data with a bilinear filter. The sum of absolute transformed difference (SATD) is used as the matching cost of FME and defined as follows. The difference values of the current block and the interpolated reference block are computed first and then processed with the Hadamard transform (HT). The resulting data are called transformed residues. Finally, the absolute values of the transformed residues are accumulated to generate the SATD cost.

The required interpolation windows for half-pixel and quarter-pixel candidates are overlapped. If the conventional two-step algorithm is adopted, the overlapped reference data are loaded twice and wasteful. Therefore, we propose a hardware-oriented one-pass algorithm. The main concept is that the half-pixel and quarter-pixel candidates are processed simultaneously to share the memory access data and thus reduce data access power. There are 49 fractional-pixel candidates in all for FME, comprising of one integer-pixel, eight half-pixel, and 40 quarter-pixel candidates as shown in Fig. 7(b). If all the candidates are searched, the computation is 2.88 (49/17) times the two-step algorithm. However, according to our simulation, 87% of the best matching candidates are located at the central 25 candidates as shown in Fig. 7(b). Therefore, we skip the marginal 24 candidates to save computation.

In order to further reduce the data processing power of the quarter-pixel candidates, the linearity of HT is utilized. Equation (3) shows the linearity of HT. $HT(\cdot)$ means the HT function. $a$ and $b$ are scalars, and $\mathbf{A}$ and $\mathbf{B}$ are two $4 \times 4$ blocks of reference data. In (4), $\mathbf{Q}$ is a $4 \times 4$ block

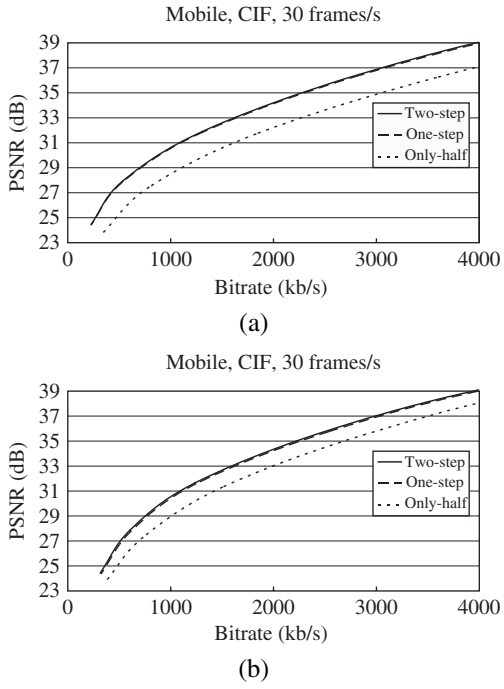


Fig. 8. Rate-distortion performance of the proposed one-pass FME algorithm. The solid, dashed, and dotted lines show the performance of the two-step algorithm in the reference software, the proposed one-pass algorithm, and the algorithm with only half-pixel refinement.

of a quarter-pixel candidate and it is bilinearly interpolated from two 4 × 4 blocks (**A** and **B**) of half-pixel candidates. $Round(\cdot)$ means the rounding function. In (5), **U** is the 4 × 4 current block. Due to the linearity, the transformed residues of **Q** can be approximated by the bilinear interpolation of the transformed residues of **A** and **B** (only the rounding effect is nonlinear). With the approximation, data processing power for HT of all quarter-pixel candidates is saved. The simulation results of the proposed one-pass FME algorithm are shown in Fig. 8. Compared to the two-step algorithm in the reference software [19], the performance degradation is only a 0.06 dB quality drop in average for CIF (352 × 288) 30 frames/s video encoding. In addition, the required memory access of the proposed one-pass algorithm is the same as with the only half refinement algorithm (i.e., MVs are only refined to the half-pixel precision), but the rate-distortion performance is much better

$$HT(a\ \mathbf{A} + b\ \mathbf{B}) = a\ HT(\mathbf{A}) + b\ HT(\mathbf{B}) \tag{3}$$

$$\mathbf{Q} = Round\left(\frac{\mathbf{A}+\mathbf{B}}{2}\right) \tag{4}$$

$$\begin{aligned} HT(\mathbf{Q}-\mathbf{U}) &= HT\left(Round\left(\frac{\mathbf{A}+\mathbf{B}}{2}\right)-\mathbf{U}\right) \\ &\approx HT\left(Round\left(\frac{\mathbf{A}-\mathbf{U}+\mathbf{B}-\mathbf{U}}{2}\right)\right) \\ &\approx Round\left(\frac{HT(\mathbf{A}-\mathbf{U})+HT(\mathbf{B}-\mathbf{U})}{2}\right). \end{aligned} \tag{5}$$

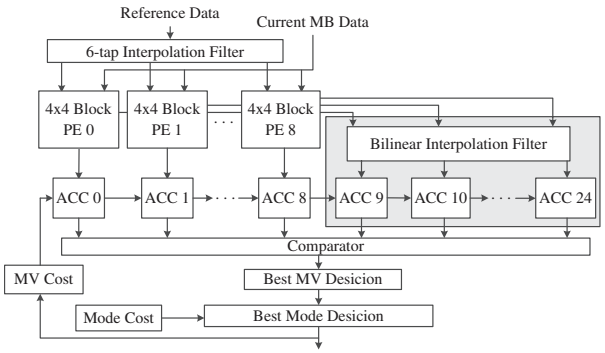


Fig. 9. Architecture of fractional motion estimation. The processing engines on the left side are used to generate the matching costs of integer-pixel and half-pixel candidates. The transformed residues are reused to generate the matching costs of quarter-pixel candidates with the processing engines inside the light gray box on the right side. Then, the 25 matching costs are compared to find the best MV.

### C. Architecture

The proposed FME architecture is shown in Fig. 9. This architecture is based on the previous work [16], with additional PEs in the light gray box. At first, the reference pixels are read from search window memory and input to the "6-tap interpolation filter" to generate the half-pixel reference data. The reference and the current MB data are input to "4×4 block PE" to compute the transformed residues. The absolute values of the transformed residues are then accumulated with the MV costs to generate the matching costs of one integer-pixel and eight half-pixel candidates. On the other side, the transformed residues of half-pixel candidates are reused and input to "Bilinear Interpolation Filter" to generate the approximate transformed residues and the matching costs of 16 quarter-pixel candidates with (5). The final 25 matching costs are compared to find the best fractional MV. After comparing matching costs of different inter-modes, the best inter-mode can be decided. With the proposed architecture and one-pass algorithm, a large amount of data access power can be saved. In addition, the throughput of the FME engine is doubled compared to the previous architecture with the conventional two-step algorithm.

## V. Parameterized Power-Scalable Encoding System

In addition to optimization for low power, we also want to provide power-aware functionality in our encoder. At first, a pre-skip algorithm [18] is adopted to conditionally save the whole ME computation. Then, power-scalable parameters are introduced to IME, FME, intra prediction (IP), and DeBlocking (DB) engines. These parameters can flexibly control the power consumption of the whole encoding system.

### A. Pre-Skip Algorithm

In a video frame, many MBs are usually located at the still background region. In addition, motion of MBs inside a large moving object can be predicted well with the MVPs. Therefore, we adopt a pre-skip algorithm [18] to compute the SAD values at the origin (0, 0) and MVP before the process of

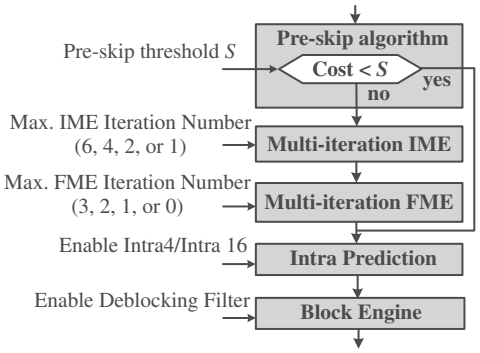


Fig. 10. Illustration of the parameterized power-scalable H.264/AVC encoder. "IME Iteration Number" is defined as the number of initial candidates for the proposed parallel-VBS-FSS algorithm. "FME Iteration Number" is defined as the number of refined inter-modes for the proposed advanced mode predecision algorithm. More than 128 configurations can be provided in this system.

IME. If the precalculated SAD costs of a certain MB are lower than a predefined threshold $S$, the IME and FME computation is skipped because (0, 0) or MVP is good enough to predict the motion of that MB. $S$ can be adaptively adjusted according to the video contents to achieve content-awareness and power-scalability. In our design, $S$ is a fixed input parameter.

### B. Parameterized Encoding System

The proposed parameterized power-scalable encoding system is shown in Fig. 10. For each MB, the pre-skip algorithm is first applied to evaluate the motion activity. If the skipped criterion is passed, the computation of IME and FME is saved. On the contrary, multi-iteration IME and FME are required to find the more precise MVs. There is one parameter that controls the maximal number of initial candidates for IME (also called the maximal IME iteration number). The IME engine processes the initial candidates in an order defined in Fig. 4 and terminate early once the maximal iteration number is achieved. There is another parameter that controls the number of refined modes ($N$) in the proposed advanced mode predecision algorithm for FME (also called the maximal FME iteration number). A larger iteration number makes better coding performance but higher power consumption, and vice versa. If smaller iteration numbers are chosen for IME and FME, most ME computation is saved. Under this condition, IP and DB will become the main power sources. We provide two parameters to turn on or turn off all the Intra $4\times4$ (I4) modes and Intra $16\times16$ (I16) modes, respectively. In addition, we still have one parameter to enable/disable DB engine. Finally, there are more than four (IME) $\times$ 4 (FME) $\times$ 2 (I4) $\times$ 2 (I16) $\times$ 2 (DB) = 128 power modes if the pre-skip threshold $S$ is also taken into consideration. Here, we just provide a parameterized encoding system. How to adjust the parameters to achieve best coding performance is beyond the scope of this design and will be the subject of future work.

## VI. Flexible System Architecture

In previous sections, we showed the module design and the power-scalable algorithms. In this section, we will show the proposed system architecture which not only can improve hardware efficiency in terms of area and power but also flexibility for realization of a power-aware system.

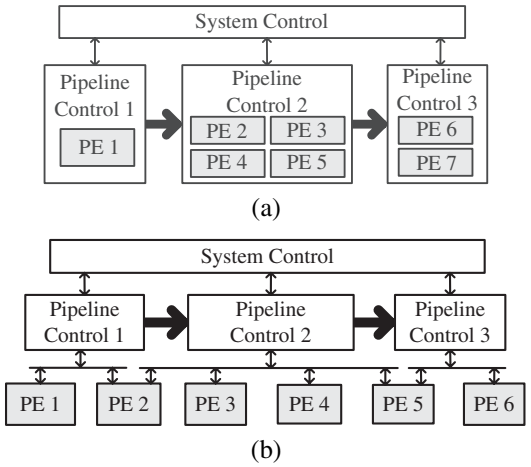


Fig. 11. Illustration of two types of control systems. (a) Conventional two-layer control system. Processing engines can be operated in only one pipeline stage and (b) proposed three-layer control system. Processing engines are flexibly operated in different pipeline stages.

### A. Conventional Two-Layer Control System

The conventional two-layer control system adopted in [8] is shown in Fig. 11(a). MB pipelining is a common technique for video encoding systems to improve the system throughput. In this example, there are three stages of MB pipeline. The PEs are tightly coupled with the pipeline stage controls, and each PE can be operated only in one pipeline stage. For example, "PE3" can only be used by "Pipeline Control 2." The limited system flexibility leads to restricted algorithmic development. In addition, power efficiency is also not good enough because it is hard to gate the clock sources of PEs separately. The power manager can only gate the clock in the unit of one pipeline stage. This is defined as coarse-grained clock gating in this paper. For example, we can shut down the clock of the second pipeline stage in Fig. 11(a) only when all four PEs, from PE2 to PE5, are idle.

### B. Proposed Three-Layer Control System

To solve the problems mentioned above, a new system hierarchy is adopted in our design. The proposed three-layer control system is shown in Fig. 11(b). As we can see, PEs are loosely coupled with pipeline controls. Each PE can be flexibly operated in more than one pipeline stage. Therefore, algorithmic development becomes more flexible. A pipeline control has to send a "request" signal to the PE for the sake of data processing. If the PE is originally idle, it will be activated and respond to an "acknowledge" signal to the pipeline control. After the task is finished, PE will feedback a "finish" signal, enter the idle state, and wait for the next request. Under this architecture, PEs can be shared between different pipeline stages and thus hardware utilization is improved. In addition, power efficiency also becomes better because the

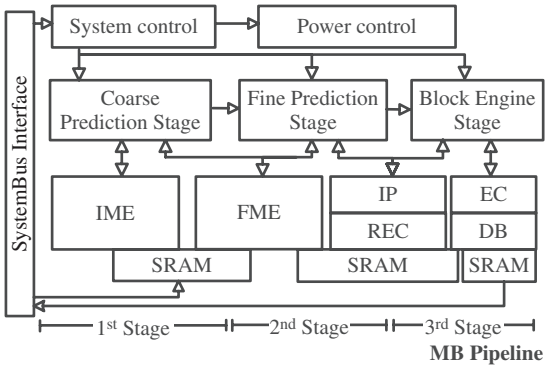


Fig. 12. Proposed system architecture for the H.264/AVC encoder. The three-layer control system is adopted to improve system flexibility, including the system control, the pipeline control, and the PE control. There are three MB-pipeline stags comprising "Coarse Prediction," "Fine Prediction," and "Block Engine" stages. "Power control" is used to online monitor the states of processing engines and stage controls. Once they enter the idle state, the clock sources will be gated for power saving.

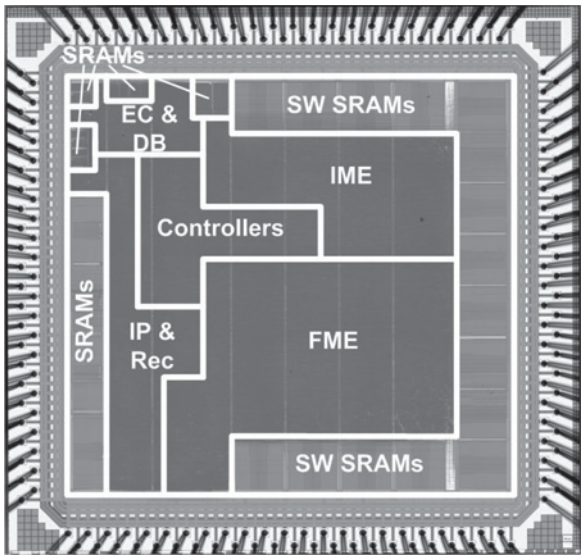


Fig. 13. Chip photo of the proposed H.264/AVC encoder.

TABLE I
LIST OF CHIP SPECIFICATIONS OF THE PROPOSED H.264/AVC ENCODER

| | |
|---|---|
| Technology | TSMC 0.18$\mu$m 1P6M CMOS |
| Pad/Core Voltage | 3.3/1.8-V |
| Core Area | 3.47×3.70 mm$^2$ |
| Logic Gates | 452.8 K (2-input NAND Gate) |
| SRAM | 16.95 KB |
| Max. Reference Frame | 1 for D1, 2 for CIF/QCIF |
| Max. Search Range | H[−32, 31], V[−16, 15] |
| Max. Operating Frequency | 54 MHz |
| Power Consumption | 67.2–43.5 mW for D1, 54MHz, 1.8-V |

clock source of each PE and each pipeline control can be gated separately. This is defined as fine-grained clock gating in this paper.

The proposed system architecture for our H.264/AVC encoder is shown in Fig. 12. From top to bottom, there are three layers of controls—the system control, the pipeline control, and the PE control. From left to right, there are three pipeline stages for improvement of the through. The first is "Coarse Prediction Stage" which implements the pre-skip algorithm and IME. The second one is "Fine Prediction Stage" which implements the FME, and IP/Reconstruction (REC) for luminance pixels. The last one is "Block Engine Stage" which implements IP/REC for chrominance pixels, entropy coding (EC), and DB.

Because the matching cost of MVP with fractional precision should be computed for the pre-skip algorithm, FME engine is required to be operated in the first pipeline stage. However, "Fine Prediction Stage" also needs FME engine for the refinement of MVs. In the conventional two-layer control system, hardware sharing between two pipeline stages is impossible. As a result, two FME engines are inevitable. On the contrary, hardware sharing is possible under the proposed flexible system architecture and thus the hardware cost is reduced.

Retiming is a technique used to balance critical paths between pipeline registers. With this technique, higher operating frequency can be achieved, and then the hardware throughput is improved. The same concept can also be used in MB pipeline design of a video encoding system. As can be seen in Fig. 12, IP and REC engines are operated in two pipeline stages on the strength of our flexible system architecture. The processes of luminance and chrominance data are respectively distributed to the second and the third stages to achieve retiming of MB pipeline. The second pipeline stage is the throughput bottleneck of our encoding system. Without the capability of MB pipeline retiming, there may be three other solutions. The first solution is that IP and REC engines process both luminance and chrominance data in the second pipeline stage. It makes the processing cycles in the second stage longer and degrades the throughput. The second solution is that IP and REC engines process both luminance and chrominance data in the third pipeline stage. This solution requires an additional memory to store luminance current and prediction data between the second and the third pipeline stages. In addition, it may also make the third pipeline stage become the throughput bottleneck. The third solution is to use two IP and two REC engines in different pipeline stages, but the hardware cost is doubled. No matter what solutions are chosen, hardware efficiency is degraded.

Finally, power efficiency can also be improved under the proposed system architecture because of the support of fine-grained clock gating. "Power Control" automatically monitors the states of each PE and pipeline control to separately shut down the clock sources once they enter the idle states. As a result, around 20% power of the whole encoding system can be saved by fine-grained clock gating according to our gate-level simulation.

## VII. IMPLEMENTATION RESULTS

We have implemented the proposed H.264/AVC encoder with TSMC 0.18 $\mu$m CMOS technology. The chip photo is shown in Fig. 13. The maximal encoding capability is D1 ($720 \times 480$) 30 frames/s video encoding with 67.2 to 43.5 mW power consumption. The required hardware resources are

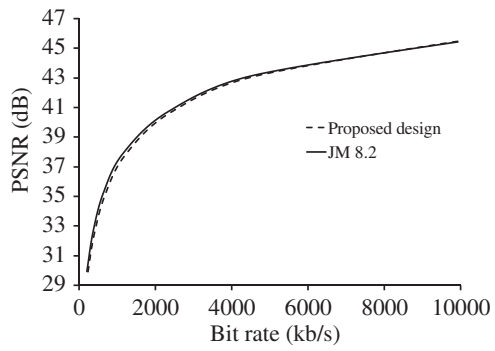


Fig. 14. Rate-distortion curve of the proposed H.264/AVC encoder for D1 (720 × 480) 30 frames/s *Soccer* sequence. Only 1 reference frame is used in this simulation.

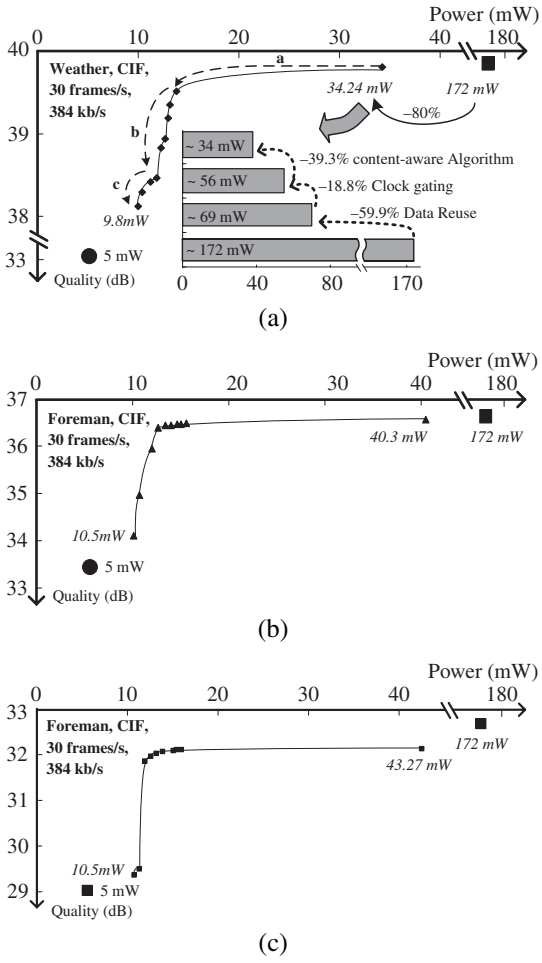


Fig. 15. Power–performance curves of the proposed H.264/AVC encoder for (a) *Weather*, (b) *Foreman*, and (c) *Stefan* sequences. The black square on the up-right side is the reference point of Huang's H.264/AVC encoder [8]. The black circle on the bottom-left side is the reference point of Lin's MPEG-4 encoder [24]. The curve "a" shows the power scalability from two reference frames to 1 reference frame. The curve "b" shows the power scalability of multi-iteration IME and FME. The curve "c" shows the power scalability of IP and DB.

TABLE II
POWER COMPARISON OF BASELINE PROFILE H.264/AVC ENCODING FOR D1 VIDEOS

| | Chang's [11] | Lin's [12] | This Paper |
|---|---|---|---|
| Technology | TSMC 0.13 $\mu$m | UMC 0.13 $\mu$m | TSMC 0.18 $\mu$m |
| Power | 163–27 mW | 23.61 mW | 67.2–43.5 mW |

452.8 K logic gates and 16.95 KB SRAMs. Please see Table I for the detailed chip specifications.

The rate-distortion curve of our design for the D1 (720×480) 30 frames/s *Soccer* sequence is shown in Fig. 14. The curve shows the best coding performance we can provide with the highest power consumption. *Soccer* is a sporty video with high motion activity and considered as the worst case of our design. Compared to the reference software [19], the simulation results of our design show 2.69% bit rate increase and 0.12 dB quality drop in average.

The power–performance curves of the proposed power-aware H.264/AVC encoder are shown in Fig. 15. Two reference points of Huang's H.264/AVC encoder [8] (the black square) and Lin's low-power MPEG-4 encoder [24] (the black circle) are also shown in the figure. The best coding performance of our design (on the up-right side of the figure) is close to Huang's design which adopts the full search algorithm. The comparative 80% power reduction comes from the content-aware strategy at the algorithm level, the clock gating technique at the circuit level, and the data reuse technique at the architecture level. Due to the proposed parameterized power-scalable system, our design can provide more than 128 configurations for different power requirements. Some of the configurations are shown in Fig. 15. Finally, our design can provide an ultra low-power mode close to Lin's design on the bottom-left side.

In Table II, we list the power data of three H.264/AVC encoders under D1 (720 × 480) video encoding. As we can see, the proposed design is comparable to the two state-of-the-art H.264/AVC encoders in terms of power efficiency.

## VIII. CONCLUSION

In this paper, a low-power and power-aware H.264/AVC video encoder has been proposed. The power efficiency was co-optimized at the algorithm, architecture, and circuit levels. At the start, we adopted hardware-oriented algorithm to consider the data reuse issue of ME at the algorithm level. Then, content-aware strategies were utilized to reduce computation and maintain coding performance. Suitable parallel architectures have been presented to achieve good data reuse capability for data access power reduction. Finally, the proposed flexible system architecture improves hardware efficiency in terms of area with MB pipeline retiming and power with fine-grained clock gating. In addition, power-aware functionality is also supported by the proposed parameterized encoding system. As a result, the proposed H.264/AVC encoder can provide competitive power efficiency under D1 (720×480) 30 frames/s video encoding and the best power configurations compared to the previous state-of-the-art designs.

## REFERENCES


[1] T. Mudge, "Power: A first-class architectural design constraint," *IEEE Comput.*, vol. 34, no. 4, pp. 52–58, Apr. 2001.

[2] M. Etoh and T. Yoshimura, "Advances in wireless video delivery," in *Proc. IEEE*, vol. 93, no. 1, pp. 111–122, Jan. 2005.

[3] M. Bhardwaj, R. Min, and A. P. Chandrakasan, "Quantifying and enhancing power awareness of VLSI systems," *IEEE Trans. Very Large Scale Integration (VLSI) Syst.*, vol. 9, no. 6, pp. 757–772, Dec. 2001.

[4] C.-J. Lian, S.-Y. Chien, C.-P. Lin, P.-C. Tseng, and L.-G. Chen, "Power-aware multimedia: Concepts and design perspectives," *IEEE Circuits Syst. Mag.*, vol. 7, no. 2, pp. 26–34, 2007.

[5] D. Linden, *Handbook of Batteries*. 2nd ed. New York: McGraw-Hill, 1995.

[6] *Advanced Video Coding for Generic Audiovisual Services*, Int. Telecommun. Union-Telecommun. (ITU-T) and Int. Standards Org./Int. Electrotech. Comm. (ISO/IEC) JTC 1, Recommendation H.264 and ISO/IEC 14 496-10 (MPEG-4) AVC, Mar. 2003.

[7] T. Wiegand, G. J. Sullivan, G. Bjontegaard, 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.

[8] Y.-W. Huang, T.-C. Chen, C.-H. Tsai, C.-Y. Chen, T.-W. Chen, C.-S. Chen, C.-F. Shen, S.-Y. Ma, T.-C. Wang, B.-Y. Hsieh, H.-C. Fang, and L.-G. Chen, "A 1.3TOPS H.264/AVC single-chip encoder for HDTV applications," in *Proc. IEEE ISSCC Dig. Tech. Papers*, San Francisco, CA, 2005, pp. 128–129.

[9] H. Mizosoe, D. Yoshida, and T. Nakamura, "A single chip H.264/AVC HDTV encoder/decoder/transcoder system LSI," in *Proc. IEEE Int. Conf. Consumer Electron.*, Las Vegas, NV, 2007, pp. 1–2.

[10] Z. Liu, Y. Song, M. Shao, S. Li, L. Li, S. Ishiwata, M. Nakagawa, S. Goto, and T. Ikenaga, "A 1.41w H.264/AVC real-time encoder SoC for HDTV1080P," in *Proc. Dig. Symp. VLSI Circuits*, Kyoto, Japan, 2007, pp. 12–13.

[11] H.-C. Chang, J.-W. Chen, C.-L. Su, Y.-C. Yang, Y. Li, C.-H. Chang, Z.-M. Chen, W.-S. Yang, C.-C. Lin, C.-W. Chen, J.-S. Wang, and J.-I. Guo, "A 7mw-to-183mw dynamic quality-scalable h.264 video encoder chip," in *Proc. IEEE ISSCC Dig. Tech. Papers*, San Francisco, CA, 2007, pp. 280–281.

[12] Y.-K. Lin, D.-W. Li, C.-C. Lin, T.-Y. Kuo, S.-J. Wu, W.-C. Tai, W.-C. Chang, and T.-S. Chang, "A 242mw 10mm$^2$ 1080p H.264/AVC high-profile encoder chip," in *Proc. IEEE ISSCC Dig. Tech. Papers*, San Francisco, CA, 2008, pp. 314–315.

[13] Y.-H. Chen, T.-C. Chen, and L.-G. Chen, "Hardware oriented content-adaptive fast algorithm for variable block-size integer motion estimation in H.264," in *Proc. IEEE Int. Symp. Intell. Signal Process. Commun. Syst.*, Hong Kong, 2005, pp. 341–344.

[14] T.-C. Chen, Y.-H. Chen, S.-F. Tsai, S.-Y. Chien, and L.-G. Chen, "Fast algorithm and architecture design of low-power integer motion estimation for H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 5, pp. 568–577, May 2007.

[15] C.-Y. Chen, S.-Y. Chien, Y.-W. Huang, T.-C. Chen, T.-C. Wang, and L.-G. Chen, "Analysis and architecture design of variable block-size motion estimation for H.264/AVC," *IEEE Trans. Circuits Syst. Part I: Fundamental Theory Applicat.*, vol. 53, no. 3, pp. 578–593, Mar. 2006.

[16] T.-C. Chen, Y.-W. Huang, and L.-G. Chen, "Fully utilized and reusable architecture for fractional motion estimation of H.264/AVC," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Montreal, QC, 2004, pp. 9–12.

[17] T.-C. Chen, Y.-H. Chen, and L.-G. Chen, "Low power and power aware fractional motion estimation of H.264/AVC for mobile application," in *Proc. IEEE Int. Symp. Circuits Syst.*, Island of Kos, Greece, 2006, pp. 5331–5334.

[18] Y.-H. Chen, T.-C. Chen, and L.-G. Chen, "Power-scalable algorithm and reconfigurable macro-block pipeline architecture of H.264 encoder for mobile application," in *Proc. IEEE Int. Conf. Multimedia Expo*, Toronto, ON, 2006, pp. 281–284.

[19] *H.264/AVC Reference Sofware JM8.2* [Online]. Available: http://iphome.hhi.de/suehring/tml/download/

[20] K. Danckaert, K. Masselos, F. Catthoor, H. J. D. Man, and C. Goutis, "Strategy for power-efficient design of parallel systems," *IEEE Trans. Very Large Scale Integration (VLSI) Syst.*, vol. 7, no. 2, pp. 258–265, Jun. 1999.

[21] Y.-W. Huang, T.-C. Wang, B.-Y. Hsieh, and L.-G. Chen, "Hardware architecture design for variable block size motion estimation in MPEG-4 AVC/JVT/ITU-T H.264," in *Proc. IEEE Int. Symp. Circuits Syst.*, Thailand, 2003, pp. 796–799.

[22] S. Y. Yap and J. V. McCanny, "A VLSI architecture for variable block size video motion estimation," *IEEE Trans. Circuits Syst. Part II: Analog Digital Signal Process.*, vol. 51, no. 7, pp. 384–389, Jul. 2004.

[23] H. F. Ates and Y. Altunbasak, "SAD reuse in hierarchical motion estimation for the H.264 encoder," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Philadelphia, PA, 2005, pp. 905–908.

[24] C.-P. Lin, P.-C. Tseng, Y.-T. Chiu, S.-S. Lin, C.-C. Cheng, H.-C. Fang, W.-M. Chao, and L.-G. Chen, "A 5mW MPEG4 SP encoder with 2-D bandwidth-sharing motion estimation for mobile applications," in *Proc. IEEE ISSCC Dig. Tech. Papers*, San Francisco, CA, 2006, pp. 1626–1635.


**Yu-Han Chen** was born in Taipei, Taiwan in 1981. He received the B.S. degree in electrical engineering from the National Taiwan University, Taipei, in 2003, and is currently pursuing the Ph.D. degree at the Graduate Institute of Electronics Engineering.

His research interests include image/video signal processing, motion estimation, algorithm and architecture design of H.264 video coders, and low-power and power-aware video coding systems.

**Tung-Chien Chen** was born in Taipei, Taiwan in 1979. He received the B.S. degree in electrical engineering and the M.S. degree in electronics engineering in 2002 and 2004, respectively, from the National Taiwan University, Taipei, where he is pursuing the Ph.D. degree in electronics engineering.

His major research interests include motion estimation, algorithm and architecture design of MPEG-4 and H.264/AVC video coding, and low-power video coding architectures.

**Chuan-Yung Tsai** was born in Kaohsiung, Taiwan, in 1982. He received the B.S. degree in electrical engineering in 2004, from the National Taiwan University, Taipei, Taiwan, where he is currently pursuing the Ph.D. degree at the Graduate Institute of Electronics Engineering, National Taiwan University.

His research interests include algorithm and architecture design of H.264 video encoder/decoders, low-power video coding systems, and intelligent architecture for video processing.

**Sung-Fang Tsai** was born in Hsinchu, Taiwan in 1983. He received the B.S. degree in electrical and electronics engineering from National Taiwan University, Taipei, Taiwan in 2005, where he is working toward the M.S. degree at the Graduate Institute of Electronics Engineering.

His major research interests include motion estimation and algorithm and architecture design of H.264/AVC video coding standard.

**Liang-Gee Chen** (S'84–M'86–SM'94–F'01) received the B.S., M.S., and Ph.D. degrees in electrical engineering from the National Cheng Kung University, Taiwan in 1979, 1981, and 1986, respectively.

In 1988, he joined the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan. From 1993 to 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 the National Taiwan University. His current research interests include DSP architecture design, video processor design, and video coding systems.

Dr. Chen was the general chairman of the 1999 IEEE Workshop on Signal Processing Systems: Design and Implementation. He serves as an Associate Editor of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE TRANSACTIONS ON VLSI SYSTEMS, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II: ANALOG AND DIGITAL SIGNAL PROCESSING, and PROCEEDINGS OF THE IEEE. He was elected the IEEE Circuits and Systems Distinguished Lecturer for 2001–2002.