We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS BOOK CITATION INDEX INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Chapter 8

Algorithm and VLSI Architecture Design for MPEG-Like High Definition Video Coding-AVS Video Coding from Standard Specification to VLSI Implementation

Haibing Yin

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/52965

1. MPEG-like and AVS Video Coding Standard

In multimedia system, there are several video coding standards such as MPEG-1/2/4 [1]-[3], H.264/AVC [4], VC-1 [5], they are the source coding technology basis for digital multimedia applications. Despite of the emerging HEVC standard [6], H.264/AVC is the most mature video coding standard [4] [9]. China Audio and Video Coding Standard (AVS) is a new standard targeted for video and audio coding [7]. Its video part (AVS-P2) had been formally accepted as the Chinese national standard in 2006 [7]. Similar with MPEG-2, MPEG-4 and H. 264/AVC, AVS-P2 adopts block-based hybrid video coding framework. AVS achieves equivalent coding performance with H.264/AVC. There are different coding tools and features in different standards. However, the crucial technologies they employed are very similar with coincident framework. These similar standards are MPEG-like video standards.

In MPEG-like video encoders, motion estimation (ME) and motion compensation (MC) give a temporal prediction version of the current macroblock (MB). Intra prediction (IP) gives the spatial prediction version. Simultaneously, the predicted MB is coded and followed by transform (DCT), quantization (Q), inverse transform (IDCT), and inverse quantization (IQ). The distorted image is reconstructed with in-loop deblocking (DB) filter. Entropy coding (EC) adopts variable length coding to exploit symbol statistical correlation.

AVS-P2 is also a MPEG-like standard similar with H.264/AVC [4]. Its major coding flow is similar with those of other MPEG-like standards. There are also some differences between AVS and H.264/AVC. There are five luminance and four chrominance intra prediction modes on the basis of 8x8 blocks in AVS. Also, only 16x16, 16x8, 8x16, and 8x8 MB inter partition modes are used in AVS, in which quarter pixel motion compensation with 4-tap frac-

tional pixel interpolation is adopted. Being different from H.264/AVC baseline profile, the Jizhun profile in AVS supports bidirectional prediction in B frames using a novel "symmetric" mode [7]. Combined with forward, backward, symmetric, and direct temporal prediction modes, there are more than fifty MB inter prediction modes. The industrialization for the AVS standard is being on and leaded by the AVS industry alliance. Efficient AVS video encoder design is important for AVS standard industrialization to dig the standard compression potential.

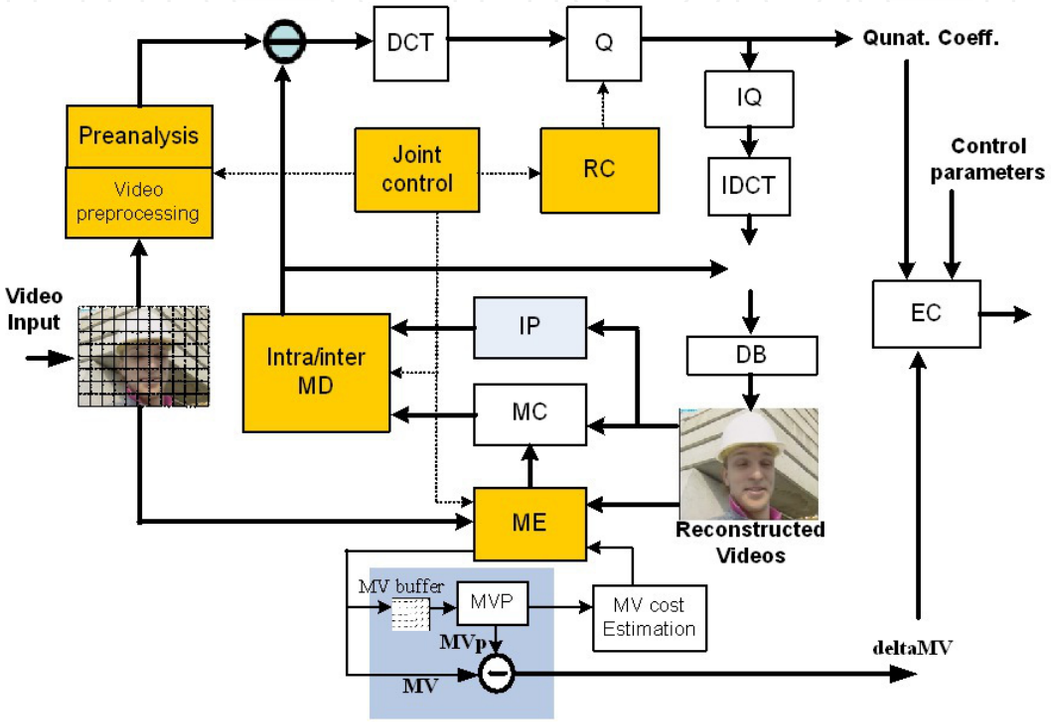

Figure 1. The modules to be jointly optimized in MPEG-like video encoder framework.

With the technology development and video quality requirement increment, consumer demand is being generated for larger picture sizes and more complex video processing [8]-[10]. High definition (HD) video application has become the prevailing trends. A wide range of consumer applications require the ability to handle video resolutions across the range from 720P (1280x720) and full high definition (full-HD, 1920x1080) to quad full high definition (QFHD, 3840x2160) and even Ultra HD (7680x4320) [15]-[22].

HD applications result in higher bit rate and complex video coding [15] [21]. Achieving higher video compression is one important task for video coding expert group and related corporations, especially for HD and super HD applications. Efficient HD MPEG-like video encoder implementation is a huge challenge.

H.264/AVC and AVS standards offer the potential for high compression efficiency. However, it is very crucial to design and optimize video encoder to fully dig and explore the compression potential, especially for the HDTV applications. In this chapter, we discuss the

http://dx.doi.org/10.5772/52965

design considerations for HD video encoder architecture design, focusing on algorithm and architecture design for crucial modules, including integer and fractional pixel motion estimation, mode decision, and the modules suffering from data dependency, such as intra prediction and motion vector prediction.

2. High Definition Video Encoder Hardware Implementation

2.1. VLSI Implementation

AVS and H.264/AVC video encoders may be implemented on platforms such as general CPU or DSP processor, multi-core processor, or hardware platforms such as FPGA (Field Programmable Gate Array) and ASIC (Application Specific Integrated Circuit). For efficient HD video encoder, FPGA and ASIC are well-suited platforms for VLSI implementation. These platforms offer huge hardware computation (macrocells or hardware gate) and on-chip storage (SRAM) resources, which are both important and indispensible for professional HD MPEG-like video encoder implementation.

The hardware architectures for MPEG-4 video encoders were reviewed in [11]. Also, there are several only intra-frame encoder architectures reported in [12]-[14]. The predominating VLSI architectures for HD H.264/AVC encoder architectures were reported in the literature. However, algorithm and architecture further optimization is still important and urgent.

2.2. Design Challenges

There are several challenges as for HD video encoder architecture design, including ultra high complexity and throughput, high external memory bandwidth and on-chip SRAM consumption, hardware data dependency, and complex prediction structures. Moreover, multiple target performance trade-off should be taken into consideration.

The first challenge is complexity and throughput. H.264 and AVS requires much higher computation complexity than the previous standards, especially for HDTV applications. There are some coding tools that contribute to performance improvement, however resulting in high computation complexity, such as complex temporal prediction with multiple reference frame (MRF), fractional motion vector (MV) accuracy, and variable block size motion estimation (VBSME), intra prediction with multiple prediction modes, Lagrangian mode decision (MD), and context-based adaptive binary arithmetic coding (CABAC). As a result, the processing throughput is dramatically high. Taking 1080P@30Hz as an example, there are 8160 macroblocks (MB) in one frame, and the MB pipelining throughput is 244800 MBs per second. In QFHD@ 30fps format, the throughout is as four time as that in 1080P@30fps. In the-state-of-the-art architectures [15]-[21], the average MB pipeline interval generally varies from 100 to 500 cycles. Under this constraint, the architecture designs, for IME with large search range and FME with multiple modes, are both huge challenges.

The second challenge is the processing sequential flow and data dependency. There are frame, MB, and block level data dependencies. The frame-level dependencies due to I, P,

and B frames contribute the considerable system bandwidth. The MB-level sequential flows include intra/inter prediction, MB reconstruction (REC), EC, and DB filter. At the block level, one block intra prediction (IP) is context-dependent with the up, left, and up right blocks. In the reconstruction loop, DCT, Q, IQ, and IDCT are processed in turn. The motion vector prediction (MVP) is context-dependent with the up, left, and up right blocks. These hierarchical data dependencies are harmful for hardware pipelining. It is important to efficiently map the sequential algorithms to parallel and pipelined hardware architectures to improve the hardware utilization and the throughput capacity.

Third, high SRAM consumption and external memory bandwidth are major challenges. Local SRAM buffers are necessary to achieve data communication among adjacent pipeline stages in pipelined architecture. Reference pixel SRAM buffers for IME and FME are the largest buffer due to the large size search window. Buffer structure and data organization are highly related with hardware architecture. As a result, on-chip buffer structure and data organization are important consideration factors for hardware architecture design.

External memory bandwidth is another challenge. There are huge data exchanges between external memory and on-chip SRAM buffer for real-time video coding. The reference pixel access operations are the largest bandwidth consumers with almost 80% consumption. MRF motion estimation directly doubles the bandwidth consumption and aggravates the bandwidth burden greatly.

Fourth, multiple target performance optimization is another challenge. In terms of hardware architecture efficiency, there are multiple target parameters concerned. Typical target performance parameters are R-D performance, hardware cost, on-chip SRAM consumption, processing throughput, external memory bandwidth, and system power dissipation, etc.

How to achieve trade off is critical for architecture design. Multiple target performance parameters are all factors to be considered for architecture design. It is very difficult to satisfy all these constraints and reach optimal trade-off. It is very necessary to make in-depth research at algorithm and architecture level optimization to tradeoff multiple mutually exclusive factors.

2.3. Algorithm and Architecture Joint Design

As analyzed above, HD video encoder architecture design is a multiple target optimization problem, and challenged by several factors. Among these multiple target parameters, on-chip SRAM size and external memory bandwidth are very crucial. These two targets have important influences on data organization and on-chip buffer structure [23] [24]. Fig.1 gives the inter-relationship among algorithm, architecture, data organization, and buffer structure.

The hardware oriented algorithm is customized under the hardware architecture constraint, with data organization and data flow considered. Hardware architecture is designed with the buffer structure considered according to the algorithm characteristics constraint. Data organization and on-chip buffer structure are jointly designed to achieve efficient data reuse and regular control flow for massive data streaming. On the one hand, efficient data reuse alleviates high memory bandwidth burden and decrease the SRAM consumption. On the

http://dx.doi.org/10.5772/52965

other hand, regular control flow simplifies the architecture RTL (Register-Transfer-Level) code implementation.

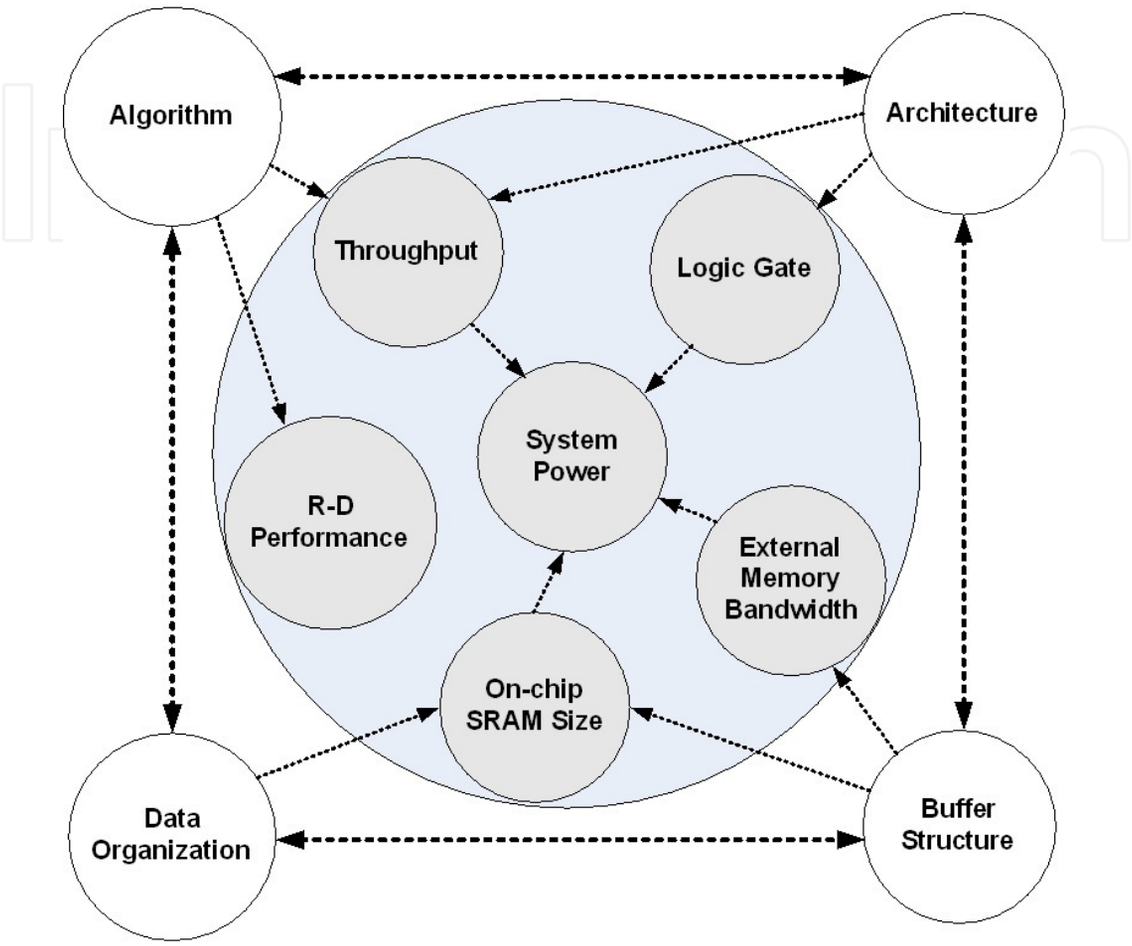

Figure 2. Algorithm, architecture, data organization, and buffer structure

The multiple target performance parameters are complex. The hardware oriented algorithm directly determines the R-D performance. The hardware architecture determines the logic gate consumption. The buffer structure determines the external memory bandwidth consumption, and determines the on-chip SRAM consumption jointly with the data organization. The system power dissipation is more complicate, and determined by the throughput capacity, logic gate, memory bandwidth, and SRAM size. Also, it is directly and indirectly related with algorithm, architecture, data organization, and buffer structure.

According to above analysis, algorithm and architecture should be jointly designed under the multiple target performance trade off consideration. Data organization and on-chip buffer structure are highly related with algorithm and architecture. They should be focused on during the mapping process from algorithm to architecture.

2.4. Multiple Target Parameter Optimization

In order to achieve multiple target performance optimization, it is necessary to explore the inter-function mechanism among the multiple targets. Also, how to make exact and fair comparison among multiple target performance parameters is a basic but important problem. It is difficult to build appropriate multiple target performance evaluation models to guide algorithm and architecture joint design. The following factors jointly contribute to this dilemma.

First, different profile and level combination, as well as the video specification are targeted in prevailing architectures [12]-[21]. There are different advanced coding tools in different profiles. As a result, it is not easy to evaluate the multiple target performance of the architectures in different profile and level combinations.

Second, there is complex inter-relation among multiple target performance parameters. Logic gate and on-chip SRAM consumption are mutually interdependent, and highly related with the throughput and architecture. System power dissipation is related with logic gate, SRAM, and system clock frequency (throughput). The external memory bandwidth is related with the system throughput and on-chip SRAM. These target performance parameters are all inter-dependent, and it is not easy to accurately measure the inter-influence mechanism for multiple target performance evaluation.

Third, R-D performance fair comparison is very difficult. On the one hand, R-D performance results reported in the architectures [15]-[21] are derived with different benchmark algorithms. On the other hand, different test sequences are used for R-D performance simulation. Even the same PSNR results reported may correspond to different algorithm performance. PSNR is not the most suitable criterion for accurate video quality assessment.

Fourth, different architectures target for different applications. Some works focus on professional high-end video applications, such as digital TV and broadcasting, in which the compression efficiency is the first target with the highest priority. Some works focus on portable applications, in which power dissipation is the first important target. Different target performance parameters cherish different priority levels in different application targets. This factor is preferred to be considered for multiple target performance evaluation.

The above multiple target performance parameters, with different applications, different profile and levels, are the design constraints for multiple target performance optimization.

3. Hardware Oriented Algorithm Customization

3.1. Multiple Module Joint Algorithm Optimization

In AVS and H.264/AVC video encoder, there are several normative modules whose algorithms are deterministic and not allowed for customization. They are transformation (DCT), quantization (Q), inverse quantization (IQ), inverse transformation (IDCT), intra prediction (IP), motion vector prediction (MVP), motion compensation (MC), deblocking (DB) filter,

http://dx.doi.org/10.5772/52965

and entropy coding (EC). Among them, DCT, Q, IQ, and IDCT jointly form the reconstruction (REC) loop. There are other four modules whose algorithms are customizable according to the application targets. They are video preprocessing or video preanalysis, motion estimation (ME), mode decision (MD), and rate control (RC). Fig.1 gives the video coding framework with these two types of module partition.

The modules with customizable algorithms are very important for architecture design. In VLSI architectures, the REC and IP are usually combined and embedded with the MD module to break the block level data dependency. The MC module is usually combined and embedded with the FME module to reuse the interpolation hardware circuit. As a result, we mainly focus on the four critical modules: IME, FME with MC, MD with IP and REC, and MVP for data dependency in this work. The architectures of the DB and EC modules also influence the throughput, hardware efficiency and power dissipation. Nevertheless, we mainly focus on the modules with customizable algorithm.

3.2. Integer and Fractional Motion Estimation

Motion estimation (ME), including integer-pel ME (IME) and fractional-pel ME (FME), is the most complex module in MPEG-like video encoder. HD ME implementation is highly challenged due to not only large search window, but also new tools such as variable block size ME (VBSME) and multiple reference frames. Moreover, data dependency in block level motion vector prediction (MVP) should be considered for rate distortion optimized ME. Thus, hardware friendly ME algorithm modifications are very important [25] [26]. MVP is combined with the IME and FME modules for algorithm and architecture design.

3.2.1. IME Algorithm Analysis

Full search algorithm is usually adopted due to its good quality and high regularity [25] [26], and it is well-suited for hardware implementation. However, it is challenged due to large search range. Hardware friendly ME algorithm customization is necessary for co-optimization [25]. Fast algorithms can be classified into several categories [15]-[21]: predictive search, hierarchical search, and reduction in search positions and algorithm simplification techniques.

The first category is the predictive ME algorithm. If a predictive MV is estimated using MV field correlation, local search can be employed instead of global search. These types of algorithms achieve small SRAM and logic consumption with high throughput. Predictive ME algorithms achieve almost no performance loss in the sequences with smooth motion. However, R-D performance loss is unavoidable in the sequences with complex motion due to MV prediction malfunction.

Hierarchical multi-resolution ME algorithm is efficient for HD video coding with large search window [18] [24]. Its idea is to predict an initial MV estimate at the coarse level images and refine the estimate at the fine level images. These algorithms are well-suited for hardware implementation due to control regularity and good performance. Relatively, hierarchical search algorithms achieve better tradeoff between R-D performance and hardware cost.

Under the assumption that the matching cost monotonically increases as the search position moves away from the one with minimum cost, convergence to the optimal position still can be achieved without matching all candidates. Consequently, computation is reduced by decimation of search positions. The type algorithms are well-suited for software based video encoder with tradeoff between computation and performance. However, they are ill-suited for hardware implementation due to high irregularity. Also, this method usually traps in local minima resulting in performance degradation due to frequent failure of monotonically distribution assumption in sequences with complex motion.

Simplification techniques are proposed and combined with IME algorithms, especially for full search, to alleviate the high complexity in HD cases. Typical methods include simplification of matching criterion and bit-width reduction [15].

3.2.2. FME Algorithm Analysis

FME contribute to the coding performance improvement remarkably, but the computation consumption is dramatically high. The optimal integer pixel motion vectors (MV) of different size blocks are determined at the IME stage by SAD reuse. At the FME stage, half and quarter pixel MV refinements are performed centered about these integer pixel MVs sequentially.

Although the factional candidate motion vectors are no more than 49 points, FME complexity is very high due to complex interpolation calculation and VBSME support. The FME algorithm is customizable. Typical hardware oriented FME algorithms include five categories [15]-[21]: candidate reduction, search order, criterion modification, interpolation simplification, and partition mode reduction.

First, shrinking the search range is efficient to reduce the search candidates. There are 49 candidates, comprising of one integer-pixel, eight half-pixel, and 40 quarter-pixel candidates. As shown in Fig.3, 49 candidates may be reduced to 17 candidates, 25 candidates, and 9 candidates, and 6 candidates respectively.

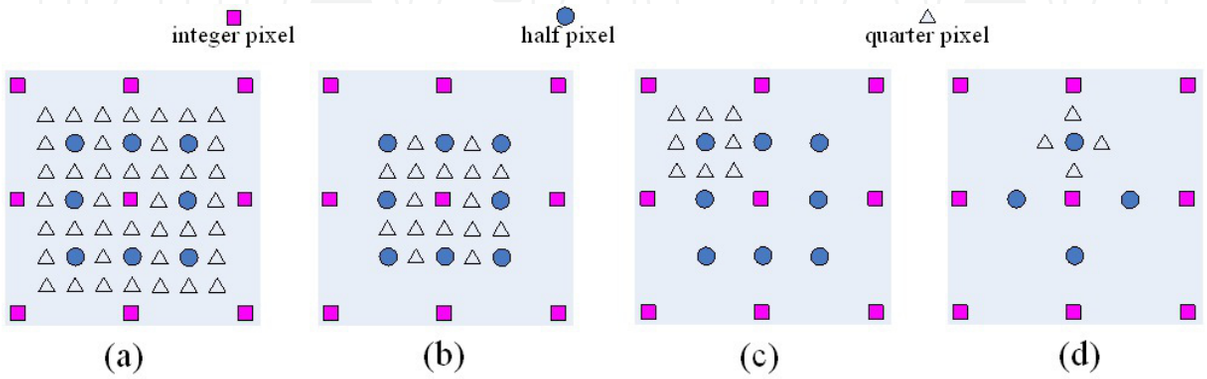

Figure 3. Candidate MVs in different FME algorithms.

http://dx.doi.org/10.5772/52965

Second, search order is important in FME due to data flow design consideration in fraction pixel interpolation. Single-iteration and two-iteration search order are two typical techniques. Full search is usually in single-iteration within 49 or 25 candidates as shown in (a) and (b) in Fig.3, with high data reuse efficiency. Two-iteration is usually employed to search optimal half-pixel MV at the first iteration stage, as shown in (c) and (d) in Fig.3, then quarter-pixels are refined at the second iteration stage. Relatively, data reuse efficiency in two-iteration is lower than that in single-iteration method.

Third, matching criterion modification is employed in some works. SATD + λR_{MV} is the typical criterion, and Hadamard transformation is used to calculate the SATD from inter-prediction residue. R_{MV} is the coding bit cost of the MV residue. λ is the Lagrangian multiplier for rate distortion optimized motion estimation.

Fourth, interpolation simplification is employed in some works to alleviate the computation burden. Six-tap and two-tap interpolation filters are used for standard half and quarter pixel interpolations. The interpolation used for fraction pixel MV motion search is allowed for simplification at the cost of R-D performance degradation.

Fifth, variabe block size (VBS) partition preselection technology is usually combined with FME algorithm to alleviate the data processing burden accounting for multiple partition modes. In HD video encoders, block partition preselection is acceptable. In some works, only blocks larger than 8x8 are supported to alleviate the throughput burden [15] [26]. Some heuristic measures are employed to exclude some partition modes [16] [20] [21].

3.2.3. The Proposed IME Algorithm

Accounting for the design challenges, they are two types of IME architectures. The first type of architectures are based on zero motion vector (MV) centered search algorithm [15] [17]-[19]. All reference pixels in the search window are buffered in on-chip buffer with large size SRAM consumption. The other types of architectures are based on predefined MV centered local search algorithm [16] [21] [27]. In these works, local search is performed within local search window centered about the predefined center MV (MVc), for example a predicted MV (MVp). As a result, only partial reference pixels are buffered into on-chip SRAM buffer. These architectures achieve small SRAM consumption and fast search speed, however suffering from search accuracy degradation due to inefficient MVc estimation.

The center MV based local search algorithm is the most suitable solution for HD and ultra HD applications. It is crucial to improve the MVc accuracy to sustain this type algorithm's advantage. Traditional MV prediction algorithms utilize the motion filed correlation to estimate the center MV. It is efficient for the sequences with regular motion. However, they may malfunction in sequences with complex motion.

According to the predominating IME architectures [15]-[22], multi-resolution algorithms are well-suited for HD encoder implementation with good tradeoff between performance and complexity. In this work, we tends to employ multi-resolution search algorithm to search an appropriate candidate center MV to compensate the malfunction due to conventional MV prediction algorithm. Multiple center MV selection is employed to estimate the MVc.

The proposed predictive based motion estimaiton algorithm is shown in Fig.4. Multi-resolution coarse presearch is employed to presearch a candidate center MV (MV_p). Spatial and temporal domain MV median predictions are employed to determine two predictive center candidates (MV_s and MV_t). The MV of skip mode, MV_{skip}, is also taken as one candidate. As a result, these four candidate center MVs are selected to estimate the center MVc. This measure is adopted to improve the MVc prediction accuracy.

The proposed multi-resolution algorithm is performed from the coarsest level L_2 (16:1 downsampled), and the middle level L_1 (4:1 down-sampled), to the finest level L_0 (undown-sampled) sequentially. The 256 pixels in one MB (at level L_0) are shown in Fig.5-(a). They are 4:1 down-sampled to four 8x8 blocks (level L_1) indexed by m and n, respectively marked using different symbols:×(mn=00),●(mn=01),▲(mn=10), and ■(mn=11). Similarly, each 8x8 block at level L_1 is 4:1 down-sampled to four 4x4 subblocks (level L_2) indexed by p and q, respectively marked using red, blue, green and black colors. The three-level down-sampling and the indices (m~q) are shown from (a) to (c) in Fig.5. Similarly, all reference pixels are also down-sampled into sixteen interlaced reference sub-search windows.

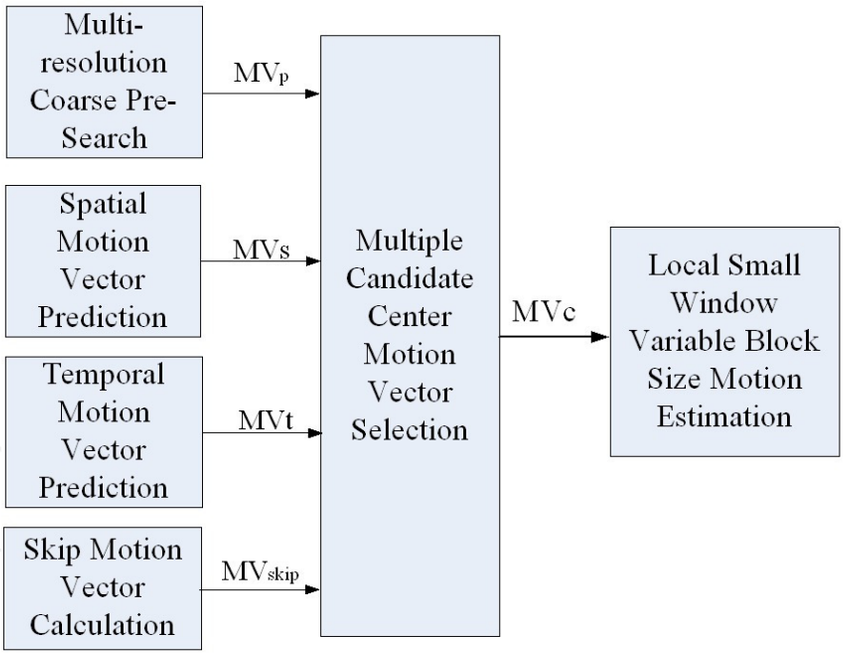

Figure 4. The proposed multiple candidate multi-resolution IME algorithm.

The control flows of the proposed IME algorithm are illustrated in Fig.6. Suppose the whole integer pixel search window is $\pm SR_X \times \pm SR_Y$. Here ±32 × ±32 is used as an illustration example accounting for page limitation. Motion vector refinement is processed by three successive hierarchical stages as follows.

http://dx.doi.org/10.5772/52965

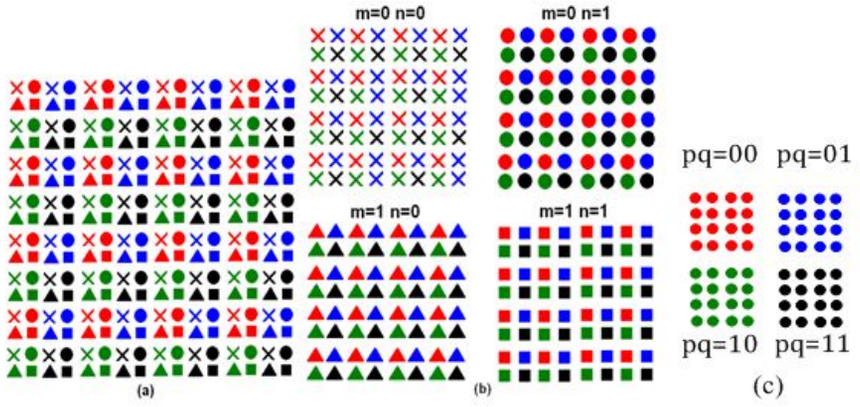

Figure 5. Pixel organization and illustration for multi-resolution IME algorithm.

First, full search is done at level L_2 to check all downsampled candidate motion vectors (black points) in the whole search window shown in Fig.6-(a). To accelerate search speed, four-way parallel searches are employed using four downsampled pixel samples (mn=01). As shown in Fig.6-(a), there are four-way parallel matching operations issued. Each way searches four horizontally adjacent candidates by this four-way parallelism. As a result, the proposed algorithm achieves the throughput of sixteen candidates at each cycle at level L_2.

Second, motion vector refinement at level L_1 is shown in Fig.6-(b). Only the pixels marked with ●(mn=01) attend in SAD calculation for L_1 level refinement. Four refinement centers are shown with four red circles as shown in Fig.6-(a). Four-way local searches at level L_1 are simultaneously performed within local search window $\pm SR_{XL1} \times \pm SR_{YL1}$ centered about four center candidates respectively. One optimal MV (MVp) is finally selected. Then, the final center MVc is estimated from MV_p, MV_s, MV_t, and MV_{skip} using median estimation.

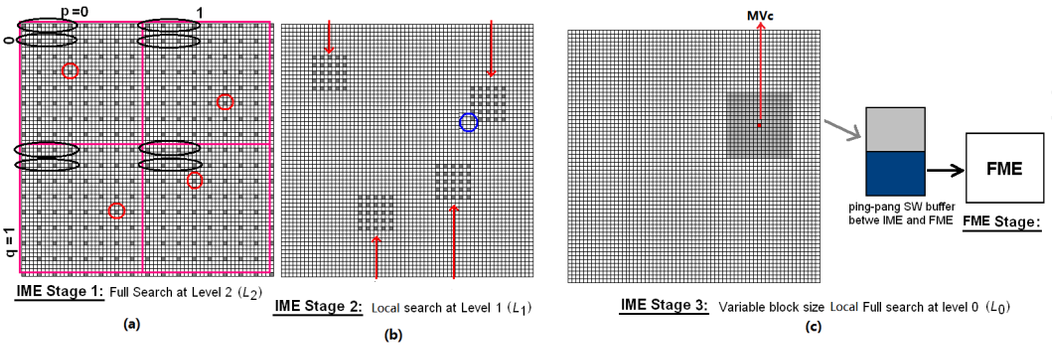

Figure 6. Structure of the proposed multi-resolution IME algorithm.

Third, variable block size IME is performed at stage 3 at level L_0 only within local search window with size of $\pm SR_{XL0} \times \pm SR_{YL0}$. The resulting R-D performance degradation is small due to the high MV correlation existing in different size blocks of one MB if the local search window is large enough [21].

If the system throughput is not enough for real-time coding, for example in QFHD format, N-way hardware parallelism may be employed at L_0 level for variable block size IME refinement. N is an integer and determined according to the system throughput. The search window size parameters are as follows: SR_X=128, SR_Y=96, SR_{XL1}=10, SR_{YL1} =8, and SR_{XL0}=16, SR_{YL0} =12. These parameters are all customizable according to the application targets and video specification.

Corresponding to the algorithm modification, the MB level pipeline structure should be modified. To improve the throughput efficiency for HD video coding, we deepen the pipeline structure and separate the conventional one-stage IME into three pipeline stages: integer pixel presearch, local search window reference pixel fetch, and local integer pixel motion estimation. The system level pipeline structure will be given in the forthcoming section (system pipeline structure).

The reference pixels are buffered twice, during the presearch stage and the local integer pixel motion estimation stage. At the presearch stage, only quarter-downsampled reference pixels are buffered into on-chip buffer. At the local integer pixel stage, only the reference pixels in the local small search window centered about MVc are buffered into on-chip buffer.

3.2.4. The Proposed FME Algorithm

FME contributes to the coding performance improvement remarkably, but the computation consumption is also very high. The optimal integer pixel MVs of VBS blocks are determined at the IME stage by SAD reuse. At the FME stage, half and quarter pixel MV refinements are performed centered about these integer pixel MVs sequentially. We adopt two-iteration FME algorithm framework as shown in Fig.3-(c).

On-chip SRAM consumption for the reference pixels in HD video encoder is dramatically high. To decrease the on-chip SRAM consumption for reference pixels buffering, we propose an efficient buffer share mechanism between IME and FME with algorithm simplification. Only the local search window centered about MVc are buffered in ping-pong structed buffer for IME and FME data reuse.

There are strong correlations existing in the MVs of different size blocks in the same MB [26]. As a result, there exists a local search window (LSW) which contains almost all displaced blocks needed in the whole window case for FME refinement. Thus, FME only needs to be performed within this LSW.

Another important problem in hardware oriented FME algorithm is the huge bidirectional interpolation consumption burden. In AVS Jizhun profile, a novel bidirectional prediction, “symmetric” mode, is adopted. In this mode, only forward MV (mvFw_Sym) is coded in syntax stream, while backward MV (mvBw_Sym) is predicted from mvFw_Sym by

$$\text{mvBw\_Sym}=-\left(\frac{\text{mvFw\_Sym}\times\text{BlockDistanceBw}}{\text{BlockDistanceFw}}\right) \quad (1)$$

http://dx.doi.org/10.5772/52965

Here, BlockDistanceFw and BlockDistanceBw are the temporal distances between the current block and its forward and backward reference frames. mvBw_Sym and mvFw_Sym are all quarter pixel MV. To obtain the fractional pixel displaced block, we need to perform half and quarter pixel interpolation successively. If symmetric mode is adopted in both IME and FME, the interpolation computation cost will be very high, and the normal FME pipeline rhythm is also disturbed. So, simplification for symmetric mode FME is necessary.

At the IME stage, although the mvFw_Sym is integer pixel accuracy, its corresponding mvBw_Sym is quarter pixel accuracy. Some cycles are desired to finish the quarter pixel interpolation, so this extra cycle consumption will lower the throughput efficiency of the IME module. Thus, symmetric mode is not supported in IME in this work.

Symmetric mode FME refinement is followed after forward and backward individual FME refinements. mvFw_Sym is initialized as the quarter pixel accuracy MV (mvFw_normal) of normal forward FME to calculate the corresponding backward MV in the symmetric mode. There are eight half-pixel and eight time quarter-pixel candidate MVs to be refined in FME. As a result, only eight times half-pixel and quarter-pixel interpolation are needed respectively for forward reference MB, and totally sixteen times half-pixel and quarter-pixel interpolation are needed respectively for the backward displaced blocks. This extra interpolation computation is acceptable and has no conflict with symmetric FME refinement.

3.3. Rate Distortion Optimized Mode Decision

3.3.1. Data Processing Throughput Burden Analysis

Intra prediction (IP) incurs block level data dependency and makes efficient mode decision (MD) algorithm and architecture design more difficult. In general, the reconstruction loop (REC) is combined with MD. IP is usually arranged with MD at the same pipeline stage. MD algorithm is customized with IP jointly considered.

To maximize the R-D performance, the most commonly used method is the rate-distortion optimization (RDO) based MD algorithm. It evaluates the cost function (RDcost) of all candidate modes, and the mode with the minimal RDcost is selected for final coding. In some architectures, simplified MD criterion is used instead of RDcost. Three typical simplified criterions are SAD, SATD, and WSAD (weighted SAD). By employing Lagrangian optimization technique, WSAD criterion achieves superior performance than SAD or SATD criterions.

Suppose S and S' are the original MB and the reconstructed one, and P is the predicted version of a certain mode. Qp and λ are the quantization step and the Lagrange multiplier. Two typical mode decision criterions RDcost and WSAD are described by

$$RD_{cost}(mode, Qp) = SSD(S,S',mode,Qp) + \lambda \times R_{MB}(S,S',mode,Qp) \quad (2)$$

$$WSAD(mode,\lambda) = SAD(S,P,mode,Qp) + \lambda \times R_{MBheader}(S,P,mode,Qp) \quad (3)$$

SSD is the sum of the squared difference between S and S', while SAD or SATD is the SAD or SATD value between S and P. R_{MB} is the bits of all syntax elements in the MB. $R_{MBheader}$ is the coding bit of the syntax elements in the MB header.

RDO based MD achieves superior performance due to Lagrangian optimization. In the case of RDcost criterion, genuine distortion is measured with SSD, and genuine rate is used and measured with R_{MB}. Comparatively, only rate is considered in WSAD, in which rate is estimated with SAD and $R_{MBheader}$. It is the measure simplifications of rate and distortion in WSAD that result in the performance degradation compared with RDcost.

RDO based MD contribute to coding performance considerably. However, the resulting complexity is very high due to abundant candidate modes. SSD between S and the reconstructed block S' is computed for distortion measure. Rate R is computed by entropy coding (EC). In the end, RDcost is obtained according to R and SSD. The mode with the minimal RDcost is finally selected. The computation costs of R and D for all candidate modes are high. As a result, RDO based MD is computationally intensive. It is challenging to implement architecture design for genuine RDO based MD. Almost all H.264/AVC encoder architectures adopt simplified MD criterion. WSAD, SATD or SAD criterion is used instead of RDcost [15]-[21].

RDO off based MD achieves considerable complexity reduction at the cost of performance degradation. In some works [28], RDO off based mode preselection technique is employed to select partial intra and inter candidate modes, and these candidate modes are further selected using the RDO MD criterion, and the mode with the minimal RDcost is taken as the final coding mode. This combined algorithm achieves better trade off in terms of multiple target performance optimization [28].

Relatively, challenges of RDO based MD in AVS is relatively lower than that of H.264/AVC. It is possible to implement RDO based MD by adopting mode preselection to alleviate the throughput burden. RDO based MD for hardware implementation is challenged by two factors, data dependency and throughput burden.

In AVS Jizhun profile, five luminance and four chrominance modes are adopted for 8x8 block intra prediction. Thus, there are totally $5\times4 + 4\times2 = 28$ blocks to be checked for RDO based intra mode decision in 4:2:0 format videos.

There are five inter modes: P_skip, 16×16, 16×8, 8×16, and 8×8 in P frames. Comparatively, the inter prediction modes of B frames are more complex. An inter prediction mode of B frame is the combination result of two factors. One is the temporal prediction direction such as forward, backward, and bidirectional (symmetric). The other factor is the MB partition mode such as 16x16, 16x8, 8x16, and 8x8. The temporal prediction direction and MB partition mode combination results in abundant inter coding modes in B frames.

In this work, mode preselection mechanism is used for throughout burden alleviation.

http://dx.doi.org/10.5772/52965

3.3.2. Mode Preselection and Algorithm Simplification

We take two measures to alleviate the serious throughput burden. On the one hand, genuine RDO based MD is adopted for intra mode selection in I frames to sustain the fidelity of anchor frame of the whole GOP, while WSAD based MD is used for intra mode selection in P, B frames based on two considerations. One is that there are many candidate modes to be checked, so candidate mode elimination is highly expected. Another is that the percentages of intra modes is low in P and B frames, so simplified WSAD based intra mode decision in P/B frames results in negligible performance degradation.

On the other hand, two factors in MB inter prediction modes are separately selected for mode decision. Temporal prediction direction measures the temporal correlation. FME searches the quarter pixel MV justly based on this measure. So, temporal prediction direction is pre-selected at the FME stage using the WSAD criterion. The selected temporal prediction mode may be forward, backward or symmetric (f/b/s). While MB partition mode is to describe the motion consistency of one MB. If four blocks in a MB have consistent motion, the optimal MB partition mode will be 16×16. If four blocks in a MB have highly irregular motion, the optimal MB partition mode will be 8×8. The MB partition mode selection is chosen by the RDO based MD algorithm.

With the above two simplified measures, candidate modes of P and B frames are largely reduced. The worst case occurs in B frames. The temporal prediction (f/b/s) of each 8x8 mode (B_8x8) in B frames is selected using the WSAD criterion. Then, there are still two modes in each block in B_8x8 partition mode, i.e. direct mode (B_8x8_direct) and f/b/s mode (B_8x8_f/b/s). As a result, there are seven candidate modes to be selected. They are respectively skip/direct, 16x16, 16x8, 8x16, 8x8_f/b/s, 8x8_direct, and the intra mode pre-selected based on the RDO off criterion WSAD.

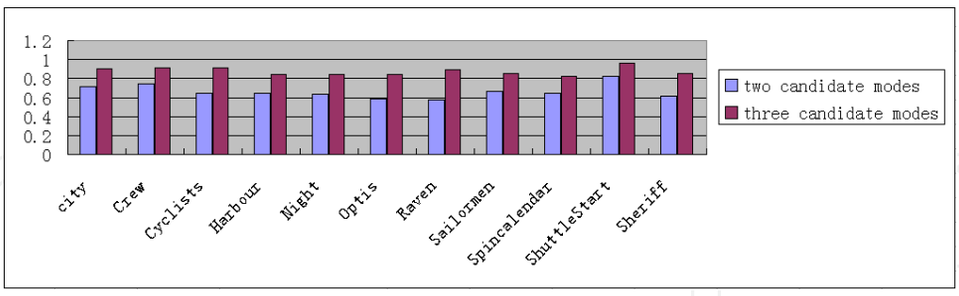

Figure 7. The mode matching probability between two and three candidate modes and the optimal mode using the RDO criterion.

There is intrinsic relationship between WSAD distribution of all modes and the optimal mode selected by RDO based MD. We find that the mode with the smallest WSAD value is usually the optimal mode selected with RDcost criterion. Certainly, these two modes mismatch sometimes. If we can preselect partial modes based on the WSAD criterion, what about the matching probability between these preselected modes and the optimal mode in the case of RDcost criterion? We had made investigation on the mode matching statistics us-

ing ten standard 720P test sequences. Fig. 7 gives the mode matching probability between two and three candidate modes and the optimal mode, which are selected by WSAD criterion and the RDcost criterion respectively. According to Fig.7, the matching probability varies from 0.6 to 0.8 in the case of two candidate modes; while the probability varies from 0.8 to 0.99 in the case of three candidate modes.

With this conclusion, we can preselect three inter modes with higher probabilities based on the WSAD criterion to further alleviate the throughput burden. Then, the selected three inter modes, the selected intra mode, and the skip/direct mode are checked using the RDO based mode decision. The simplified algorithm achieves fast decision speed by mode pre-selection and breaking the dependency between prediction direction and MB partition mode.

3.4. Data Dependency Immune Motion Vector Prediction

Residue coding is adopted for MV coding to utilize the motion field correlation. Thus, a predicted MV is desired for MV coding bit estimation for rate distortion optimized matching cost calculation. Moreover, this MV prediction is simultaneously desired at IME, FME, MD, and EC stages.

Quarter pixel accuracy MVs of the left, up, up-right, and up-left adjacent blocks in the optimal MB coding modes are employed for MV prediction. In general, IME, FME, and MD are arranged at adjacent pipeline stages. Thus, quarter pixel accuracy MVs of the blocks in the optimal modes are unavailable for MV prediction in IME and FME. This block level data dependency in spatial MV prediction disturbs the normal pipelining rhythm.

Ideally, quarter pixel MVs are desired for MV prediction at all pipelining stages. This can be easily implemented in software based encoder with sequential processing. However, it is challenged in hardware case with pipeline structure. As shown in Fig.8-(a), MV prediction for RDO based IME for current block (C00) needs quarter pixel MV of its left block (MVA) in the left MB, however it is being on the FME stage, also the MB partition mode is still unknown until FME stage has finished. Similar problem exist in the case of MV prediction for RDO based FME. Thus, algorithm simplification is desired to break this dependency.

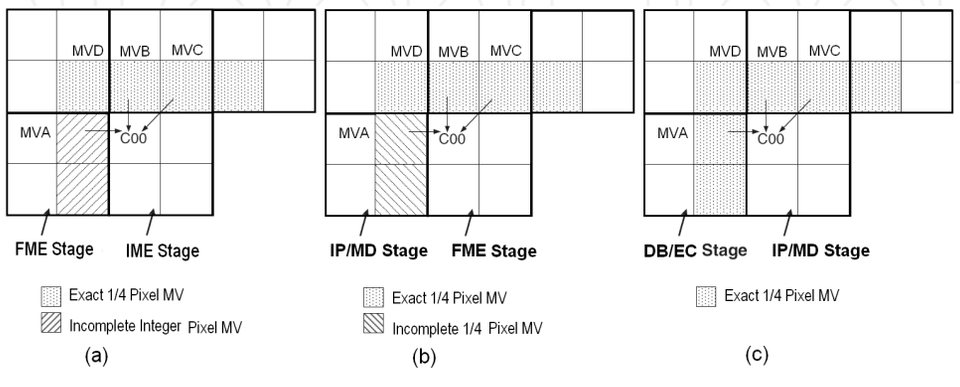

Figure 8. Dependency in MV predictor and simplified algorithm, (a): MV predictor for IME, (b): MV predictor for FME, (c): MV predictor for MD.

http://dx.doi.org/10.5772/52965

As shown in Fig.8-(a), incomplete integer pixel MV of the left block in the case of 8x8 MB partition mode is use for C00 MV prediction for IME. Similarly, incomplete quarter pixel MV of the left block is used for C00 MV prediction for FME. Here, the incomplete MV of 8x8 MB partition mode is used because the MB is being at the MD stage. Exact quarter pixel MVs of the neighboring blocks are used for MV prediction of the MD and EC stages.

4. Hardware Architecture

4.1. System Pipeline Structure and System Architecture

Pipeline structure is crucial for system architecture design. In H.264 and AVS encoder architectures, four-stage MB pipeline structure was adopted in the architectures [15] [16] [19] [20]. The sequential coding modules are arranged into four stages, and they are IME, FME, MD with IP, as well as EC and DB.

Three-stage MB-pipeline architecture was proposed to decrease the pipeline latency, and save on-chip memory buffer. The architectures in [17] [18] adopted this pipeline structure. FME and IP are combined at the same stage. However, brutal algorithm and high parallelism are desired for algorithm simplification on mode decision to satisfy the system throughput constraint.

The proposed system architecture with improved six-stage MB level pipeline structure is shown in Fig.9. In accordance with the predictive MV based local motion estimation algorithm in this work, the IME module in conventional four-stage pipeline structure is separated into three stages: IME presearch, local search window reference pixel fetch, and integer pixel VBSME.

As shown in Fig.9, the forward and backward SW reference pixels are stored in Forw. Luma Ref. Pels SRAMs and Back. Luma Ref. Pels SRAMs. Luma Ref, Reg Array and Back. Luma Ref. Reg Array, whose size is very small. Multi-resolution IME predict the center MV (MVp) first, then variable block size ME (VBSME) is performed and the local small Luma SW is transferred simultaneously into the dual-port Local Luma Ref. Pels SRAMs, by which efficient data share between IME and FME is achieved.

The chrominance (Chrom) components do not attend in matching cost calculation in IME and FME, thus it is unnecessary to load the whole Chrom SW into on-chip buffer. According to MVp, we can only load the corresponding local small chrom SW, i.e. Local Forw. Chrom Ref. Pels SRAM and Local Back. Chrom Ref. Pels SRAM. Similarly, the Forw. Chrom Reg. Array and Back. Chrom Reg. Array are employed to perform data format transform and buffering. Thus, this local SW buffer saves 80% Chrom SW SRAM consumption compared with the unoptimized case.

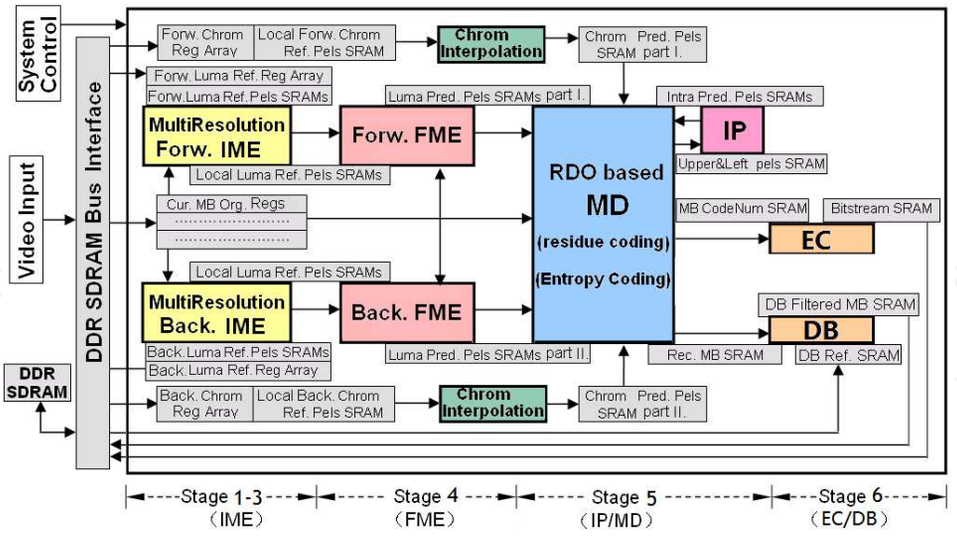

Figure 9. The proposed pipeline structure and system architecture in MPEG-like video encoder

The quarter pixel interpolation versions in the displaced blocks of all possible inter mode are buffered in the Luma Pred. Pels SRAMs (part I and II) and Chom Pred. Pels SRAM (part I and II) to implement data share between FME and IP/MD stages.

To achieve circuit reuse of the residue coding and the EC loops between IP/MD and EC/DB stages, the MB CodeNum SRAM is employed to store the CodeNum fields of all coefficients in the blocks of the selected optimal mode. Thus, bitstream can be easily generated at the following EC stage according to the CodeNum using Golomb exp-coding, and the coded bitstream is buffered in the Bitstream SRAM to wait for external SDRAM bus transactions.

4.2. Motion Estimation Architecture with MVP

In HD and ultra HD video encoders, multiple parallel processing element (PE) arrays are usually desired to improve throughput. Three-level hierarchical sequential MV refinement is employed to improve the search accuracy. So, it is preferred to adopt reconfigurable PE array structure to achieve efficient PE reuse at adjacent levels.

4.2.1. Integer Pixel Presearch

Integer pixel presearch performs successive refinements from level L_2 to level L_1. The block diagram of the proposed IME presearch architecture is given in Fig.10. It should be mentioned that integer pixel presearch only target for the center MV (MVc) estimation, and variable block size motion estimation is not adopted here.

The basic unit for motion estimation is processing element (PE), which performs SAD (sum of absolute difference) calculation for one pixel. Sixteen parallel processing element units are combined as a group i.e. processing element array (PEA). The task of processing element array (PEApq) is to calculate SAD for 4x4 block indexed by p and q shown in Fig.5-(c), which is 16:1 down-sampled from level L_0. Four-way parallel processing element array, consisting

http://dx.doi.org/10.5772/52965

of 64 parallel processing elements, work as a group as a processing element array subset (PEAS), accounting for SAD calculation of one 8X8 block (mn=01). Search luminance reference pixels and current MB are fetched from external memory and inputted to the luminance reference pixel buffer (LRPB) and current Sub-MB register (CSMR) respectively.

IME presearch controller accepts encoding parameters from MB controller and main processor, and coordinates all sub-modules for multi-resolution MV refinements. SAD values and MV costs are inputted to the WSAD adder tree and four-input WSAD comparator for SAD reuse and MV selection. Four-way PEAS parallelism structure is employed to achieve 4X speed to cover large search window for real-time HD video coding.

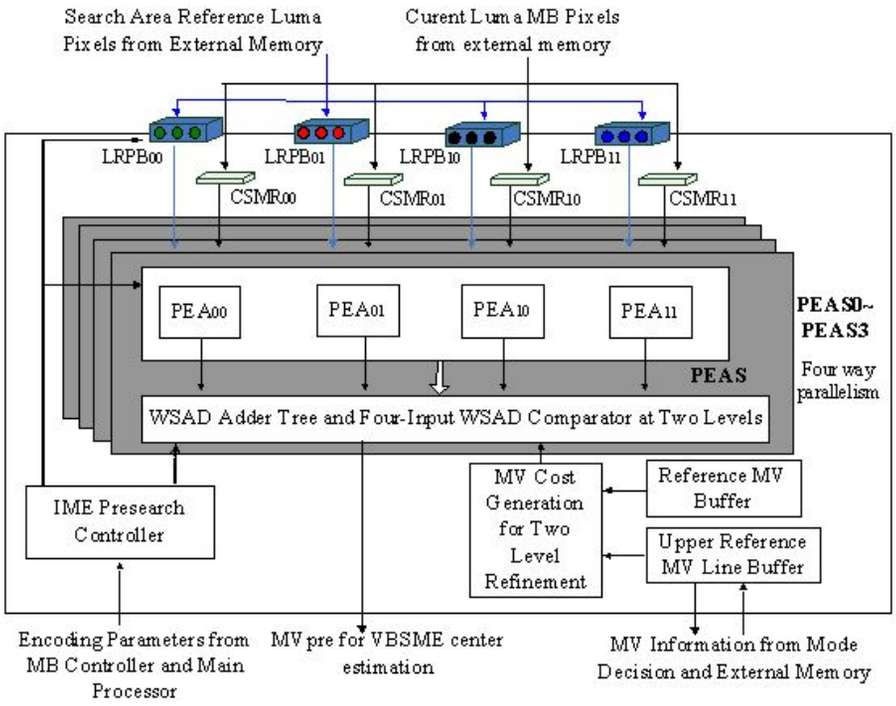

Figure 10. The architecture for integer pixel presearch.

At the level L_2 stage, four-way parallelism is employed with 4x4 processing element arrays, totally 256 processing element (PE) units. This two-dimension PE arrays achieve the throughput of sixteen candidates each cycle at level L_2 as shown in Fig.6-(a).

At the level L_1 stage, four PE array modules (PEA_{00}~PEA_{11}) are combined to implement one PE array subset (PEAS). As a result, sixteen parallel PEA units are mapped to four-way PEAS units to achieve 4-way parallel local refinement at level L_1, as a result achieving the throughput of four candidate MVs each cycle at level L_1.

4.2.2. Local Integer Pixel Motion Estimation Stage

Local integer pixel variable block size IME is at the third pipeline stage. Only block size no smaller than 8x8 is considered due to the observation that smaller block size partition achieves trivial performance improvement in HD cases with high complexity [17] [22].

Fig. 11 gives the whole structure for integer pixel variable block size IME. Triple-buffered SRAM is employed to store the luminance reference pixels centered around MVc. This structure supports simultaneous access for three clients: next MB reference pixel refreshment, current MB variable block size IME, previous MB FME.

Here, the 256-PE array is the basic unit for block matching cost calculation, and it calculates the SAD value for one candidate motion vector, 256 original and reference pixels attend in SAD calculation. In general, the larger the local search window size, the higher rate distortion performance achieved. In this work, local search window with size of 32 x 24 is used for local refinement. If only one-way 256-PE array is employed, the throughput is only one candidate MV each cycle. That means that at least 768 cycles are desired for one MB processing. In order to achieve high throughput capability, N-way parallelism may be employed. In this work, N=2 and N=3 are adopted for 1080P and QFHD format respectively.

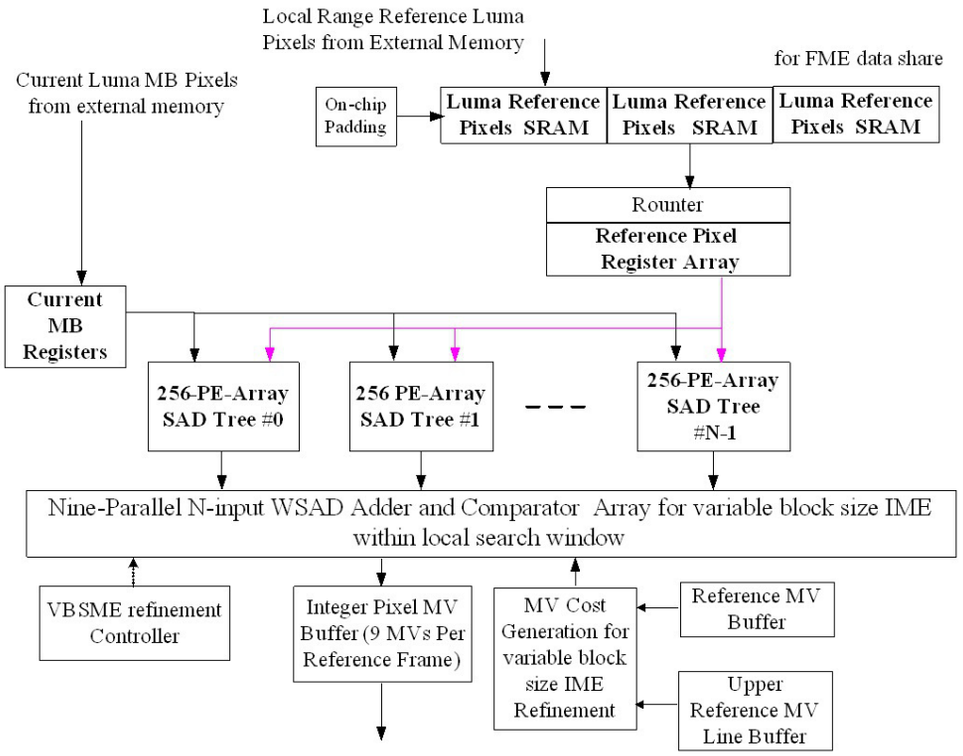

Figure 11. The structure for integer pixel variable block size motion estimation.

In order to support variable block size IME, 8x8 block is taken as the basic processing unit. Variable block size IME is implemented by employing SAD reuse [25]. Thus, 256-PE array is combined with SAD adder tree for SAD reuse to implement variable block size IME, i.e. 256-

http://dx.doi.org/10.5772/52965

PE Array SAD Adder Tree, as shown in Fig. 11. There are nine possible MB partition modes, and N adjacent motion vectors are simultaneously searched. Here, nine partition blocks are respectively 16x16, 16x8_1, 16x8_2, 8x16_1, 8x16_2, 8x8_1, 8x8_2, 8x8_3, and 8x8_4. As a result, nine SAD values of N adjacent motion vectors are simultaneously inputted into the module, titled as Nine-parallel N-input WSAD Adder and Comparator, to select nine optimal motion vectors for nine partition blocks. This architecture is similar with work in reference [15]. Variable block size mode partition is determined at the FME stage.

4.3. The Architecture of Mode Decision with IP

4.3.1. Data Dependency Removal in VLSI Architecture

An important problem in mode decision VLSI architecture design is the block level data dependency due to intra prediction. This data dependency breaks the normal pipeline rhythm for intra prediction and mode decision. An intelligent mode decision scheduling mechanism is proposed in Fig.12 and Fig.13 to eliminate the data dependency in P/B and I frames respectively.

Inter mode decision scheduling in P and B frames is used as the illustration example shown in Fig.12. First, intra modes of B_{00}, U and V blocks are successively fed to the pipeline for RDcost estimation. Then four luminance blocks and U, V blocks of the skip/direct modes are followed. Suppose T is the block level pipeline period in the MD architecture. At the time of 6T, the intra mode of B_{00} has finished the pipelining and the reconstructed pixels are ready. During the period from 7T to 8T, the intra modes of B_{01} is preselected based on WSAD criterion and then intra mode RDcost calculation for B_{01} can initiate at the time of 10T. Using the same mechanism, intra mode RDcost calculation of B_{10} and B_{11} are inserted between luminance blocks and initialized at the time of 17T and 24T. With this intelligent pipeline scheduling strategy, the data dependency problem of intra prediction is solved in P and B frames with 100% hardware utilization efficiency.

Similarly, the intra mode decision scheduling mechanism in Fig.13 is implemented with inevitable utilization discount, in which the period from 15T to 18T is idle to wait for pixel reconstruction. The RDO based intra mode decision for I frames can achieve 85.7% hardware utilization efficiency.

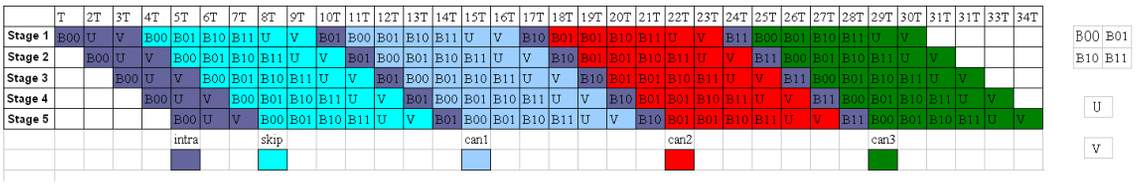

Figure 12. The pipeline scheduling strategy for intra and inter mode decision in P and B frames.

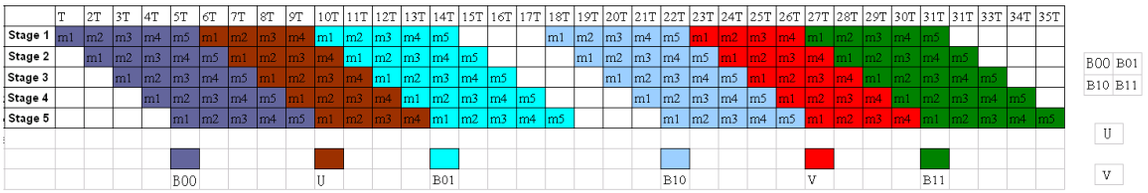

Figure 13. The pipeline scheduling strategy for intra mode decision in I frames.

4.3.2. The MD VLSI Architecture

The proposed VLSI architecture for RDO based mode decision is given in Fig.14. Seven prediction versions of the current MB are buffered in the Ping-pang buffers for data share between FME, IP, and MD. Seven prediction MB buffers (Pred. Pels. Buf.) from no. 1 to no. 7 store the predictions of the 16x16_f/s/b, 16x8_f/s/b, 8x16_f/s/b, 8x8_f/s/b, 8x8_direct, intra_preselected and direct/skip modes in P and B frames, In each mode, there are six blocks B_{00}~B_{11}, U, and V. Also, the current MB is also buffered for fluent MD pipelining.

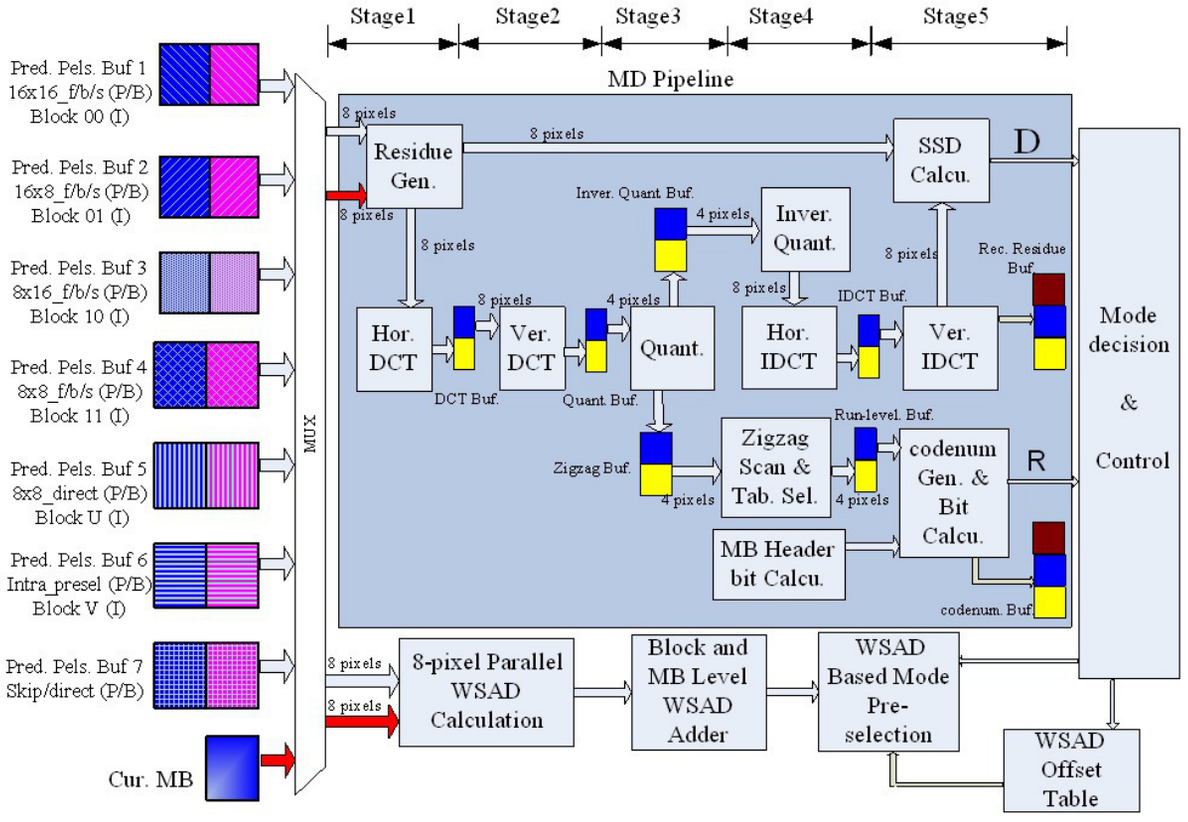

Figure 14. The proposed VLSI architecture of RDO based mode decision.

To achieve the desired throughout at MB level mode decision pipeline, eight pixels of one line in the original block and its predicted block are fetched from the buffers in each cycle

http://dx.doi.org/10.5772/52965

and fed into the residue generation (residue Gen.) module to calculate the residue in parallel. The integer DCT in AVS is based on 8x8 block. Horizontal DCT (Hor. DCT) and vertical DCT (Ver. DCT) should be processed in turn to implement the butterfly based fast DCT. Thus, Hor. DCT and Ver. DCT are arranged into adjacent block level pipeline stages to achieve high throughput with the transpose DCT buffer. Eight residue pixels in one line are fed into the Hor. DCT module in parallel.

Quantization (Quant) in AVS needs two multipliers with wide bit width. If eight-way parallel structure is employed. The circuit consumption of multiplexers will be high. Thus, Four-way parallel structure is adopted for the Quant module. Different data throughput rate between Ver. DCT and Quant are buffered and balanced by the Quant buffer. The quantized coefficients are buffered to the inverse quantization (Inver. Quant) module and zigzag buffers simultaneously with necessary data store and format transform for the following zigzag scan and Inver Quant modules. Inverse quantization is very simple, thus it is combined with Hor. IDCT at the same stage. Hor. IDCT and Ver. IDCT are similar with those of DCT. The Ver. IDCT module produces the reconstructed residue, which is fed to SSD calculation (Calcu.) module for SSD calculation and MB reconstruction by the reconstructed residue buffer.

The other parallel data processing loop is the mode preselection. The intra mode in P and B frames and the inter modes are preselected based on WSAD criterion. During the MD stage, mode pre-selection is simultaneously done by employing the eight-pixel parallel WSAD calculation, block and MB level WSAD adder, and the WSAD based mode preselection modules. WSAD offset table interface is used for algorithm optimization for WSAD based mode preselection in the future.

The codenum fields of one MB of the optimal mode are buffered in the codenum buffer for data share between MD and the following entropy coding (EC), which generates the final bitstream simply according to codenum fields instead of quantized coefficients using Exp-Golomb coding. Redundant run-level, table selection, VLC table are eliminated by codenum data share between MD and EC instead of quantized coefficients.

5. Experiment and Simulation Results

5.1. Algorithm Rate Distortion Performance

The whole modified video coding algorithms are developed and evaluated with hardware implementation consideration for Jizhun profile AVS video encoder. The modifications includes the proposed hierarchical ME algorithm, the simplified MV prediction and symmetric FME algorithms, the simplified WSAD based intra mode decision in P and B frames, the simplified inter mode decision algorithm. The modified whole video encoder is compared with the anchor algorithm, which is the standard RM52J with full search and fully RDO based MD (RM52J+FS). RM52J was developed only for syntax compliance test and verification. The source code efficiency is low. Its C model version was developed with efficient data structure and coding style. The simulation results are given in Fig.15.

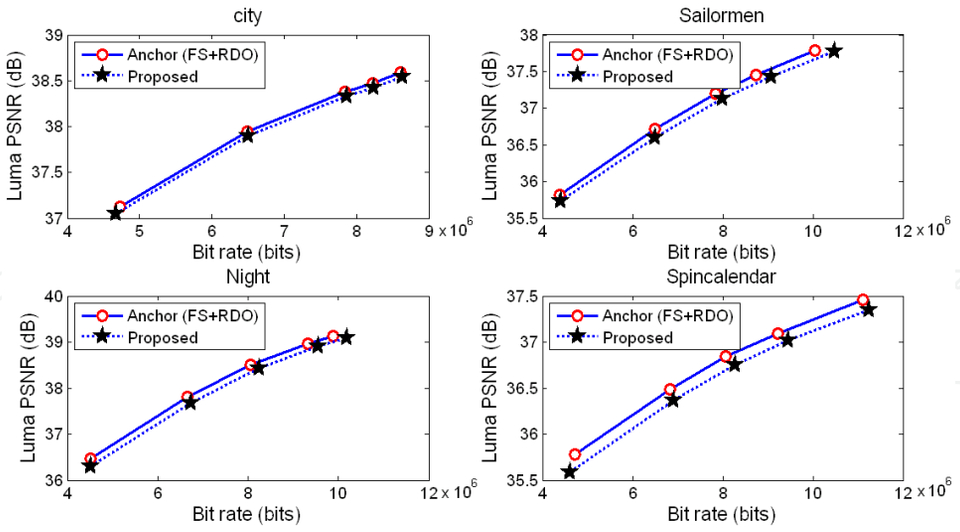

Figure 15. The rate distortion curves of typical 720P test sequences.

Compared with the optimal anchor without any simplification, the average PSNR degradation of the proposed algorithms is generally smaller than 0.1dB. The worst case occurs in the case of the "Spincalendar" sequence with complex motion and prediction modes. The worst PSNR degradation is smaller than 0.15dB. The major PSNR degradation is derived from the IME and MD algorithm simplifications. It should be emphasized that the reference benchmark is optimal algorithm with full search motion estimation and fully RD optimized mode decision. This quality loss is achieved with considerably hardware implementation convenience. The subject visual quality in HD case is almost negligible.

5.2. Hardware Cost

The proposed architecture was implemented with Verilog-HDL language and synthesized by Design complier with SMIC 130nm 1P9M standard cell library. Table 1 shows the hardware cost of the proposed architecture and comparisons to other typical reference designs. The profile and level, video format, reference frame number, variable block partition mode, pipeline structure, throughput, system frequency, gate count, and SRAM consumption are also taken as comparison factors.

According to the results in table 1, the system throughput efficiency is only lower than that of the architecture in [21]. The logic cost and SRAM consumption of this work is competitive to other reference works.

AVS Jizhun profile is equivalent to the main and high profile of H.264/AVC. B frame is supported in the proposed architecture. Two reference frame motion estimation is supported in P frame coding. As a result, the computation and throughput burden is 2X as the works with only one reference motion estimation.

http://dx.doi.org/10.5772/52965

	T.C. Chen [15]	H.C.Chang [16]	Z. Y. Liu [17]	Y. K. Lin [18]	Yi. H. Chen [19]	L.F. Ding [20]	Proposed
Profile/level	Baseline@ Level3.1	Baseline@ Level 3.1	Baseline@Level4	High@ Level4	High/ SVC	Multiview/ High	Jizhun
Video format	720p@ 30fps	720p@ 30fps	Full HD	Full HD	Full HD	FullHD / QFHD	Full HD @60i
Ref. frame	1/4	1	1	1	3	1	2
VBS Mode	All	All	above 8x8	All	All	All	above 8x8
Intra Mode	All	All	All	All	All	All	All
Pipeline	4 stage	4 stage	3 stage	3 stage	4 stage	8 stage	6 stages
Throughput (cycles/MB)	1000	1000	820	592	678	312	600
frequency	108Mhz	108MHz	200Mhz	145Mhz	166Mhz	280Mhz	150Mhz
Logic Gate	923K	470K	1140K	593K	2079K	1732K	1230
SRAM Size	35KB	13.3KB	108.3KB	22KB	81.7KB	20.1KB	29KB
Power	785mW	183mW	1409mW		411mW	522mW	764mW
CMOS Technology	UMC .18um 1P8M	TSMC 0.13um	0.18um 1P6M	UMC 0.13um	UMC 90nm	TSMC 90nm	UMC .13um 1P8M

Table 1. Comparison of the High Definition Encoder Chips and Architectures

Fair R-D performance is not easy. Different reference codes and different test sequences are used for rate distortion performance comparison. Comparison is expected using the same test sequences with the same benchmark. More important, the frame level average PSNR results are used in current works. PSNR may not be accurate for perceptual video quality. As a result, the PSNR comparison to reference works are not provided in this work due to the fact that fair comparison.

Multi-resolution motion estimation and partial RDO based mode decision are employed in this work. As a result, the coding performanc of this work is competitive to other works with brunt algorithm simplifications on motion estimation and mode decision.

6. Conclusion and Future Works

6.1. Conclusions

In this chapter, we propose systematic works for efficient high definition MPEG-like, such as AVS, video encoder VLSI implementation. VLSI Implementation challenge, algorithm and architecture joint design, and multiple target performance optimizations are reviewed and analyzed. Hardware oriented algorithm customizations are made for the crucial algo-

rithm customizable modules, including integer pixel and fractional pixel motion estimation, mode decision, intra prediction, and motion vector prediction. Then, hardware architectures of the crucial modules, motion estimation with motion vector prediction, mode decision with intra prediction and reconstruction loop, are proposed. Algorithm performance and multiple target performance are respectively reported. This work targets for high definition AVS video encoder architecture. However, the algorithms and architectures are well-suited for H.264/AVC, even HEVC video encoder design.

6.2. Future Trends

Perspectives and future trends in video encoder architecture design may focus on the following factors. First, low-power video encoder architecture is desired for in mobile systems or portable applications. Low power algorithm and architecture design are one emerging issue. Second, professional video encoder architecture is preferred in the high-end applications such as the broadcasting enterprises. Third, algorithm and architecture joint design will be an important trend for HD video encoder architecture. Fourth, algorithm and architecture design for ultra HD and HEVC video encoder will become one research trend.

Acknowledgements

This work was supported by NSFC 60802025, ZJNSF Y1110114 LY12F01011, S&T project of Zhejiang province 2010C310075, and the open project of State Key Laboratory of ASIC&System of Fudan University 10KF010, and the open project of SKL of Novel Soft Technology, Nanjing University (KFKT2012B09).

Author details

Haibing Yin*

Address all correspondence to: yinhb@cjlu.edu.cn

Jiliang University, China

References

[1] ISO/IEC. (1993). Information technology- Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s- Part 2: Video.

[2] ISO/IEC13818-2. (2000). Information technology- Generic coding of moving pictures and associated audio information: Video.

[3] ISO/IEC. (2004). Information technology-Coding of audio-visual objects- Part 2: Visual. *ISO. Retrieved 2009-10-30*.

[4] ITU-T (2005). Recommendation and International Standard of Joint Video Specification. ITU-T Rec. H.264/ ISO/ IEC AVC, Mar.., 14496-10.

[5] SMPTE 421M. VC-1 Compressed Video Bit stream Format and Decoding Process. http://www.smpte.org/smpte_store/standards/pdf/ s421m.pdf.

[6] Documents of the first meeting of the Joint Collaborative Team on Video Coding (JCT-VC)- Dresden, Germany. (2010). 15-23 April,ITU-T. 23 April. Retrieved 21 May.

[7] Information technology- Advanced coding of audio and video- Part 2: Video. (2005). *AVS Standard Draft*.

[8] Wiegand, T., Sullivan, G. J., Bjontegaard, G., & Luthra, A. (2003). Overview of the H. 264/AVC video coding standard. IEEE Trans. Circuits Syst. Video Technol Jul.., 13, 560-576.

[9] http://en.wikipedia.org/wiki/x264.

[10] MSU Seventh MPEG-4 AVC/H.264 Video Codecs Comparison. http://compression.ru/video/codec_comparison/h264_2011/.

[11] Po-Chin, Tseng., Yung-Chi, Chang., Yu-Wen, Huang., Hung-Chi, Fang., Chao-Tsung, Huang., & Liang-Gee, Chen. (2005). Advances in Hardware Architectures for Image and Video Coding-A Survey. Proceeding of IEEE January,., 93(1), 184-197.

[12] Yu-Wen, Huang., Bing-Yu, Hsieh., Tung-Chien, Chen., & Liang-Gee, Chen. (2005). Analysis, fast algorithm, and VLSI architecture design for H.264/AVC intra frame coder. *in IEEE Transactions on Circuits and Systems for Video Technology*, 15(3), 378-401.

[13] Yu-Kun, Lin., Chun-Wei, Ku., De -Wei, Li., & Tian-Sheuan, Chang. (2009). A 140-MHz 94 K Gates HD1080p30s Intra-Only Profile H.264 Encoder. IEEE Trans. Cir. Syst. Video Tech March., 19(3)

[14] Chun-Wei, Ku., Chao-Chung, Cheng., Guo-Shiuan, Yu., Min-Chi, Tsai., & Tian-Sheuan, Chang. (2006). A High-Definition H.264/AVC Intra-Frame Codec IP for Digital Video and Still Camera Applications. IEEE Trans. Cir. Syst. Video Tech August., 16(8)

[15] Huang, W. Y., et al. (2005). A 1.3 TOPS H.264/AVC single-chip encoder for HDTV applications. in IEEE ISSCC Dig. Tech. Papers, Feb., , 128-129.

[16] Hsiu-Cheng, Chang., Jia-Wei, Chen., Bing-Tsung, Wu., Ching-Lung, Su., Jinn-Shyan, Wang., & Jiun-In, Guo. (2009). A Dynamic Quality-Adjustable H.264 Video Encoder for Power-Aware Video Applications. IEEE Trans. Cir. Syst. Video Tech Dec, 19(12), 1739-1754.

[17] Zhenyu, Liu., Yang, Song., & Satoshi, Goto. (2009). HDTV 1080P H.264/AVC Encoder Chip Design and Performance Analysis. IEEE Journal of Solid-state Circuits Feb,., 44(2)

[18] Yu-Kun, Lin., Chia-Chun, Lin., Tzu-Yun, Kuo., & Tian-Sheuan, Chang. (2008). A Hardware-Efficient H.264/AVC Motion-Estimation Design for High-Definition Video. IEEE Trans. Cir. Syst. Video Tech July,., 55(6)

[19] Yi-Hau, Chen., Tzu Der, L. G., & Chuang , Chen. (2008). An H.264/AVC Scalable Extension and High Profile HDTV 1080p Encoder Chip. 2008 Symposium on VLSI Circuits Digest of Technical Papers Jun., 104-105.

[20] Yu-Han, Chen., Tung-Chien, Chen., Chuan-Yung, Tsai., Sung-Fang, Tsai., & Liang-Gee, Chen. (2009). Algorithm and Architecture Design of Power-Oriented H. 264/AVC Baseline Profile Encoder for Portable Devices. IEEE Trans. Cir. Syst. Video Tech August,., 19(8), 1118-1128.

[21] Li-Fu, Ding., Wei-Yin, Chen., et al. (2010). A 212 MPixels/s 4096 2160p Multiview Video Encoder Chip for 3D/Quad Full HDTV Applications. IEEE Journal of Solid-State Circuits Jan,., 45(1), 46-58.

[22] Hai, bing., Yin, Hong., gang, Qi., Huizhu, Jia., Don, Xie., & Wen, Gao. (2010). Efficient Macroblock Pipeline Structure in High Definition AVS Video Encoder VLSI Architecture. *IEEE International Symposium on Circuits and Systems, Paris, France, 30 May-2 June*.

[23] Tuan, C. J., Chang, S. T., & Jen, C.W. (2002). On the data reuse and memory bandwidth analysis for full-search block-matching VLSI architecture. IEEE Trans. Circuits Syst. Video Technol Jan., 12(1), 61-72.

[24] Ching-Yeh, Chen., Chao-Tsung, Huang., Yi-Hau, Chen., & Liang-Gee, Chen. (2006). Level C+ Data Reuse Scheme for Motion Estimation With Corresponding Coding Orders. IEEE Trans. Circuits Syst. Video Technol April., 16(4)

[25] Yu-Wen, Huang., Ching-Yeh, Chen., et al. (2006). Survey on Block Matching Motion Estimation Algorithms and Architectures with New Results. *Journal of VLSI Signal Processing*, 42, 297-320.

[26] Haibing, Yin., Huizhu, Jia., & Wen, Gao. (2010). A Hardware-Efficient Multi-resolution Block Matching Algorithm and its VLSI Architecture for High Definition MPEG-like Video Encoders. IEEE Trans. Circuits Syst. Video Technol September, 20(9), 1242-1254.

[27] Tsung-Han, Tsai., & Yu-Nan, Pan. (2011). High Efficiency Architecture Design of Real-Time QFHD for H.264/AVC Fast Block Motion Estimation. IEEE Trans. Circuits Syst. Video Technol Nov., 21(11), 1646-1658.

[28] Hai bing., Yin, Honggang., Qi, Zhu, Hui., Chuang, Jia., Xiao, Zhu., & Xie, Dong. (2010). Algorithm Analysis and Architecture Design for Rate Distortion Optimized

Mode Decision in High Definition AVS Video Encoder. Signal Processing: Image Communication October, 25(9), 633-647.

[29] Heng-Yao, Lin., Kuan-Hsien, Wu., Bin-Da, Liu., & Jar-Ferr, Yang. (2010). An Efficient VLSI Architecture for Transform-Based Intra Prediction in H.264/AVC, IEEE Trans. Circuits Syst. Video Technol June, 20(6), 894-906.

[30] Chih-Hung, Kuo., Li-Chuan, Chang., & Kuan-Wei, Fan. (2010). Hardware/Software Codesign of a Low-Cost Rate Control Scheme for H.264/AVC, IEEE Trans. Circuits Syst. Video Technol Feb, 20(2), 250-261.

[31] Xiang, Li., Norbert, Oertel., et al. (2009). Laplace Distribution Based Lagrangian Rate Distortion Optimization for Hybrid Video Coding, IEEE Trans. *Circuits Syst. Video Technol*, 19(2), 193-205.

[32] Lian, C. J., Chien, S. Y., Lin, C. P., Tseng, P. C., & Chen, L. G. (2007). Power aware multimedia: Concepts and design perspectives. *IEEE Circuits Syst. Mag.*, 7(2), 26-34.

[33] Zhou, J. D., Zhou, X., He, J., Zhu, J., Kong, P., & Liu, S. Goto. (2011). A 530 Mpixels/s 4096 2160@60 fpsH.264/AVC high profile video decoder chip. *IEEE J. Solid-State Circuits*, 46(4), 777-788.

[34] Kim, H., & Park, I. (2001). High-performance and low-power memory interface architecture for video processing application. IEEE Trans. Circuits Syst. Video Technol Nov., 11(11), 1160-1170.

IntechOpen