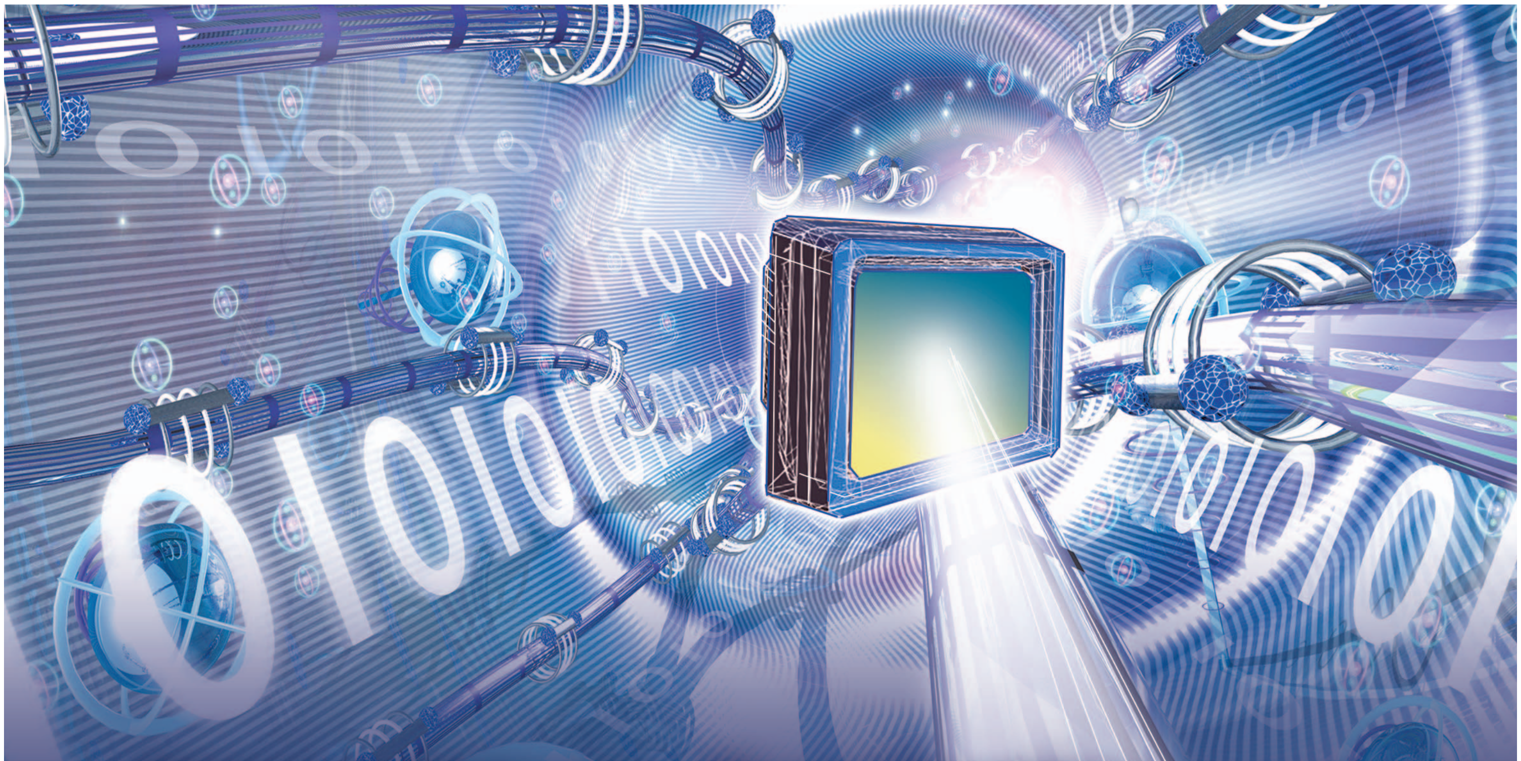


Algorithm Analysis and Architecture Design for HDTV Applications

A Look at the H.264/AVC Video Compressor System

Tung-Chien Chen, Hung-Chi Fang, Chung-Jr Lian, Chen-Han Tsai, Yu-Wen Huang, To-Wei Chen, Ching-Yeh Chen, Yu-Han Chen, Chuan-Yung Tsai, and Liang-Gee Chen

The new H.264/AVC coding standard significantly outperforms previous video coding standards with many new coding tools. However, the high performance comes at a price. Its extraordinarily huge computational complexity and memory access requirement makes it difficult to design a hardwired codec for real-time applications. Furthermore, due to the complex, sequential, and highly data-dependent characteristics of the essential algorithms in H.264/AVC, both the pipelining and the parallel processing techniques are too constrained to be directly employed. The hardware utilization and throughput are also decreased because of the block/macroblock/frame-level reconstruction loops. In this article, we suggest some techniques to design the H.264/AVC video coding system for HDTV applications. The design exploration is made according to software profiling. The design considerations of system scheduling and pipelining are discussed followed by the architecture optimization of the significant modules. The efficient H.264/AVC video coding system is achieved by combining these techniques.

H.264 Overview

H.264/AVC can save 25–45% and 50–70% of bit rates compared with MPEG-4 advanced simple profile (ASP) and MPEG-2, respectively [1]. Although the motion-compensated transform-coding structure is still adopted, many new features are used to achieve much better compression performance and subjective quality. To remove spatial redundancy, H.264/AVC intraprediction suggests 13 prediction modes to improve prediction. To remove more temporal redundancy, interprediction is enhanced by motion estimation (ME) with quarter-pixel accuracy, variable block sizes (VBSs), and multiple reference frames (MRFs). Moreover, the advanced entropy coding tools use content adaptivity to reduce more statistic redundancy. The perceptual quality is improved by the in-loop deblocking filter. Interested readers can refer to [2]–[4] for a quick and thorough study.

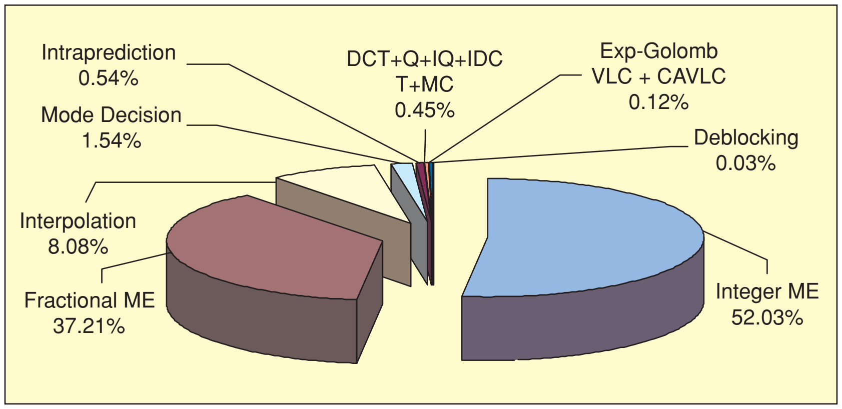
There are many potential applications of H.264/AVC. Ongoing applications range from high-definition digital video disc (HD-DVD) or BluRay for home entertainment to digital video broadcasting for handheld terminals (DVB-H). However, computational complexity comes with the coding performance of H.264/AVC. According to the instruction profiling with the HDTV specification, the H.264/AVC decoding process requires 83 giga-instructions per second (GI/s) computation and 70 GB/s memory access. As for the H.264/AVC encoding process, up to 3,600 GI/s computation and 5,570 GB/s memory access are required. For real-time applications, the acceleration by dedicated hardware is a must.

It is difficult to design efficient architectures for an H.264/AVC hardwired codec. In addition to extraordinarily huge computational complexity and the memory access requirement, the coding path is very long because it includes intra/interprediction, block/macroblock (MB)/frame-level reconstruction loops, entropy coding, and in-loop deblocking filter. The reference software [5] adopts sequential processing of many blocks in one MB, which restricts the parallel architecture design for hardware. The block-level reconstruction loop caused by intraprediction induces the bubble cycles and decreases the hardware utilization and throughput. Some coding tools have multiplex modes. A larger gate count is required if multiple processing elements (PEs) are designed for different modes without resource sharing and data reuse. Some coding tools involve many data dependencies to enhance the coding performance. Considerable storage space also is required to store the correlated data during the encoding process.

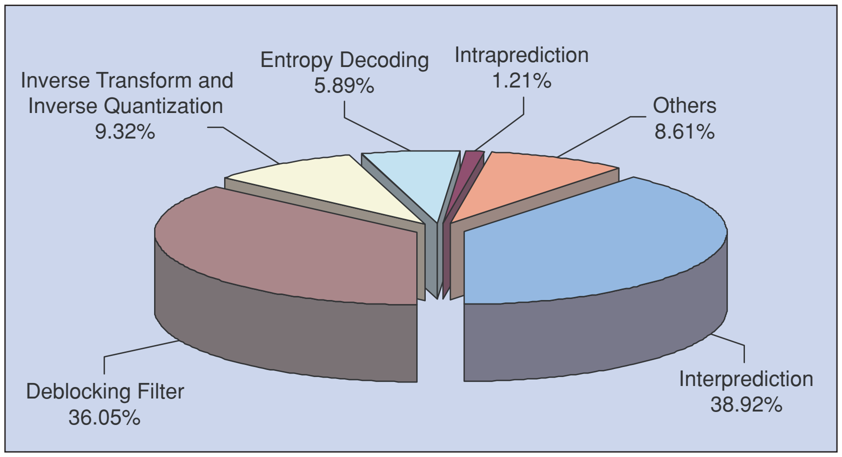
Software Profiling

We will use software profiling to show the necessity of acceleration by the dedicated hardware and to find the critical parts of the whole. To focus on the target specification, a software C model is developed by extracting all baseline profile compression tools from the reference software [5]. The iprof [6], a software analyzer on the instruction level, is used for the instruction profiling. The focused design case is targeted at SDTV [720 × 480, 30 frames per second (f/s)]/HDTV720p (1280 × 720, 30 f/s) videos with a maximum reference frame number of four/one f/s and a maximum search range (SR) of H[−64, +63] and V[−32, +31]. According to the simulation results, the computational complexity and memory access for SDTV/HDTV720p are 2,470/3,600 GI/s and 3,800/5,570 GB/s. It is about ten times more complex than that of MPEG-4 SP [7]. This is mainly due to MRF-ME and VBS-ME in interprediction. For the full search (FS) algorithm, the complexity of integer ME (IME) is proportional to the number of reference frames while that of fractional ME (FME) is proportional to the MB number constructed by variable blocks and the number of reference frames. The huge computational loads are far beyond the capability of today's general-purpose processors. The run-time percentages of P-frames in the H.264/AVC encoder are shown in Figure 1. Interprediction occupies 97.32% computation and is the processing bottleneck of the H.264/AVC interframe coding. Mode decision and intraprediction dominate the rest and occupy 77% computation of intraframe coding.

As for the decoder with HDTV1024p (2048 × 1024, 30 f/s) specification, 83 GI/s and 70 GB/s of computation and memory access are required, which is about two to three times more complex than MPEG-4 SP. The run-time percentages of several main tasks are shown in Figure 2. The interprediction and deblocking filter contribute the most computation time (39% and 38%), while IQ/IDCT, entropy decoding, and intraprediction



1. Run-time profile of interframe for H.264/AVC encoding procedure.



2. Run-time profile of H.264/AVC decoding [8].

occupy the rest. Note that the complexity of the encoding process is much higher than that of the decoding process.

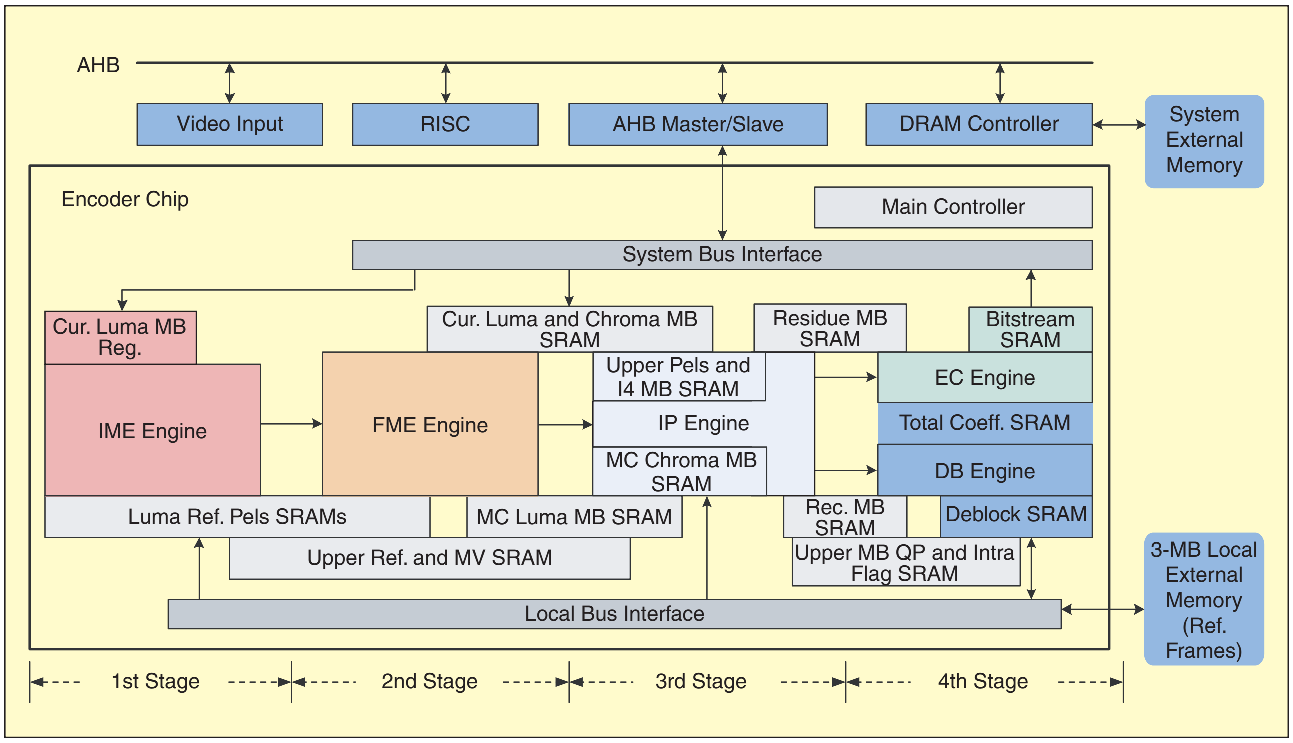
Design Space Exploration

The first major design challenge of an H.264/AVC hardwired compression system is parallel processing under the constraint of sequential flow. According to the software profiling, H.264/AVC requires much more computational complexity than the previous coding standards. Therefore, a high degree of parallel processing is required, especially for HDTV applications. However, the H.264/AVC reference software [5] adopts many sequential processes to enhance the compression performance. It is hard to efficiently map a sequential algorithm to a parallel hardware architecture. For system scheduling, the coding path, which includes intra/interprediction, block/MB/frame-level reconstruction loops, entropy coding, and in-loop deblocking filters, is very long. The sequential encoding process should be partitioned into several tasks and processed MB by MB in pipelined structure to improve the hardware utilization and throughput. For module architecture, the problem of sequential algorithm is critical for ME because it is the most computationally intensive part and requires the greatest degree of parallelism. The FME must be done after the IME. Besides, in FME, the quarter-pixel-precision refinement must be processed after the half-pixel-precision refinement. Moreover, the inter-Lagrangian mode decision takes motion vector (MV) costs into consideration, which also causes inevitable sequential processing. The modified hardware-oriented algorithms are required to enable parallel processing without noticeable quality drop. The analyses in processing loops and data dependencies are also helpful to map the sequential flow into the parallel hardware.

The second design challenge of an H.264/AVC hardwired compression system is reconstruction loops. In addition to the frame-level reconstruction loop for ME and motion compensation (MC) in H.264/AVC, the intraprediction induces the MB- and block-level reconstruction loops. Because the reconstructed pixels of the left and upper neighboring MBs/blocks are required to predict the current MB/block, the intraprediction of the current MB/block cannot be performed until the neighboring MBs/blocks are reconstructed. The reconstruction latency is harmful for hardware utilization and throughput if the intraprediction and reconstruction engines are not jointly considered and scheduled.

Data dependencies are the third design challenge. The new coding tools remove more temporal, spatial, and statistic redundancies by use of many data dependencies. The frame-level data dependencies contribute to the considerable system bandwidth. The dependencies between neighboring MBs constrain the solution space of MB pipelining, and those between neighboring blocks limit the possibility of parallel processing. Since a great deal of data and coding information may be required by the following encoding and decoding procedures, the storage space of both off-chip memory and on-chip buffer are largely increased. To reduce the chip cost, the functional period or lift-time of these data must be jointly considered with the system architecture and the processing schedule.

The fourth problem is abundant modes. Many coding tools of H.264/AVC have multiplex modes. For example, there are 17



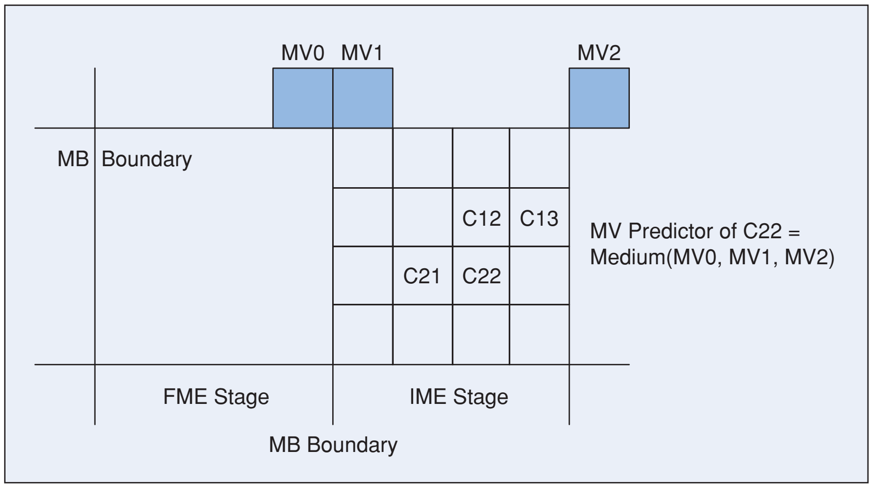
3. Block diagram of the H.264/AVC encoding system with four-stage MB pipelining. Five major tasks, including IME, FME, IP, EC, and DB, are partitioned from the sequential encoding procedure and processed MB by MB in pipelined structure [12].

different modes for intraprediction and 259 kinds of partitions for interprediction. Six kinds of two-dimensional (2-D) transform functions, 4 × 4/2 × 2 DCT/IDCT/Hadamard transforms, are involved in reconstruction loops. Adaptive filter taps and two-filter direction also must be supported for in-loop deblocking filters. Reconfigurable processing engines and reusable prediction cores are important to efficiently support all these functions.

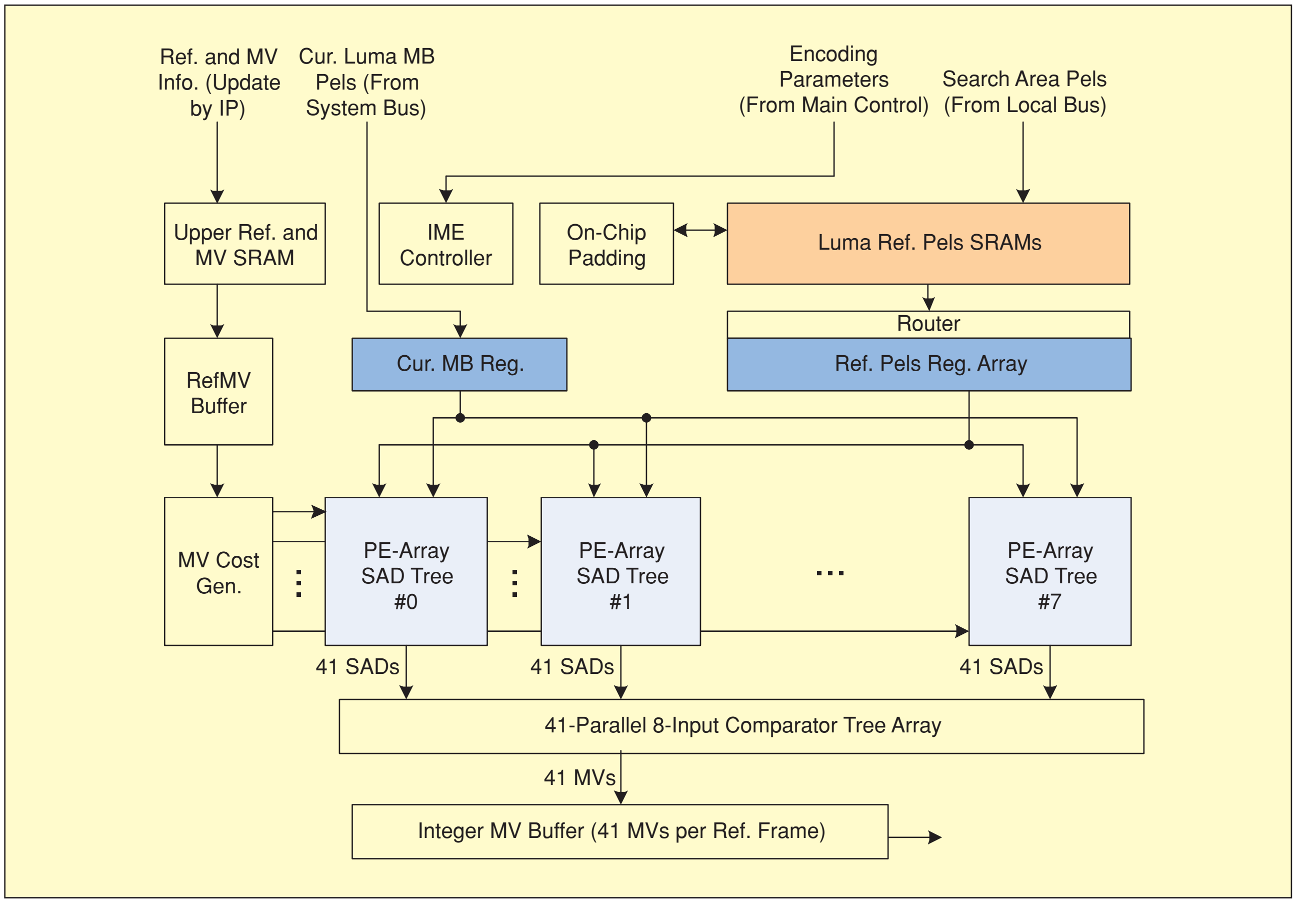
Last but not least, the bandwidth requirement of the H.264/AVC encoding system is much higher than that of the previous coding standards. The MRF-ME contributes the heaviest traffic for loading reference pixels. Neighboring reconstructed pixels are required by intraprediction and deblocking filters. Lagrangian-mode decision and context-adaptive entropy coding have data dependencies between neighboring MBs, and transmitting related information contributes considerable bandwidth as well. Hence, an efficient memory hierarchy combined with data sharing and data reuse (DR) schemes must be designed to reduce the system and the local memory bandwidth.

Architecture of H.264/AVC Encoding System

The traditional two-stage MB pipelining [9], [10], ME and block engine (BE), cannot be efficiently applied to H.264/AVC. We have extracted five major functions from the H.264/AVC encoding procedure and mapped them into four-stage MB pipelining with suitable task scheduling [11]. To complete the system, we will also describe the design consideration and



4. The modified MVP. To facilitate the parallel processing and MB pipelining, the MVPs of all 41 blocks are changed to the medium of MV0, MV1, and MV2.


Ref. and MV Info. (Update by IP)
Cur. Luma MB Pels (From System Bus)
Encoding Parameters (From Main Control)
Search Area Pels (From Local Bus)
Upper Ref. and MV SRAM
IME Controller
On-Chip Padding
Luma Ref. Pels SRAMs
Router
RefMV Buffer
Cur. MB Reg.
Ref. Pels Reg. Array
MV Cost Gen.
PE-Array SAD Tree #0
PE-Array SAD Tree #1
PE-Array SAD Tree #7
41 SADs
41 SADs
41 SADs
41-Parallel 8-Input Comparator Tree Array
41 MVs
Integer MV Buffer (41 MVs per Ref. Frame)

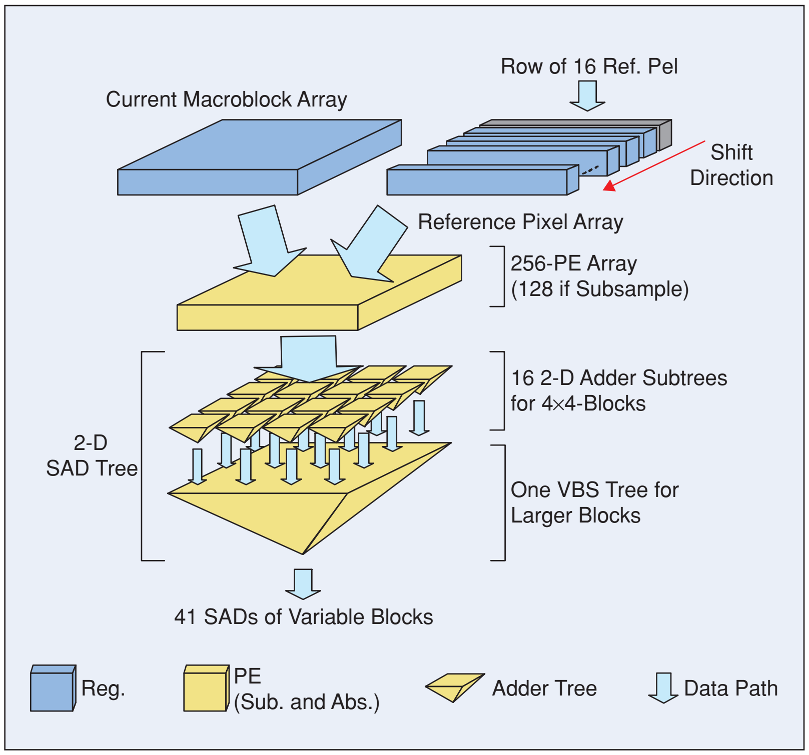
5. Block diagram of the low-bandwidth parallel IME engine. It mainly comprises eight PE-array SAD trees, and eight horizontally adjacent candidates are processed in parallel.

optimization for the significant modules in the following sections. With these techniques, an efficient implementation for an H.264/AVC encoding system can be achieved [12].

The system architecture of four-stage MB pipelining is shown in Figure 3. Five major tasks—IME, FME, intraprediction with reconstruction loop (IP), entropy coding (EC), and in-loop deblocking filter (DB)—are partitioned from the sequential encoding procedure and processed MB by MB in pipelined structure.

This system pipelining has several design issues. The prediction includes IME, FME, and intraprediction in H.264/AVC. Because of the diversity and computational complexity of these algorithms, it is difficult to implement IME, FME, and intraprediction with the same hardware. But, if we put IME and FME engines in the same MB pipeline stage, it leads to very low utilization due to the sequential processing. Even if the resource sharing is achieved, the operating frequency becomes too high. Therefore, FME is initially pipelined MB by MB after IME to double the throughput. As for intraprediction, because of the MB- and block-level reconstruction loop, it cannot be separated from the reconstruction engine. In addition, the reconstruction process should be separated from ME to achieve the highest hardware utilization, just like the two-stage MB pipelining structure. Therefore, engines of intraprediction together with forward/inverse transform/quantization are located in the same IP stage. In this way, the MB- and block-level reconstruction loops can also be isolated in this pipelining stage. The EC encodes MB headers and residues after transformation and quantization. The DB generates the standard-compliant reference frame after reconstruction. Since the EC/DB can be processed in parallel, they are placed at the fourth stage. The reference frame will be stored in the external memory for the ME of the next frame, which constructs the frame-level reconstruction loop. Note that the luma MC is placed in FME stage to reuse the interpolation circuits and the Luma Ref. Pels SRAMs. The compensated MB is transmitted to the IP stage for generation of residues after mode decision between intra- and intermodes. On the other hand, chroma MC is implemented in the IP stage since it can be executed only after the intra-/intermode decision. In summary, five main functions extracted from the encoding procedure are mapped into the four-stage MB pipelined structure. To achieve high utilization, the processing cycles of the four stages are balanced with different degrees of parallelism.

As for the reduction in system bandwidth, many on-chip memories are used for several purposes. First, to find the best matched candidate, a huge amount of reference data is required by IME and FME. Since the pixels of neighboring candidate blocks are considerably overlapped, as are the search windows (SWs) of neighboring current MBs, the bandwidth of the system bus can be greatly reduced if we design the local buffers to store reusable data. Second, rather than be transmitted by the system bus, the raw data, such as luma motion-compensated MB, transformed and quantized residues, and reconstructed MB, are shifted forward via shared memories. Third, because of the data dependency, one MB is processed according to the data of the upper and the left MBs. The local memories, rather than the system memory, are used to store the related data during the encoding process. For the software implementation, the external bandwidth requirement is up to 5,570 GB/s. As for the hardware solution with an embedded local search window buffer, the external bandwidth requirement is reduced to 700 MB/s. After all these techniques are applied, the final external bandwidth requirement is about 280 MB/s.



6. PE-array SAD tree architecture. The costs of 16 4 × 4-blocks are separately summed up by 16 2-D subtrees and then reused by one VBS tree for larger blocks.

Low-Bandwidth Parallel Integer ME

The IME searches for the best matches in coarse resolution for all block sizes and reference frames. With the given SR and reference frame number, the Lagrangian matching costs are calculated for all candidates in the FS algorithm. The IME requires the most computational complexity and memory bandwidth in H.264/AVC. A large degree of parallelism is required for the SDTV/HDTV specifications, but the sequential Lagrangian mode decision flow makes it impossible to design the parallel architecture for IME. Techniques on algorithmic and architectural levels are used to enable the parallel processing and reduce the required memory bandwidth.

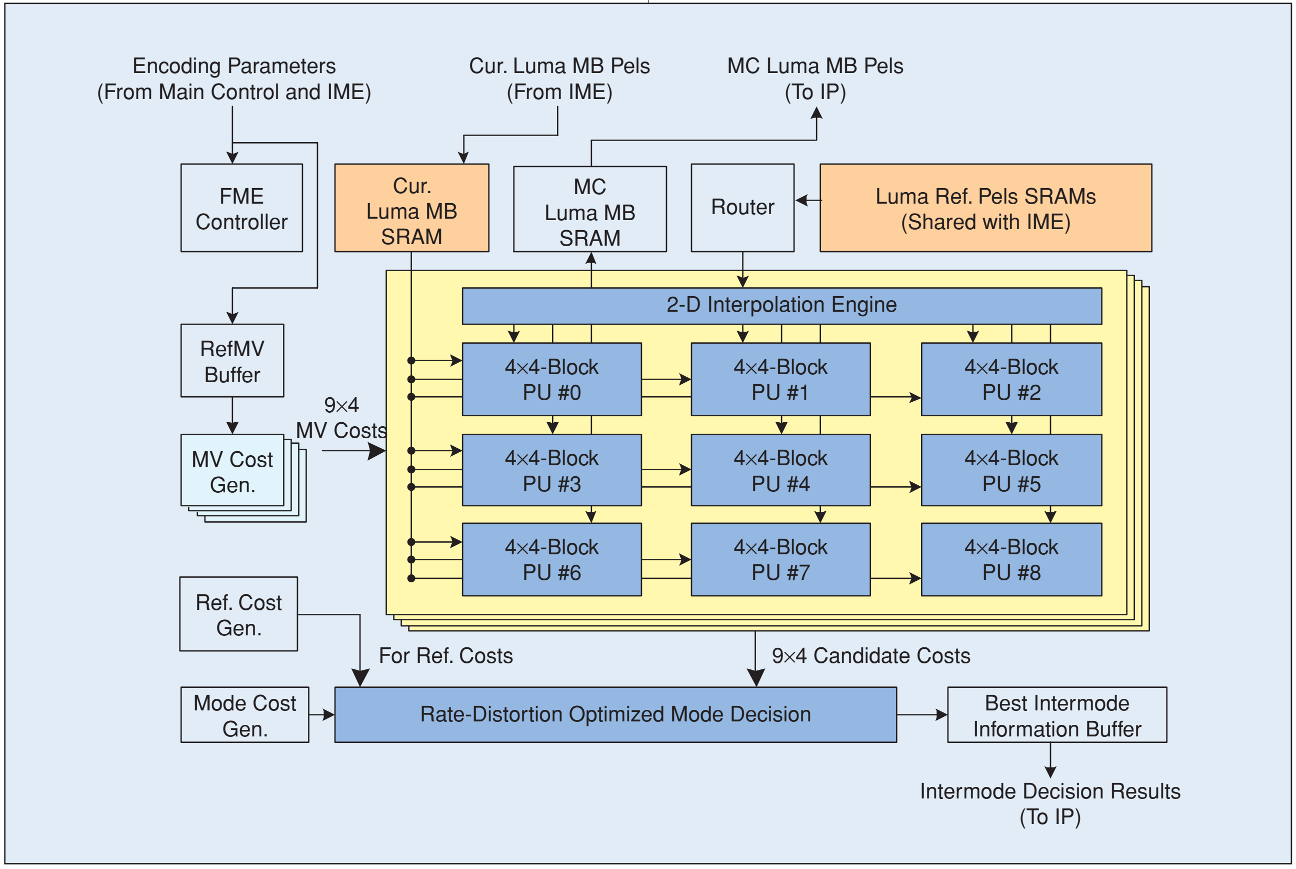
The MV of each block is generally predicted by the medium values of MVs from the left, up, and up-right neighboring blocks. The rate term of the Lagrangian cost function can be computed only after MVs of the neighboring blocks are determined, which causes inevitable sequential processing. To solve this problem, the modified MV predictor (MVP) is applied for all of the 41 blocks in one MB, as shown in Figure 4. The exact MVPs of variable blocks are changed to the medium of MVs of the up-left, up, and up-right MBs. For example, the exact MV cost of the C22 4 × 4-block is the medium of the MVs of C12, C13, and C21. We change the MVPs of all 41 blocks to the medium of MV0, MV1, and MV2 to facilitate the parallel processing. At high bit rates, which are larger than 1 Mb/s for SDTV videos, the quality loss is near zero. At low bit rates, the quality degradation is about 0.1 dB [13].

H.264/AVC's extraordinarily huge computational complexity and memory access requirement makes it difficult to design a hardwired codec for real-time applications.

Figure 5 shows the low-bandwidth parallel IME architecture, which mainly comprises eight PE-array SAD trees. Each PE array and its corresponding 2-D SAD tree compute the 41 SADs of variable blocks for one candidate in parallel. Eight horizontally adjacent candidates are processed in each cycle. Because SWs of neighboring current MBs are considerably overlapped, as are the pixels of neighboring candidates, a three-level memory hierarchy—including external memory, Luma Ref. Pels SRAMs, and Ref. Pels Reg. array—is used to reduce bandwidth requirement. Three kinds of DR are implemented: MB-level DR, intercandidate DR, and intracandidate DR.

The Luma Ref. Pels SRAMs are embedded first to achieve MB-level DR. When the ME process is changed from one current MB (CMB) to another CMB, there is an overlapped area between neighboring SWs. Therefore, the reference pixels of the overlapped area can be reused in local SRAMs. The MB-level DR can greatly reduce the external memory bandwidth.

After that, the Ref. Pels Reg. array acts as the cache between PE-array 2-D SAD tree and luma Ref. Pels SRAMs. It is designed to achieve intercandidate DR. A horizontal row of reference pixels, which is read from SRAMs, is stored and shifted downward in Ref. Pels Reg. array. When one candidate is processed, 256 reference pixels are required. When eight horizontally adjacent candidates are processed in parallel, the reference pixels can be horizontally reused. Not (256 × 8) but (256 + 16 × 7) reference pixels are required for eight candidates. Besides, when the ME process is changed to the next row of eight candidates, most data can be reused in Ref. Pels array by vertically



7. Block diagram of the FME engine. There are nine 4 × 4-block PUs to process nine candidates around the refinement center for each reference frame. One 2-D interpolation engine is shared by all 4 × 4-block PUs to achieve DR and local bandwidth reduction.

adjacent candidates. The intercandidate DR can save internal memory bandwidth.

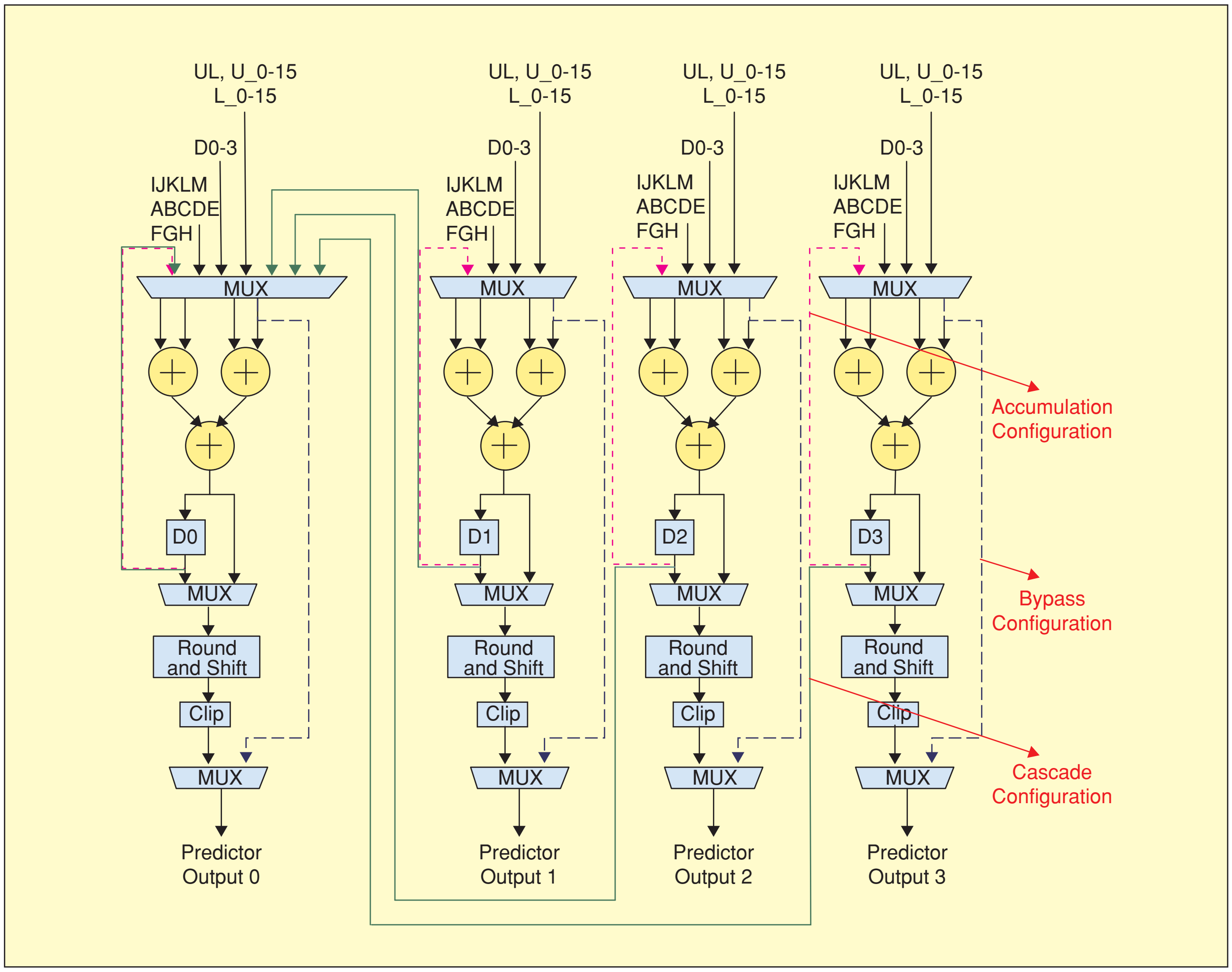
Figure 6 shows the architecture of a PE-array SAD tree. The costs of 16 4 × 4 blocks are separately summed up by 16 2-D subtrees and then reused by one VBS tree for larger blocks. This is intracandidate DR. All 41 SADs for one candidate are simultaneously generated and compared with the 41 best costs. The intracandidate DR can save both computational requirement and internal memory bandwidth.

Ongoing applications of H.264/AVC range from high-definition digital video disc or BluRay for home entertainment to digital video broadcasting for handheld terminals.

Parallel Fractional ME with Lagrangian Mode Decision

The IME searches for the best matches in coarse resolution for variable block sizes and multiple reference frames, while the FME refines these results in fine resolution and decides the best combination of all possible blocks. After the IME, the half-pixel MV refinement is performed around the best integer search positions. The SR of half-pixel MV refinement is ±1/2 pixel along both horizontal and vertical directions. The quarter-pixel MV refinement is then performed around the best half search position with ±1/4 pixel SR. Each half or quarter refinement has nine candidates, including the refinement center and its eight neighbors. The refinement flow will be iteratively processed for all blocks and subblocks in all reference frames.

The main challenge for FME hardware design is to achieve parallel processing under the constraints of sequential FME procedure. For example, the searching center of quarter-resolution refinement depends on the result of half-resolution refinement, and the sequential process is inevitable. The loop of variable block sizes is not suitable to be unrolled because 41 MVs of VBS-ME may point to different positions. The memory bitwidth of SW will become too



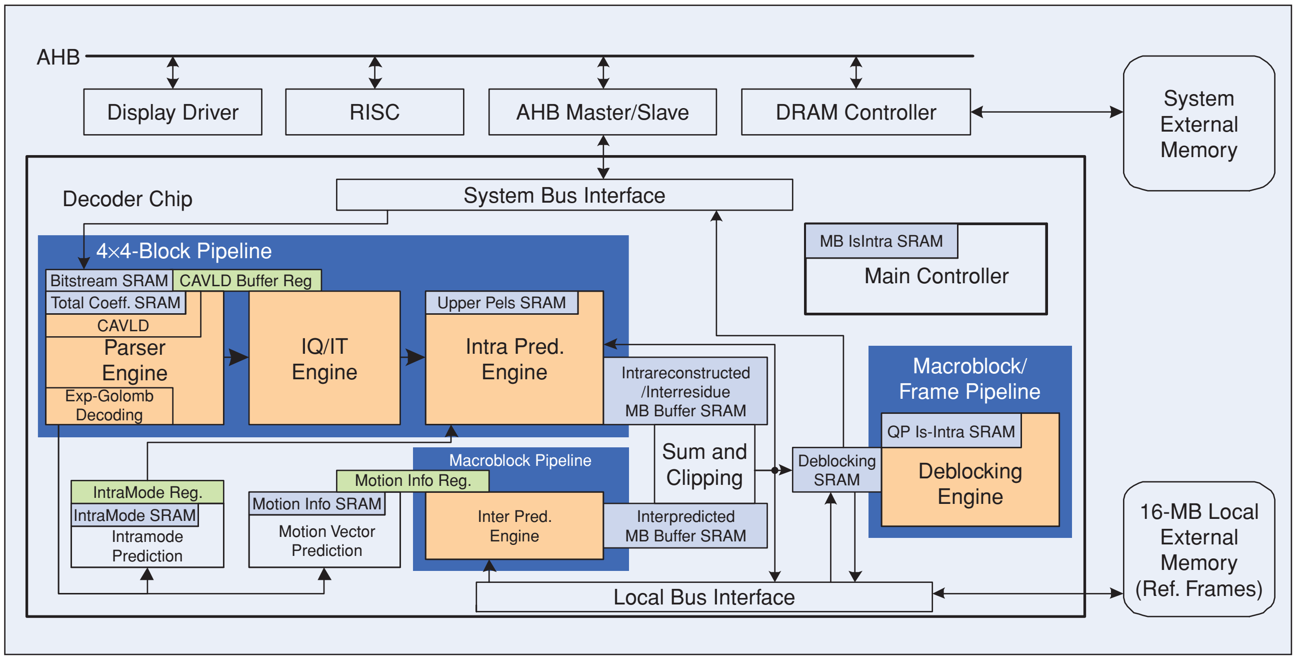
8. Four parallel reconfigurable intrapredictor generator. Four different configurations are designed to support all intraprediction modes in H.264/AVC.

large if the reference pixels of VBS-ME are read in parallel. MV costs of the inevitable sequential processing order among VBS-ME also must be considered.

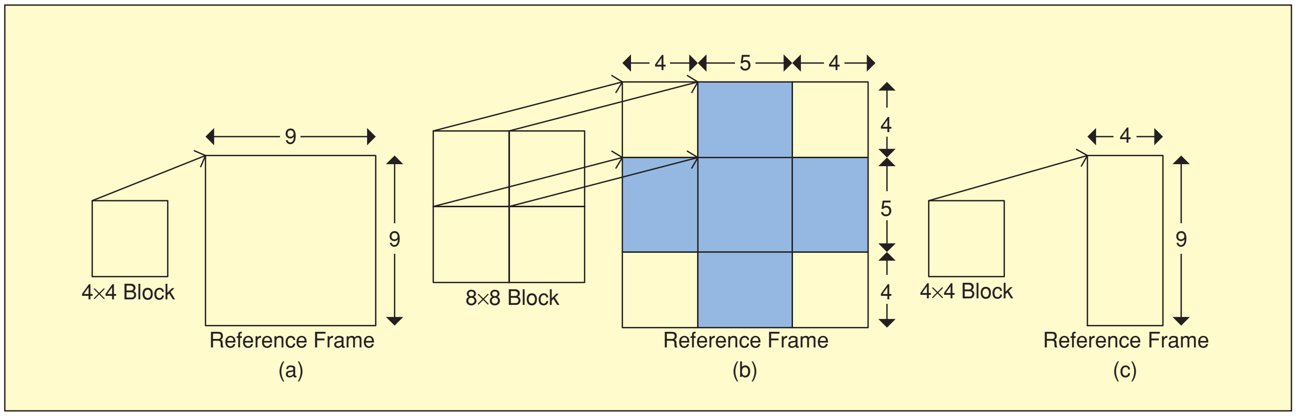
Figure 7 shows the parallel FME architecture [14]. The design concepts are stated as follows. First, the variable blocks range from 16 × 16 to 4 × 4. Therefore, the 4 × 4 block can be the smallest common element. That is, every block and subblock in a MB can be decomposed into several 4 × 4 elements with the same MV. We can concentrate on designing a 4 × 4-block processing unit (PU) to calculate the distortion cost of each 4 × 4 element. Then, the folding technique is applied to reuse the 4 × 4-block PU for larger block sizes. Second, there are nine 4 × 4-block PUs. In each refinement process, nine candidates around the refinement center are processed in parallel. When we interpolate the fractional pixels, most source and intermediate data can be reused by neighboring candidates. The redundant memory access and computation can be saved, which reduces the chip area and on-chip memory bandwidth. As shown in Figure 7, one 2-D interpolation engine is shared by nine 4 × 4-block PUs for each reference frame. Third, each 4 × 4-element PU is arranged with four degrees of parallelism to process four horizontally adjacent pixels in parallel. Most horizontally adjacent integer pixels can be reused for the horizontal filtering to further reduce the on-chip memory bandwidth.

Reconfigurable Intrapredictor Generator

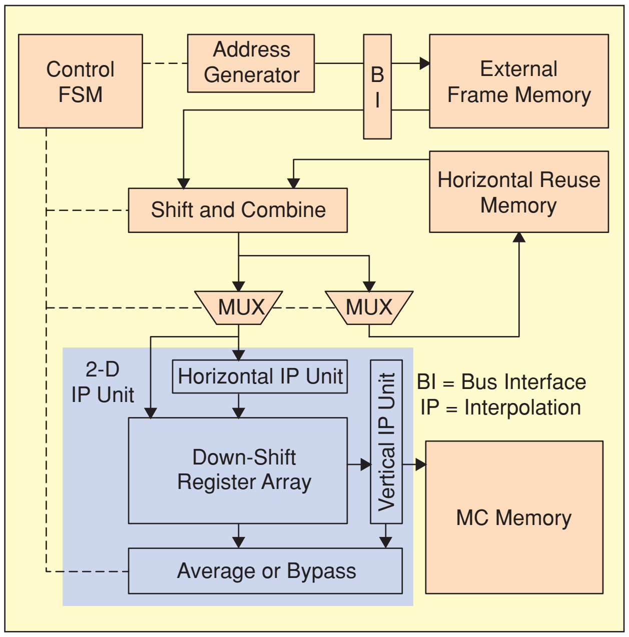
The intraprediction generator supports various prediction modes, which include four I16 MB modes, eight I4 MB modes, and four chroma intra modes. If RISC-based solution is adopted, where the prediction values are generated sequentially for each mode, the required operation frequency will become too high. On the other hand, if the dedicated hardware is adopted, 17



9. Hybrid task pipelining system architecture of an H.264/AVC decoder [16].



10. (a) General case interpolation window; (b) four interpolation windows for an 8 × 8 block (shaded region means reusable); (c) interpolation window when MV is pointing to horizontal integer pixels.



11. Block diagram of the low-bandwidth MC hardware.

kinds of PEs for the 17 modes lead to high hardware costs. Therefore, the reconfigurable circuit with resource sharing for all intraprediction modes is an efficient solution [15].

The hardware architecture of the four-parallel reconfigurable intrapredictor generator is shown in Figure 8. Capital letters (A–H) are the neighboring 4 × 4-block pixels. UL, L0–L15, and U0–U15 denote one bottom-right pixel from the upper-left MB, the 16 pixels of the rightmost column from the left MB, and the 16 pixels of the bottom row from the upper MB, respectively. Four different configurations are designed to support all intraprediction modes in H.264/AVC. The I4 MB/I16 MB horizontal/vertical modes use the bypass data path to select the predictors extended from the block boundaries. Multiple PEs are cascaded to sum up the DC value for I4 MB/I16 MB/chroma DC mode. The normal configuration is used for I4 MB directional modes 3–8. The four PEs select the corresponding pixels multiple times according to the weighted factors and generate four predictors independently. Finally, the recursive configuration is designed for I16 MB plane prediction. The predictors are generated by adding the gradient values to the result of the previous cycles.

Architecture of H.264/AVC Decoding System

The design goals of determining suitable pipelining structure of a H.264/AVC decoder are low area cost and low system bandwidth. The target specification is HDTV1024p 30-f/s videos. The overall system architecture is shown in Figure 9 [16]. The previous designs of video decoders are usually based on MB pipelining structure [17]. This system architecture is based on a hybrid task pipelining structure, including 4 × 4-block-level pipelining, MB-level pipelining, and frame-level pipelining. This is because the 4 × 4 block is the smallest element of one intrapredicted block in H.264/AVC. The transforms and entropy coding are also based on 4 × 4 blocks. Therefore, a 4 × 4-block pipelining structure can be used with the benefit of less area overhead and coding latency. It requires about 1/24 of buffer size compared to the traditional MB pipelining architecture.

Interprediction produces the predicted MB pixels from previously decoded reference frames. As with intraprediction, the basic processing element of interprediction is also a 4 × 4 block. Due to the six-tap finite impulse response filter for interpolation, 9 × 9 integer reference pixels are required for a 4 × 4 block. If a block has the prediction mode larger than 4 × 4, overlapped reference frame pixels of the 4 × 4 blocks can be reused to reduce the system bandwidth. Reference frame DR will be less efficient if inter pred. engine adopts the 4 × 4-block pipelining scheme. Therefore, inter pred. engine should be scheduled to MB-level pipelining with a customized scan order to exploit the reference frame DR. All reference pixels necessary to predict an MB are read from the external memory at once to achieve the lowest memory bandwidth.

Deblocking engine is another special case that does not suit to the 4 × 4-block pipelining scheme. Deblocking engine filters the edges of each 4 × 4 block vertically then horizontally. One 4 × 4 block cannot be completely filtered until its neighboring blocks are reconstructed. This data dependency makes it impractical to fit the deblocking operation into a 4 × 4-block pipelining, since the buffer cannot be efficiently reduced and serious control overhead is required. Therefore, the MB pipelining schedule is adopted. If the decoder has to support flexible macroblock ordering (FMO) and arbitrary slide order (ASO), where the MBs in one frame may not be coded in raster-scan order, the DB unit must be scheduled to frame-level pipelining because the filtering order in one frame can not be violated in MB boundaries.

Low-Bandwidth MC Engine

According to the analysis on system-level design, MC should be scheduled on MB-level pipelining with a customized scan order to exploit the reference frame DR. We first adopt the 4 × 4-based MC. All VBSs are decomposed into several 4 × 4-element blocks and processed sequentially by the 4 × 4-based MC engine with full hardware utilization. The straightforward memory access scheme processes every decomposed 4 × 4-element block independently and always loads 9 × 9 pixels from the external memory for interpolation, as shown in Figure 10(a). The system bandwidth requirement of 4 × 4-based MC can be reduced by two bandwidth reduction techniques [18].

The first technique is interpolation window reuse (IWR). As shown in Figure 10(b), there are overlapped regions between interpolation windows for neighboring 4 × 4-element blocks when the block mode is larger than 4 × 4. The shaded regions represent the reference pixels that can be reused. The second scheme is interpolation window classification (IWC). The interpolation window is not always $(X+5) \times (Y+5)$ for an $X \times Y$ block. As shown in Figure 10(c), a 4 × 4 block with integer MV in the horizontal direction does not require horizontal filtering. Only a 4 × 9 interpolation window is required to be loaded in this situation. In brief, the IWR and IWC schemes aim to precisely control the MC hardware to load an exact interpolation window.

Figure 11 shows the MC hardware architecture. The down shift register array is designed to support vertical IWR. Horizontal reuse memory is designed for horizontal IWR. The IWC is implemented by control FSM and address generator. The shift and combine circuit packs the required integer pixels input from external frame memory and horizontal reuse memory. The 2-D IP unit performs the interpolation, and the compensated MB is buffered in the MC memory. These techniques can provide about 60–80% bandwidth reduction compared with the 4×4-based MC. After this MC machine is integrated into an H.264/AVC HDTV1024p decoder, the total system bandwidth can be reduced 40–50%.

CONCLUSIONS

An efficient hardwired video coding system is composed of the system architecture with appropriate pipelining structure, efficient memory hierarchy, delicate parallelization, and reconfigurable architecture. In this article, we discussed the state-of-the-art hardware architectures for an H.264/AVC video coding core. Many approaches were exploited to improve the hardware efficiency. Five major functional blocks extracted from the H.264/AVC encoding procedure are mapped into four-stage MB pipelining structure to significantly increase the processing capability and hardware utilization. A hybrid-task pipelining scheme, a balanced schedule with block-/MB-/frame-level pipelining was then suggested for the H.264/AVC decoder to greatly reduce the internal memory size. Combined with many bandwidth reduction techniques and DR schemes, these two system architectures are all characterized by low system bandwidth requirements. Moreover, many efficient modules are contributed to support the new H.264/AVC functionality. A parallel IME architecture is designed to dramatically reduce the memory bandwidth. A parallel FME architecture is designed to thoroughly parallelize the rate-distortion optimized mode decision with high hardware utilization. A reconfigurable intrapredictor generator can achieve resource sharing for all intraprediction modes. The bandwidth optimized MC engine exploits DR between interpolation windows of neighboring blocks. These system and module architectures can efficiently support the H.264/AVC video encoding and decoding processes with the HDTV video applications.

REFERENCES

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

[2] T. Wiegand, G.J. Sullivan, G. Bjϕntegaard, 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, July 2003.

[3] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, and T. Wedi, "Video coding with H.264/AVC: Tools, performance, and complexity," *IEEE Circuits Syst. Mag.*, vol. 4, no. 1, pp. 7–28, 2004.

[4] A. Puri, X. Chen, and A. Luthra, "Video coding using the H.264/MPEG-4 AVC compression standard," *Signal Processing: Image Commun.*, vol. 19, no. 9, pp. 793–849, Oct. 2004.

[5] *Joint Video Team Reference Software JM7.3* [Online], Aug. 2003. Available: http://iphome.hhi.de/suehring/tml/download/

[6] Iprof ftp server [Online]. Available: ftp://ftp.lis.e-technik.tu-muenchen.de/pub/iprof/

[7] H.-C. Chang, L.-G. Chen, M.-Y. Hsu, and Y.-C. Chang, "Performance analysis and architecture evaluation of MPEG-4 video codec system," in *Proc. IEEE Int. Symp. Circuits Systems (ISCAS'00)*, 2000, vol. 2, pp. 449–452.

[8] V. Lappalainen, A. Hallapuro, and T. Hamalainen, "Complexity of optimized H.26L video decoder implementation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 717–725, 2003.

[9] M. Takahashi, T. Nishikawa, M. Hamada, T. Takayanagi, H. Arakida, N. Machida, H. Yamamoto, T. Fujiyoshi, Y. Ohashi, O. Yamagishi, T. Samata, A. Asano, T. Terazawa, K. Ohmori, Y. Watanabe, H. Nakamura, S. Minami, T. Kuroda, and T. Furuyama, "A 60-MHz 240-mW MPEG-4 videophone LSI with 16-Mb embedded DRAM," *IEEE J. Solid-State Circuits*, vol. 35, pp. 1713–1721, Nov. 2000.

[10] H. Nakayama, T. Yoshitake, H. Komazaki, Y. Watanabe, H. Araki, K. Morioka, J. Li, L. Peilin, S. Lee, H. Kubosawa, and Y. Otobe, "A MPEG-4 video LSI with an error-resilient codec core based on a fast motion estimation algorithm," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC'02)*, Feb. 2005, vol. 2, pp. 296–512.

[11] T.-C. Chen, Y.-W. Huang, and L.-G. Chen, "Analysis and design of macroblock pipelining for H.264/AVC VLSI architecture," in *Proc. 2004 Int. Symp. Circuits Systems (ISCAS'04)*, 2004, pp. II273–II276.

[12] 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 1.3 TOPS H.264/AVC single-chip encoder for HDTV applications," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC'05)*, 2005, pp. 128–130.

[13] 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. I,* vol. 53, no. 3, pp. 578–593, Mar. 2006.

[14] 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. Acoustics, Speech, Signal Processing (ICASSP'04)*, 2004, pp. V9–V12.

[15] Y.-W. Huang, B.-Y. Hsieh, T.-C. Chen, and L.-G. Chen, "Analysis, fast algorithm, and VLSI architecture design for H.264/AVC intra frame coder," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 3, pp. 378–401, Mar. 2005.

[16] T.-W. Chen, Y.-W. Huang, T.-C. Chen, Y.-H. Chen, C.-Y. Tsai, and L.-G. Chen, "Architecture design of H.264/AVC decoder with hybrid task pipelining for high definition videos," in *Proc. 2005 Int. Symp. Circuits Systems (ISCAS 2005)*, 2005, pp. 2931–2934.

[17] H.-Y. Kang, K.-A. Jeong, J.-Y. Bae, Y.-S. Lee, and S.-H. Lee, "MPEG4 AVC/H.264 decoder with scalable bus architecture and dual memory controller," in *Proc. Int. Symp. Circuits Systems (ISCAS'04)*, 2004, vol. 2, pp. II-145–148.

[18] C.-Y. Tsai, T.-C. Chen, T.-W. Chen, and L.-G. Chen, "Bandwidth optimized motion compensation hardware design for H.264/AVC HDTV decoder," in *Proc. 2005 Int. Midwest Symp. Circuit Systems (MWSCAS'05)*, 2005, pp. 1199–1202.

Tung-Chien Chen, Hung-Chi Fang, Chung-Jr Lian, Chen-Han Tsai, Yu-Wen Huang, To-Wei Chen, Ching-Yeh Chen, Yu-Han Chen, Chuan-Yung Tsai, and *Liang-Gee Chen* are with the DSP/IC Design Lab, Department of Electrical Engineering and Graduate Institute of Electronics Engineering, National Taiwan University. E-mail: lgchen @video.ee.ntu.edu.tw. CD ■