

Analysis and Architecture Design of an HDTV720p30 Frames/s H.264/AVC Encoder

Priyanka Bhagwat

Dept. of Electronics & Telecommunication,
Savitribai Phule Pune University, India,

Abstract

H.264/AVC significantly outperforms previous video coding standards with many new coding tools. However, the better performance comes at the price of the extraordinarily huge computational complexity and memory access requirement, which makes it difficult to design a hardwired encoder for real-time applications. In addition, 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 constrained to be employed. The hardware utilization and throughput are also decreased because of the block/MB/frame-level reconstruction loops. In this paper, we describe our techniques to design the H.264/AVC video encoder for HDTV applications. On the system design level, in consideration of the characteristics of the key components

and the reconstruction loops, the four-stage macroblock pipelined system architecture is first proposed with an efficient scheduling and memory hierarchy. On the module design level, the design considerations of the significant modules are addressed followed by the hardware architectures, including low-bandwidth integer motion estimation, parallel fractional motion estimation, reconfigurable intrapredictor generator, dual-buffer block-pipelined entropy coder, and deblocking filter. With these techniques, the prototype chip of the efficient H.264/AVC encoder is implemented with 922.8 K logic gates and 34.72-KB SRAM at 108-MHz operation frequency.

Keyword : ISO/IEC 14496-10 AVC, ITU-T Rec. H.264, joint Video Team (JVT), single-chip video encoder, very large-scale integration (VLSI) architecture.

I. INTRODUCTION

The ITU-T Video Coding Experts Group (VCEG) and ISO/IEC 14 496-10 AVC Moving Picture Experts Group (MPEG) formed the Joint Video Team (JVT) in 2001 to develop the new video coding standard, H.264/Advanced Video Coding (AVC) [1]. Compared with MPEG-4 [2], H.263, and MPEG-2 [3], H.264/AVC can achieve 39%, 49%, and 64% of bit-rate reduction, respectively [4]. The high compression performance comes mainly from the new prediction techniques that remove spatial and temporal redundancies. To remove spatial redundancy, H.264/AVC intra prediction supports many prediction modes to make better prediction. Inter prediction is enhanced by motion estimation (ME) with quarter-pixel accuracy, variable block sizes (VBS), and multiple reference frames (MRF) to remove more temporal redundancy. Moreover, the advanced entropy coding tools [9] use content adaptivity to reduce more statistic redundancy. The perceptual quality is improved by in-loop deblocking filter. For more details, interested readers can refer to [5] for a quick and thorough study. While highly interactive and

recreational multimedia applications appear much faster in the future, it demands much higher compression ratio and quality for video contents. H.264/AVC undoubtedly plays an important role in this area. On-going applications range from High Definition Digital Video Disc (HDDVD) or BluRay for home entertainments with large screens to Digital Video Broadcasting for Handheld terminals (DVB-H) with small screens. However, the H.264/AVC coding performance comes at the price of computational complexity. According to the instruction profiling with HDTV specification, H.264/AVC encoding process requires 3600 giga-instructions per second (GIPS) computation and 5570 giga-bytes per second (GBytes/s) memory access. For real-time applications, the acceleration by a dedicated hardware is a must. However, it is difficult to design the architecture for the H.264/AVC hardwired encoder. The architecture design for the significant modules are also very challenging. Besides high computational complexity and memory access, the coding path is very long, which includes intra/inter prediction,

block/macroblock/frame-level reconstruction loops, entropy coding, and in-loop deblocking filter. The reference software [6] adopts many sequential processing of each block in the macroblock (MB), which restricts the parallel architecture design for hardware. The block-level reconstruction loop caused by intra prediction induces the bubble cycles and decreases the hardware utilization and throughput. Some coding tools have multiplex modes, and a larger gate count is required if the multiple processing elements (PEs) are separately designed for different modes without any resource sharing and data reuse. Some coding tools involve many data dependencies to enhance the coding performance, and a considerable storage space is required to store the correlated data during the encoding process. To overcome these difficulties,

II. ALGORITHM ANALYSIS AND DESIGN SPACE EXPLORATION

Our highest specification is HDTV720p (1280 720, 30 fps) video encoder for H.264/AVC baseline profile. In this section, we will first describe the instruction profiling. Then, the design considerations will be shown by the algorithm exploration. Finally, the previous works will be briefly reviewed followed by the problem definition.

A. Instruction Profiling

We exploit instruction profiling to show the computational complexity and memory access of H.264/AVC. The iprof, a software analyzer on the instruction level, is used to profile an H.264/AVC encoder on a processor-based platform (SunBlade 2000 workstation, 1.015 GHz Ultra Sparc II CPU, 8 GB RAM, Solaris 8). To focus on the target specification, a software C model is developed by extracting all baseline profile compression tools from the reference software [7]. The instructions are divided into three categories—computing, controlling, and memory access. The computing instructions are composed of arithmetic and logic ones. The controlling instructions contain jump, branch, and compare ones, while the memory access instructions denote the data transfer ones such as load and store. Table I shows the result of instruction profiling. The encoding parameters are CIF, 30 frames/s, five reference frames, 16-pel search range, , and low complexity mode decision. According to the profiling result, the encoding complexity of H.264/AVC baseline profile is about ten times more

many hardware design techniques are described for H.264/AVC video coding system in this paper. There are two critical issues to be addressed. First, for H.264/AVC encoder, the traditional two-stage MB pipelining cannot be efficiently applied because of the long critical path and feedback loop. According to our analysis, five major functions are extracted and mapped into the four-stage MB pipelined structure with suitable task scheduling. Second, the design considerations and optimizations for the significant modules, including low-bandwidth integer ME (IME), [8] parallel fractional ME (FME), reconfigurable intra predictor generator, dual-buffer block-pipelined context-based adaptive variable length coding (CAVLC) engine, and in-loop deblocking filter, are discussed.

complex than MPEG-4 simple profile [9]. This is mainly due to MRF-ME and VBS-ME in inter prediction. For the full search (FS) algorithm, the complexity of IME is proportional to the number of reference frames, while that of FME is proportional to the MB number constructed by variable blocks and the number of reference frames. Our focused design case is targeted at SDTV (720 480, 30 fps)/HDTV720p videos with four/one reference frame and maximum search range (SR) of . The computational complexity and memory access for SDTV/HDTV720p are 2470/3600 GIPS and 3800/5570 GBytes/s. The huge computational loads are far beyond the capability of today's general purpose processors (GPPs)[10]. Therefore, a dedicated hardware is essential for real-time applications.

B. Design Space Exploration

The major design challenges of an H.264/AVC hardware encoder are analyzed as follows.

- *Computational complexity and bandwidth requirement:*

According to the profiling, H.264/AVC requires much more computational complexity than the previous coding standards. This will greatly increase the hardware cost, especially for the HDTV applications. For hardware implementation, highly utilized parallel architectures with hardware-oriented encoding algorithm are required. The bandwidth requirement of H.264/AVC encoding system is also much higher than those of the previous coding standards[11]. For example, the MRF-ME

contributes the most heavy traffic for loading reference pixels. Neighboring reconstructed pixels are required by intra prediction and deblocking filter. 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 bandwidth.

- *Sequential flow:*

The H.264/AVC reference software adopts many sequential processes to enhance the compression performance. It is hard to efficiently map the sequential algorithm to parallel hardware architecture. For the system architecture, the coding path is very long, which includes intra/inter prediction, block/macroblock/frame-level reconstruction loops, entropy coding, and in-loop deblocking filter. The sequential encoding process should be partitioned into several tasks and processed MB by MB in pipelined structure, which improves the hardware utilization and the throughput. For the module architecture, the problem of sequential algorithms is critical for ME since it is the most computationally intensive part and requires the most degrees of parallelism. The FME must be done after the IME. In addition, 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. The MV of each block is generally predicted by the left, top, and top-right neighboring blocks. The cost function can be computed only after the prediction modes of the neighboring blocks are determined, which also causes inevitable sequential processing. The modified hardware-oriented algorithms are required to enable parallel processing without noticeable quality drop. The analysis of processing loops and data dependencies is also helpful to map the sequential flow into the parallel hardware.

- *Reconstruction loops:*

In traditional video coding standards, there is only a frame-level reconstruction loop generating the reference frames for ME and motion compensation (MC). In H.264/AVC, the intra prediction requires the reconstructed pixels of the left and top[12] neighboring blocks, which induces the MB-level and

block-level reconstruction loops. For the MB-level reconstruction loop, the reconstructed pixels of MB-a, MB-b, and MB-c are used to predict the pixels in MB-x for Intra 16 16 MB mode (I16 MB). Not until MB-a, MB-b, and MB-c are reconstructed can MB-x be predicted. In order to support Intra 4 4 MB mode (I4 MB), not until 4 4-intra mode of B-a, B-b, B-c, and B-d are decided and reconstructed can B-x be processed. The reconstructed latency is harmful for hardware utilization and throughput if the intra prediction and reconstruction are not jointly considered and scheduled.

- *Data dependency:*

The new coding tools improve the compression performance with many data dependencies. The frame-level data dependencies contribute 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. In addition, since many data and coding information may be required by the following encoding processes, the storage space of both off-chip memory and on-chip buffer are largely increased. In order to reduce the chip cost, the functional period, or life-time, of these data must be carefully considered with the system architecture and the processing schedule.

- *Abundant modes:*

Many coding tools of H.264/AVC that have multiplex modes. For example, there are 17 different modes for intra prediction while 259 kinds of partitions for inter prediction. Six kinds of 2-D transform, 4 4/2 2 DCT/IDCT/Hadamard transform, are involved in reconstruction loops. The reconfigurable processing engines and the reusable prediction cores are important to efficiently support all these functions.

II. PROPOSED H.264/AVC ENCODING SYSTEM

Since the traditional two-stage MB pipelining cannot be efficiently applied to H.264/AVC, in this section, five major functions are extracted and mapped[13] into the four-stage MB pipelining with a suitable task scheduling in the proposed encoding system. Furthermore, the design considerations and optimizations for the significant modules are described to complete the whole system. With these

techniques, the efficient implementation for an H.264/AVC encoding system can be achieved.

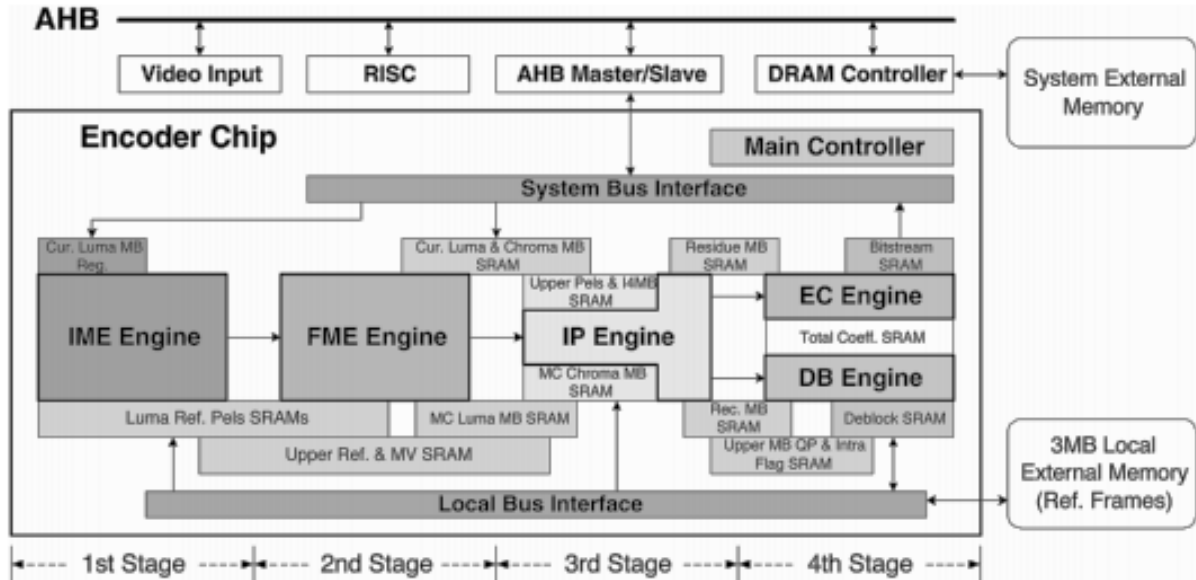


Fig. 1. Block diagram of the proposed H.264/AVC encoding system. Five major tasks, including IME, FME, IP, EC, and DB, are partitioned from the sequential encoding procedure and processed MB by MB in a pipelined structure.

A. Proposed Four-Stage Macroblock Pipelining

The proposed system architecture is shown in Fig. 1. Five major tasks, including IME, FME, intra prediction 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. Several issues of designing this system pipelining are described as follows. The prediction, which is ME only in the previous standards, includes IME, FME, and intra prediction in H.264/AVC. Because of the diversity of the algorithms and the difference in computational complexity, it is difficult to implement IME, FME, and intra prediction with the same hardware. Putting IME, FME, and intra prediction in the same MB pipelined stage leads to very low utilization. Even if the resource sharing is achieved, the operating frequency becomes too high due to the sequentially processing. Therefore, FME is firstly pipelined MB by MB after IME to double the throughput. As for intra prediction, because of the MB-level and the block-level reconstruction loops, it cannot be separated from the reconstruction engine. In addition, the reconstruction process should be separated from ME and pipelined MB by

MB to achieve highest hardware utilization[14]. Therefore, the hardware engines of intra prediction together with forward/inverse transform/quantization should be located in the same stage, IP stage. In this way, the MB-level and the block-level reconstruction loops can be isolated in this pipeline stage. The EC encodes MB headers and residues after transformation and quantization. The DB generates the standard-compliant reference frames 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 external memory for the ME of the next current frame, which constructs the frame-level reconstruction loop. Please note that, the luma MC is placed in the FME stage to reuse Luma Ref. Pels SRAMs and interpolation circuits. The compensated MB is transmitted to IP stage to generate the residues after intra/inter mode decision. The chroma MC is implemented in IP stage since it can be executed after intra/inter mode decision. In summary, five main functions extracted from the coding process are mapped into the four-stage MB pipelined structure. The processing cycles of the four stages are balanced with different degrees of parallelism to achieve high utilization. MBs within one frame are

coded in raster order with the schedule shown in Fig. 2. One horizontal column indicates the MBs with different tasks that are processed in parallel.

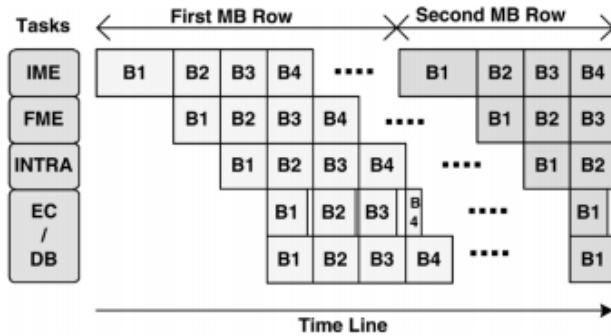


Fig.2. MB schedule of four-stage MB pipelining

B. Low-Bandwidth Parallel Integer Motion Estimation

IME requires the most computational complexity and memory bandwidth in H.264/AVC. A large degree of parallelism is required for SDTV/HDTV specifications. However, the sequential Lagrangian mode decision flow make it impossible to design the parallel architecture for IME. Therefore, techniques on algorithmic and architectural levels are used to enable parallel processing and to reduce the required hardware resources. In addition, efficient memory hierarchy and data reuse schemes are jointly applied to greatly reduce the memory bandwidth requirement.

1. **Hardware-Oriented Algorithm:** The MV of each block is generally predicted by the medium values of MVs from the left, top, and top right neighboring blocks. The rate term of the Lagrangian cost function can be computed only after the MVs of the neighboring blocks are determined, which causes an inevitable sequential processing. That is, blocks and subblocks in a MB cannot be processed in parallel. Moreover, when a MB is processed at the IME stage, its previous MB is still in the FME stage. The MB mode and the best MVs of the left blocks cannot be obtained in the four-stage MB pipelined architecture [15]. To solve these problems, the modified MVP is applied for all 41 blocks in the MB, as shown in Fig. 4. The exact MVPs of variable blocks, which are the medium of MVs of the top-left, top, and top-right blocks, are changed to the medium of MVs of the top-left, top, and top-right MBs. For example, the exact MVP 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 in order to facilitate the parallel processing and the MB pipelining.

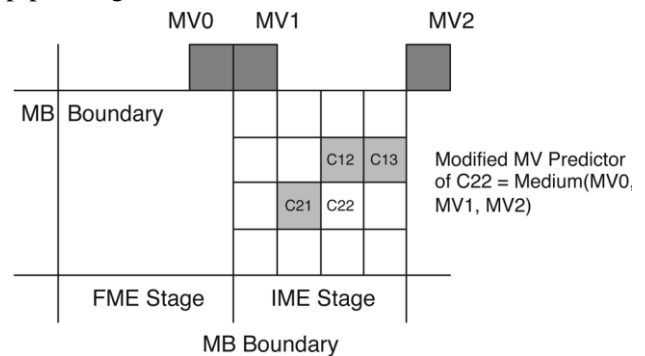


Fig. 3. Modified MVPs. In order to facilitate the parallel processing and the MB pipelining, the MVPs of all 41 blocks are changed to the medium of MV0, MV1, and MV2.

2) Architectures Design of IME:

Fig. 4 shows the low-bandwidth parallel IME architecture, which mainly comprises eight *PE-Array SAD Trees*. The current MB (CMB) is stored in *Cur.MB Reg.* The reference pixels are read from external memory and stored in *Luma Ref. Pels SRAMs*. Each PE array and its corresponding 2-D SAD tree compute the 41 SADs of VBS for one searching candidate at each cycle. Therefore, eight horizontally adjacent candidates are processed in parallel. All SAD results of VBS are input to the *Comparator Tree Array*. Each comparator tree finds the smallest SAD among the eight search points and updates the best MV for a certain block-size. Because SWs of neighboring current MBs are considerably overlapped, and so are the pixels of neighboring candidate blocks, a three-level memory hierarchy, including external memory, *Luma Ref. Pels SRAMs*, and *Ref. Pels Reg. Array*, is used to reduce bandwidth requirement by data reuse (DR). Three kinds of DR are implemented—MB-level DR, inter-candidate DR, and intra-candidate DR. The *Luma Ref. Pels SRAMs* are firstly embedded to achieve MB-level DR. When ME process is changed from one CMB to another CMB, there is the overlapped area between neighboring SWs. Therefore, the reference pixels of the overlapped area can be reused, and only a part of SW must be loaded from system memory. The system bandwidth can thus be reduced [16]. The *Ref.*

PelsReg. Array acts as the temporal buffer between *PE-Array 2-DSAD Tree* and *Luma Ref. Pels SRAMs*. It is designed to achieve inter-candidate DR. Fig. 6 shows the M-parallel *PE-arraySAD Tree* architecture. A horizontal row of reference pixels, which are 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, not (256 8) but (256 16 7) reference pixels are required. In addition, when the ME process is changed to the next eight candidates, most data can be reused in *Ref. Pels Array*. The proposed parallel architecture achieves inter-candidate DR in both horizontal and vertical directions and reduce the on-chip SRAM bandwidth.

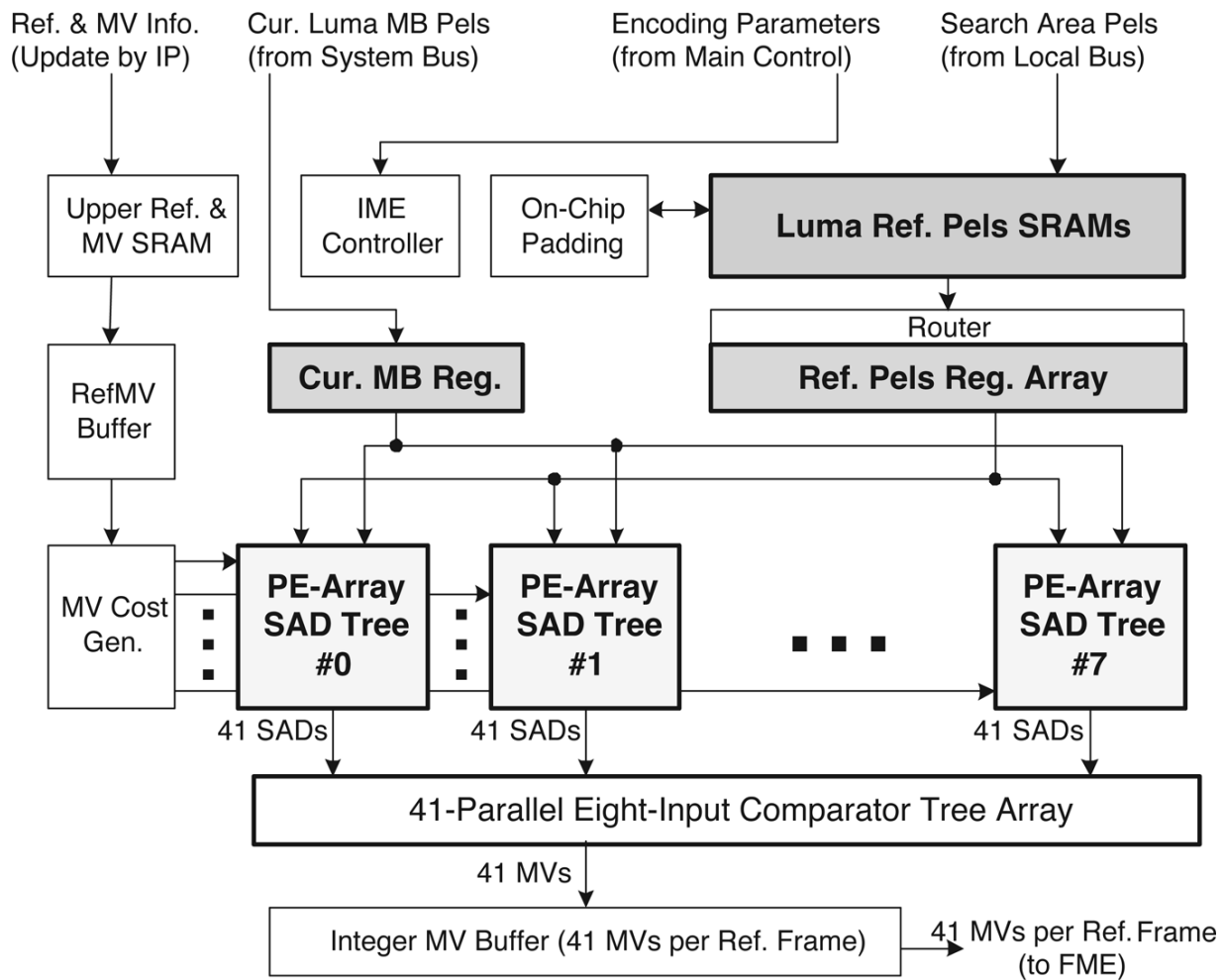


Fig.4. Block diagram of the low-bandwidth parallel IME engine. It mainly comprises eight *PE-Array SAD Tree*, and eight horizontally adjacent candidates are processed in parallel.

III. EXPERIMENTAL RESULTS

A. Implementation Results of H.264/AVC SDTV/HDTV720pEncoder

The specification of the proposed H.264/AVC encoder is baseline profile with level up to 3.1. The maximum computational capability is to real-time encode SDTV 30 fps video with four reference frames or HDTV720p 30 fps video with one reference frame.

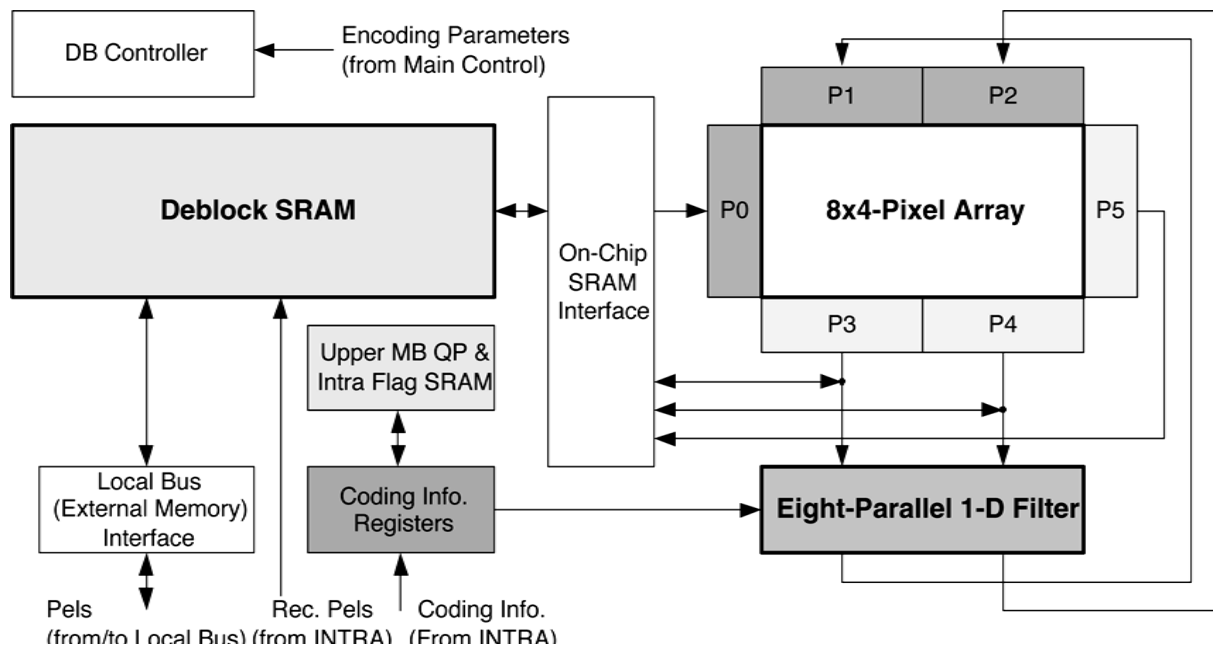


Fig 5 :Block diagram of the DB engine.

Functional Block	Gate Counts	Memory (KB)
IME Module	305211	13.71
FME Module	401885	13.82
IP Module	121012	5.01
EC Module	29332	1.27
DB Module	20152	0.91
Others	45176	0.00
Total	922768	34.72

Table 1: HARDWARE COST OF H.264/AVC ENCODER

	[21] CASII-2004	[22] ASPDAC-2005	[36] ISCAS-2003	Proposed
# of PE	16	256	256	128x8
Process	0.13 μ m	0.18 μ m	0.35 μ m	0.18 μ m
Gate Count	61 k	154 k	106 k	305 k
Frequency	294 MHz	100 MHz	66.67 MHz	81/108 MHz
Max. Spec.	4CIF	4CIF(15fps)	SDTV	SDTV/HDTV
Max SR	32x32	64x64	48x32	128x64
Max Ref.	1	1	1	4/1
GCPP	3812	601	412	298

Table2: COMPARISON OF THE H.264/AVC IME ARCHITECTURES

	Proposed [34] ICME-2003	[37] ISCAS-2005	[38] ICIP-200
Process	0.25/0.18 μ m	0.25 μ m	0.18 μ m
Gate Count	18.91/20.15 k	18.77 k	22.5 k
Frequency	100/120 MHz	100 MHz	100 MH
Filter Cycle/MB	440/440	268	243

Table 3: COMPARISON BETWEEN THE PROPOSED ARCHITECTURE AND THE NEWEST ONES

Table II shows the logic gate count profile synthesized at 120 MHz. The total logic gate counts is about 922.8 K. The prediction engines, including IME, FME, and IP stages, dominate 90% logic area. As for on-chip SRAM, 34.88 KB are required. The chip is fabricated with UMC 0.18- m 1P6M CMOS process. Fig. shows the die micrograph. The core size is 7.68 4.13 mm . The power consumption is 581 mWfor SDTV videos and 785 mW for HDTV720p videos at 1.8-V supply voltage with 81/108 MHz operating frequency. The detailed chip features are shown in

Table III. The encoded video quality of our chip is competitive with that of reference software, in which FS is implemented with Lagrangianmode decision. As shown in Fig. 19, with improvement of the Lagrangian multipliers, our compression performance is even better at high bitrate.

V. CONCLUSION

In this paper, an H.264/AVC baseline profile single-chip encoder with the silicon core size of 7.68 4.13 mm and 0.18- m CMOS technology is presented. A four-stage macroblock(MB) pipelined architecture can encode HDTV720p 30fps videos in real time at 108 MHz. The new pipelined architecture doubles the throughput of the conventional two-stage MB pipelined architecture with high hardware utilization for H.264/AVC. The encoder contains

five engines of intermotion estimation (IME), fractional motion estimation (FME), intra prediction with reconstruction loops (IP), entropy coding (EC), and in-loop deblocking filter (DB). For IME, a parallel array of eight SAD trees is designed with three-level memory hierarchy and data reuse (DR). For FME, a loop decomposition method is provided to obtain an efficient mapping from the algorithm to the architecture with a regular flow. For IP, the reconfigurable intra predictor generators are adopted. For EC, a dual-buffer block-pipelined

CAVLC module can double the throughput and utilization. For DB, an advanced filtering scheduling is proposed to reduce 50% on-chip memory bandwidth. In summary, parallel processing and pipelining techniques are used to reduce the frequency and increase the utilization, while folding and reconfigurable techniques are applied to reduce the area. With these techniques, the first single-chip H.264/AVC encoder is efficiently implemented with full search quality for HDTV applications.

REFERENCES

- [1] *Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification*, Joint Video Team, ITU-T Recommendation H.264 and ISO/IEC 14496-10 AVC, May 2003.
- [2] *Information Technology—Coding of Audio-Visual Objects—Part 2: Visual*, ISO/IEC 14496-2, 1999.
- [3] *Video Coding for Low Bit Rate Communication*, ITU-T Recommendation H.263, Feb. 1998.
- [4] *Information Technology—Generic Coding of Moving Pictures and Associated Audio Information: Video*, ISO/IEC 13818-2 and ITU-T Rec.H.262, 1996.
- [5] A. Joch, F. Kossentini, H. Schwarz, T. Wiegand, and G. J. Sullivan, "Performance comparison of video coding standards using Lagrangian coder control," in *Proc. IEEE Int. Conf. Image Processing (ICIP'02)*, 2002, pp. 501–504.
- [6] T. Wedi and H. G. Musmann, "Motion- and aliasing-compensated prediction for hybrid video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 577–586, Jul. 2003.
- [7] T. Wiegand and B. Girod, *Multi-Frame Motion-Compensated Prediction for Video Transmission*. Boston, MA: Kluwer Academic, 2002.
- [8] T. Wiegand, X. Zhang, and B. Girod, "Long-term memory motion compensated prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 1, pp. 70–84, Feb. 1999.
- [9] D. Marpe, H. Schwarz, and T. Wiegand, "Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 620–636, Jul. 2003.
- [10] P. List, A. Joch, J. Lainema, G. Bjøntegaard, and M. Karczewicz, "Adaptive deblocking filter," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 614–619, Jul. 2003.
- [11] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.
- [12] 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, 1Q, 2004.
- [13] A. Puri, X. Chen, and A. Luthra, "Video coding using the H.264/MPEG-4 AVC compression standard," in *Signal Process.: Image Commun.*, Oct. 2004, vol. 19, no. 9, pp. 793–849.
- [14] *Joint Video Team Reference Software JM7.3*, ITU-T, Aug. 2003. [Online]. Available: <http://bs.hhi.de/suehring/tml/download/>
- [15] Iprof ftp server. [Online]. Available: <http://iphome.hhi.de/suehring/tml/download>
- [16] 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 and Systems (ISCAS'00)*, May 2000, vol. 2, pp. 449–452.