

Architecture Design of Low Power Integer Motion Estimation for H.264/AVC

Tung-Chien Chen, Yu-Han Chen, Sung-Fang Tsai and Liang-Gee Chen

DSP/IC Design Lab., Graduate Institute of Electronics Engineering and Department of Electrical Engineering, National Taiwan University; Email: djchen, doliamo, bigmac, lgchen@video.ee.ntu.edu.tw

Abstract—In motion estimation, fast algorithms usually lead to an irregular searching flow, and the power reduction on architecture level is constrained for poor data reuse (DR). In this paper, a parallel IME hardware for H.264/AVC is proposed to well combine the techniques on algorithm and architecture levels. The “2-D SAD Tree” is adopted to support intra- and inter-candidate DR for the content-adaptive parallel-VBS four step search algorithm. A ladder-shaped reference data arrangement is proposed to support DR in both horizontal and vertical directions, while an advanced searching flow is applied to reduce the latency cycles. After these two techniques, 77.6% power of search window SRAMs can be reduced. According to the implementation result, in ultra low power mode, only 1.424 mW is required for realtime encoding CIF 30fps videos with 13.5 MHz operation frequency.

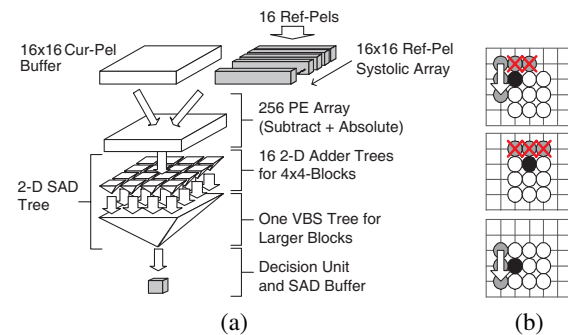


Fig. 1. (a) 2-D SAD tree architecture [5] supporting both FS and FSS; (b) Data reuse problem for FSS.

I. INTRODUCTION

The new video coding standard of H.264/AVC [1] significantly outperforms the previous standards in compression performance. However, the new techniques of variable block sizes (VBS) and multiple reference frames (MRF) [2] in integer motion estimation (IME) contributes huge computation complexity, and hardware acceleration is a must. For VLSI implementation, low power design methodology is a key factor for limited availability of power in portable or wearable devices.

According to our analysis, the optimal low power IME engine should be the parallel architecture that supports the fast algorithm with efficient data reuse (DR). For H.264/AVC IME, several hardware architectures have been proposed [3] [4] [5], but none of them has low power considerations. For previous standards, several low power IMEs architectures [6] [7] [8] with corresponding hardware-oriented fast algorithms were designed. However, the VBS, MRF, and many other functionalities are not involved. Besides, because of the irregular search pattern for fast algorithms, the power reduction on algorithm level usually constrains that on architecture level for poor DR. Obviously, the new architecture of low power IME architecture is urgently demanded in H.264/AVC system especially for portable devices. Besides, the advanced techniques are required to efficiently combine the architecture and fast algorithm.

In this paper, a low-power IME design for H.264/AVC is proposed. The rest of this paper is organized as follows. In Section II, the power reduction techniques will be described followed by problem definition. In Section III, the new and advanced techniques are proposed to efficiently reuse data, and the low power architecture is designed. The implementation result and the comparison are shown in Section IV. Finally, Section V gives a conclusion.

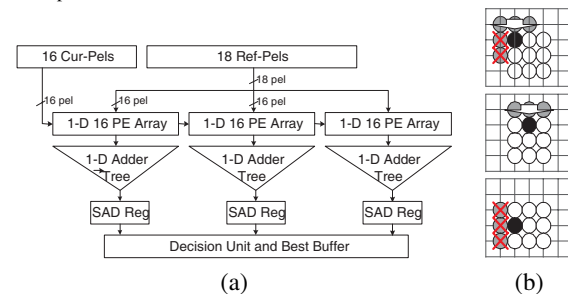


Fig. 2. (a) Parallel 1-D tree architecture [8] supporting both FS and FSS; (b) Data reuse problem for FSS.

II. POWER REDUCTION TECHNIQUES AND PROBLEM DEFINITION

In IME, in order to find the best matched candidate, a search window (SW) within one reference frame has to be searched. Because SWs of neighboring current macroblocks (CMBs) are considerably overlapped, the SW SRAMs are embedded to reduce system bandwidth by DR. In this memory hierarchy, the power consumption of IME core mainly comes from two parts. One is the data access power of reference pixels in SW SRAMs. The other is computation power of matching cost calculation. Two techniques are investigated to reduce power consumption. First, the hardware-oriented fast algorithms can reduce the computation power of logic circuits and the transmission power of SW SRAMs. Second, because pixels of neighboring candidate blocks are overlapped, the systolic register array and parallel ME core are designed to achieve DR. The transmission power of SW SRAMs can be further reduced. The later is important because SRAMs usually consumes much more power than logic circuits.

Traditional fast algorithms cannot efficiently support VBSME for H.264/AVC. Besides, most of them are not suitable for hard-

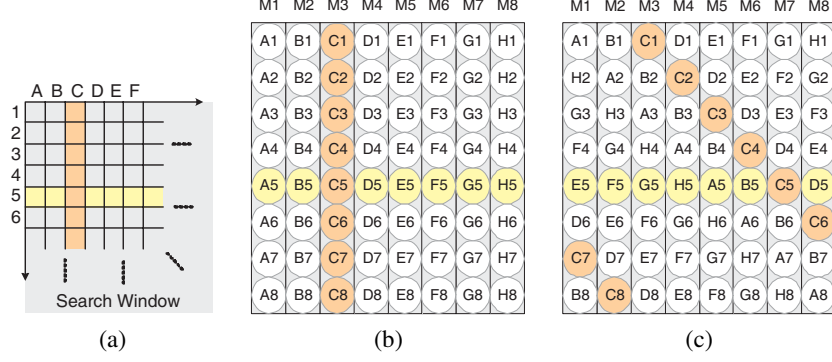


Fig. 3. (a) Physical location of SW; (b) Traditional interleaving SW data arrangement supporting 1-D random access; (c) Proposed ladder-shaped SW data arrangement supporting 2-D random access.

ware implementation. The problem has been addressed and analyzed in [9], and the content-adaptive parallel-VBS four stage search (FSS) was proposed. Three key issues are involved. First, the FSS [10] has the rectangular search pattern, and is more suitable for inter-candidate DR. Second, the intra-candidate DR is achieved by parallel-VBS scheme. We use only the 16×16 block during the iteration process. All distortion costs of the smallest 4×4 blocks are computed firstly, and the larger VBS are on-line generated by reusing these 4×4 costs. Third, in order to provide robust coding efficiency for VBSME, more initial search centers are generated from MVs of neighboring blocks. The content adaptive scheme used to expand or shrink these initial candidates are applied to achieve good trade-off between computation complexity and compression performance. In this paper, we will use this algorithm and focus on the low power techniques on architecture level.

Most of previous IME architectures supporting fast algorithms cannot support DR as efficiently as those for FS. Here comes two examples of FSS. Please note that, for simplification, the interval of square pattern is defined as one pixel for FSS in this paper. Figure 1 (a) shows the “2-D SAD Tree” architecture [5] that supports both FS and FSS. The CMB is stored in “ 16×16 Cur-Pel Buffer”. A row of 16 reference pixels is inputted and shifted downward in “ 16×16 Ref-Pel Systolic Array”. The inter-candidate DR can be achieved between vertically adjacent candidates. Residues are generated in “256-PE Array”, and then summed up by “2-D SAD Tree”. For FS algorithm, after latency of 15 cycles, this architecture can process one candidate each cycle, and each candidate requires 16 reference pixels in average. For FSS algorithm, the reference pixels can be reused only for vertically adjacent candidates, as shown in Fig. 1 (b). It requires 256 reference pixels for the rest candidates. Therefore, $1856 (16 \times 4 + 256 \times 7)$ pixels are required for 11 candidates shown in 1 (b). In average, 169 reference pixels are required for each candidate. Besides, the hardware utilization and throughput will largely decrease for the latency cycles.

Figure 2 (a) shows another “Parallel 1-D Tree” architecture [8] that is also developed for FS and FSS algorithm. 18 reference pixels and 16 CMB pixels are broadcasted to the three “1-D 16 PE Arrays”. 16 cycles are required to process three horizontally adjacent candidates in parallel. For FS algorithm, the SW data can be reused by these three candidates, and the 101

($19 \times 16 / 3$) pixels are required for each candidate. For FSS algorithm, there is DR problem for vertically adjacent candidates, as shown in Fig. 2 (b). 1886 ($101 \times 6 + 256 \times 5$) pixels are required for 11 candidates. In average, 171 reference pixels are required for each candidate.

Because an irregular searching path for fast algorithms is usually followed by poor DR, the power reduction on algorithm level usually constrains that on architecture level. The optimal low power IME engine should be the parallel architecture that supports the fast algorithm with efficient candidate-level DR, which is the key innovation of this paper.

III. LOW POWER ARCHITECTURE DESIGN

In this session, the parallel architecture supporting the proposed content-adaptive parallel-VBS FSS algorithm will be developed with “FS-like” DR rate.

A. Proposed Techniques for Candidate-level DR

We start from the “2-D Adder Tree” rather than the “Parallel 1-D Tree” as the basic architecture. Three reasons are stated as follows. First, because of the systolic array structure with larger degrees of parallelism, the “2-D Adder Tree” architecture potentially has better DR capability. Second, the “1-D Tree” architecture is usually co-work with partial distortion elimination (PDE) algorithm [11] that can terminate the unnecessary computation by comparing the partial and minimum SAD costs. However, to support the parallel-VBS, the costs of 4×4 -blocks are reused for larger blocks. The PDE cannot be efficiently applied in this situation. Third, the “2-D Adder Tree” architecture can support intra-candidate DR without partial SAD registers. This hardware overhead is largely required by the “Parallel 1-D Tree”.

As for inter-candidate DR problem for fast algorithms, it mainly comes from the access restriction of SW SRAMs. Figure 3 (a) shows the physical location of the reference pixels in SW. In tradition, the horizontally adjacent pixels are interleavingly arranged in different SW SRAMs. As shown in Fig. 3 (b), the first column, “A1–A8”, of reference pixels is placed in the memory “M1”. The second column, “B1–B8”, is placed in the memory “M2”, and so on. If there are eight memories, the ninth column is placed in the next bank of memory “M1”. In this way, a row of reference pixels, as “A5–H5” in Fig. 3 (b), can be read

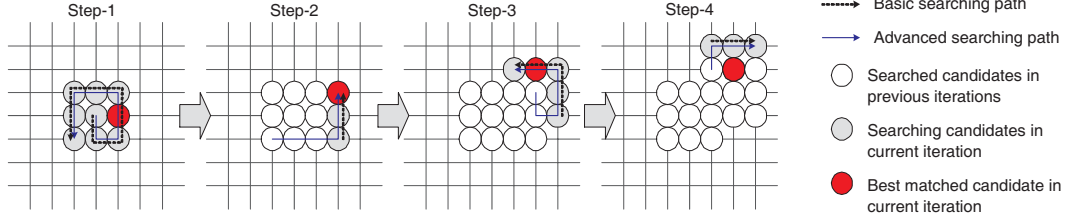


Fig. 4. Basic searching flow and advanced searching flow with 2-D random access for FSS.

in parallel. However, a column of reference pixels, as “C1–C8” in Fig. 3 (b), cannot be accessed in parallel. It is the so-called 1-D random access.

The ladder-shaped SW data arrangement is proposed to support 2-D random access. As shown in Fig. 3 (c), the second, third, fourth, and the following rows are rotated rightward by one, two, three, and etc pixels. In this way, the reference pixels of “A5–H5” and “C1–C8” are both arranged in different memories. Both the horizontally and vertically adjacent reference pixels can be accessed in parallel, which is the 2-D random access. For FS, because the search pattern is regular, the 1-D random access can efficiently support inter-candidate DR with systolic array. For fast algorithms, the search pattern can move with various direction. With 2-D random access of SW SRAMs, both the horizontally and vertically adjacent reference pixels can be read in parallel, and The proposed 2-D random access of SW SRAMs can solve the DR problems shown in Fig. 1 (b) and Fig. 2 (b).

To support inter-candidate DR with 2-D random access, the “16x16 Ref-pel Systolic Array” in Fig. 1 (a) is designed with four configurations: up-shift, down-shift, left-shift, and right-shift by one pixel. Figure 4 shows an example of FSS searching flow. The dotted line represents the basic flow. In the step-2, the systolic array is configured as up-shift configuration. The corresponding rows of reference pixels are read, and totally 18 (16+1+1) cycles are required. In the step 3, the systolic array is firstly set as up-shift configuration just like the step 2. After 18 cycle, the systolic array is changed to left-shift configuration. The corresponding two columns of reference pixels are read in the next two cycles, and two horizontally adjacent candidates can be immediately processed. Totally 20 (16+1+1+1+1) cycles are required for step-3. In the step 4, the inter-candidate DR, that cannot be supported by 1-D random access, can be achieved with right-shift configuration. 18 (16+1+1) cycles are required, just like the step-2.

Although the inter-candidate DR can be achieve in both horizontal and vertical directions, the DR rate and hardware utilization are still constrained by the long latency cycles of each step. Therefore, the advanced searching flow represented by the solid line in Fig. 4 is proposed. Because of the inter-candidate DR can be supported for any pairs of adjacent candidates, we just try to string up all required search candidates. Different from previous fast algorithms that will skip all searched candidates as many as possible, we utilize this redundant computation to tightly connect the searching flow of each step. Though the bubble cycles will occur, the long latency cycles can be eliminated. After step-1 in Fig. 4, the reusable data are stored in “16x16

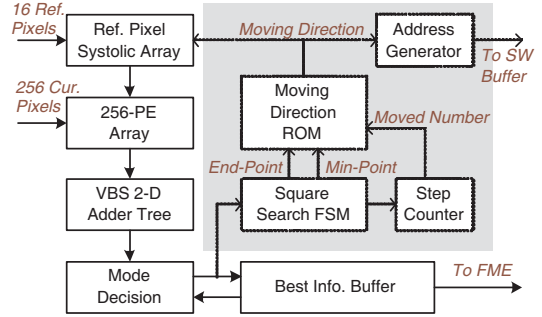


Fig. 5. Block diagram of the proposed low power IME architecture. The 2-D random access and the advanced searching flow is co-operated with ROM-based control core.

Ref-pel Systolic Array”. We use two bubble cycles to load two additional columns of reference pixels, and the step-2 can be immediately processed in the third cycle. In this example, totally 38 (24+5+6+3) cycles are required for the advanced flow, while 80 (24+18+20+18) cycles are required for the basic flow.

B. Architecture Design with ROM-Based Control Core

Figure 5 shows the block diagram of the proposed architecture. The data path is very similar to Fig. 1 except that the systolic array has four configurations. In order to support the 2-D random access and the advanced searching flow, a ROM-based FSS control core is designed. The “Moving Direction ROM” outputted the moving direction according to three parameters: the end-point (EP) and minimum-point (MP) of the previous step, and the moved number (MN) of the current step. Take the step-2 in Fig. 4 as example, the EP of previous step is bottom-left point, and the MP is right (red) point. Therefore, with the increasing of the MN, the ROM will sequentially output signals as right, right, right, up, and up. Then, the address generator and the systolic array operate in coordination according to the moving directions. The ROM size is 4×8×6, which are the maximum numbers of EP, MP, and MN, respectively.

C. Comparison

The redundancy access (RA) factor can be used to evaluate the performance of DR and is defined as follows:

$$RA = \frac{\text{SW SRAM bandwidth for reading ref-pels}}{\text{minimum requirement}}$$

The minimum requirement, or minimum required reference pixels, are the pixels number of the union of all searched candidates. For one candidate, the minimum requirement is 256 pixels. For two horizontally or vertically adjacent candidates, the

TABLE I
PERFORMANCE COMPARISON OF THE PROPOSED TECHNIQUES

Architecture	Parallel 1-D tree (Fig. 2)		2-D tree (Fig. 1)		
	1D	2D	1D	2D	2D
Random Access					
Advanced Flow	n/a	n/a	n/a	No	Yes
RA _{cand.-level}	8.77	5.09	6.86	3.23	1.54

TABLE II
COMPARISON OF POWER CONSUMPTION BETWEEN OUR ARCHITECTURE
AND PREVIOUS ARTS

	[6] Chao's ISCAS'02	[7] J.M.'s CICC'03	[8] Lin's SIPS'04	This Work
Process (μm)	0.35	0.13	0.18	0.18
Voltage (V)	3.3	1.0	1.8	1.3
Clock (MHz)	50	6.25	48.67	13.5
Power (mW)	223.6	3.28	8.46	1.40
Normalized Power (1.8V)	43.29	10.81	8.46	2.81
Search Pattern	Diamond Search	Gradient Decent	FSS/ 3SS	Parallel-VBS FSS w/ 1-ref.

minimum requirement is 272 (256+16) pixels. If the RA is two, that means the number of read pixels is twice of the minimum requirement. Please note that the moving path and search pattern shown in Fig. 4 is used as the model for the following comparison. The minimum required reference pixels in this case are 394 pixels for 20 search candidates. The comparison is shown in Tab. I. In general, the “2-D Tree” architecture has better DR efficiency than “Parallel 1-D Tree” architecture. The 2-D random access can support inter-candidate DR for both horizontal and vertical direction, while the advanced searching flow can reduce the latency cycles. After 2-D random access and advanced searching flow are applied, 77.6% ($1 - 1.54/6.86$) bandwidth and power of SW SRAMs can be saved for the “2-D Tree” architecture.

IV. IMPLEMENTATION AND SIMULATION RESULT

The proposed IME architecture is implemented in TSMC 0.18 μ 1P6M technology. The total logic gate count is 63.54K with maximum operation frequency of 27MHz. This design can support real-time encoding CIF 30fps videos with three modes, and the SRs are ± 32 pixel horizontally and ± 16 pixel vertically. In high quality mode, the coding parameter is the proposed content-adaptive parallel-VBS FSS algorithm with two reference frames. scheme [12]. In low power mode, the coding parameter is the proposed FSS with one reference frame. In ultra low power mode, the general FSS algorithm is used. That means only the MV predictor (MVP) is used as the initial search center.

Figure 6 shows power consumption results. Because the average computation complexity is generally smaller than the worst case, and the operating frequency is decided according to the worst case. The gated clock technique is implemented to turn the inoperative circuits off when ME sleeps. Besides, in the low power and ultra low power modes, the computation complexity is reduced and so is the operation frequency. When the operation frequency is 13.5 MHz, the voltage scaling down technique can be used to further reduce the power consumption. In the ultra low power mode, the power consumption is 1.424 mW for realtime encoding CIF 30fps videos at 13.5 MHz operation fre-

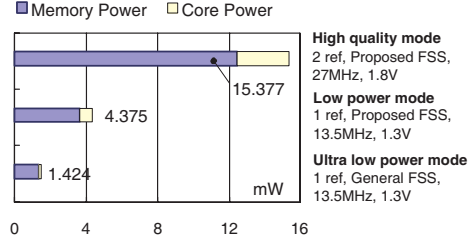


Fig. 6. Power consumption results of the proposed architecture.

quency.

The comparison with previous arts are listed in Table II. Because they are all designed for previous standards where VBS and MRF are not supported, the parameter of our design is set as general FSS with one reference frame. Since the different process scales and supply voltage are used, the normalized power consumption is calculated for the comparison. The proposed architecture can reuse data the most efficiently and has the lowest power consumption.

V. CONCLUSION

In this paper, we contributed a low-power architecture for IME of H.264/AVC. The “2-D SAD Tree” is adopted to support intra- and inter-candidate DR for the content-adaptive parallel-VBS FSS. The ladder-shaped SW data arrangement supporting 2-D random access is proposed to efficiently reuse data, and 52.92% SW SRAM power can be reduced. The advanced searching flow is applied to further reduce 52.5% power. According to the implementation result, in ultra low power mode, only 1.424 mW is required for realtime encoding CIF 30fps videos with 13.5 MHz operation frequency.

REFERENCES

- [1] Joint Video Team, *Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification*, ITU-T Recommendation H.264 and ISO/IEC 14496-10 AVC, May 2003.
- [2] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, “Overview of the H.264/AVC video coding standard,” *IEEE Transactions on CSVT*, 2003.
- [3] J.-H. Lee and N.-S. Lee, “Variable block size motion estimation algorithm and its hardware architecture for H.264,” in *Proceedings of ISCAS'04*, 2004.
- [4] Swee Yeow Yap and J.V. McCanny, “A VLSI architecture for variable block size video motion estimation,” *IEEE Transactions on CASII*, 2004.
- [5] 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,” *Accepted by IEEE Transactions on CASI*.
- [6] W.-M. Chao, C.-W. Hsu, Y.-C. Chang, , and L.-G. Chen, “A novel hybrid motion estimator supporting diamond search and fast full search,” in *Proceedings of ISCAS'02*, 2002.
- [7] M. Miyama, J. Miyakoshi, Y. Kuroda, K. Imamura, H. Hashimoto, and M. Yoshimoto, “A sub-mW MPEG-4 motion estimation processor core for mobile video application,” *IEEE Journal of Solid-State Circuits*, 2004.
- [8] S.-S. Lin, P.-C. Tseng, C.-P. Lin, and L.-G. Chen, “Multi-mode content-aware motion estimation algorithm for power-aware video coding systems,” in *Proceedings of IEEE Workshop on SIPS'04*, 2004.
- [9] Y.-H. Chen, T.-C. Chen, and L.-G. Chen, “Hardware oriented content-adaptive fast algorithm for variable block-size integer motion estimation in H.264,” in *Proceedings of ISPACS'05*, 2005.
- [10] L.-M. Po and W.-C. Ma, “A novel four-step search algorithm for fast block motion estimation,” *IEEE Transactions on CSVT*, 1996.
- [11] Telenor R&D, *ITU-T Recommendation H.263 Software Implementation*, Digital Video Coding Group, 1995.
- [12] J.-C. Tuan, T.-S. Chang, and C.-W. Jen, “On the data reuse and memory bandwidth analysis for full-search block-matching VLSI architecture,” *IEEE Transactions on CSVT*, 2002.