

Optimized 2-D SAD Tree Architecture of Integer Motion Estimation for H.264/AVC

Yibo FAN^{†a)}, Member, Xiaoyang ZENG[†], Nonmember, and Satoshi GOTO^{††}, Fellow

SUMMARY Integer Motion Estimation (IME) costs much computation in H.264/AVC video encoder. 2-D SAD tree IME architecture provides very high performance for encoder, and it has been used by many video codec designs. This paper proposes an optimized hardware design of 2-D SAD tree IME. Firstly, a new hardware architecture is proposed to reduce on-chip memory size. Secondly, a new search pattern is proposed to fully use memory bandwidth and reduce external memory access. Thirdly, the data-path is redesigned, and the performance is greatly improved. In order to compare with other IME designs, an IME design support D1 size, 30 fps with search range $[\pm 32, \pm 32]$ is implemented. The hardware cost of this design includes 118 KGates and 8 Kb SRAM, the maximum clock frequency is 200 MHz. Compared to the original 2-D SAD tree IME, our design saves 87.5% on-chip memory, and achieves 3 times performance than original one. Our design provides a new way to design a low cost and high performance IME for H.264/AVC encoder.

key words: IME, VBSME, 2-D SAD tree, H.264

1. Introduction

H.264 is the latest international video coding standard. It is also known as MPEG-4 part 10, or AVC (for Advanced Video Coding). It is jointly written by the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG) [1]. H.264 contains a number of new features that make it to compress video much more effectively than previous standards and to provide more flexibility for application to a wide variety of network environments.

Block-based Motion Estimation is very popular used in most of the video coding standards. As shown in Fig. 1, for a current MB (Macro Block) in current frame, it needs to search an area in the reference frame to find a 'best matching reference MB', which produces minimum residual data after subtracting the reference MB from the current MB. The process of finding the best matching point is known as motion estimation.

Integer Motion Estimation (IME) is the motion estimation for integer pixels in reference frame. Correspondingly, Fractional Motion Estimation (FME) is for sub-pixels, which is interpolated from integer pixels, in reference frame. IME is performed before FME, and IME occupies most of the computational cost.

Manuscript received August 5, 2010.

Manuscript revised November 7, 2010.

[†]The authors are with the State Key Lab. of ASIC & System, Fudan University, Shanghai, 200240, China.

^{††}The author is with IPS, Waseda University, Kitakyushu-shi, 808-0135 Japan.

a) E-mail: fanyibo@fudan.edu.cn

DOI: 10.1587/transele.E94.C.411

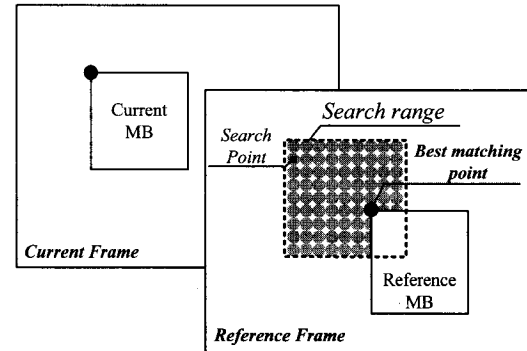


Fig. 1 Interger motion estimation.

The procedure of IME includes two steps: Firstly, the current MB and reference MB are processed by using sum of absolute differences (SAD) algorithm:

$$SAD(m, n) = \sum_{j=1}^N \sum_{i=1}^M |cur(i, j) - ref(i + m, j + n)|, \quad (1)$$

$$- P \leq m, n < P$$

$cur(i, j)$ is the pixels in current MB and $ref(i, j)$ is the pixels in reference MB. P is search range, (m, n) is one of search points in the search range, (M, N) is sub-block size. After complete this search point, it continues to search next one.

Secondly, the SAD value of current search point is compared with the minimum SAD value. If the current SAD value is smaller than the minimum value, the minimum SAD value is replaced by current one, and also the motion vector is calculated according to current search point. After all of search points are processed, the search point with minimum SAD in the search range is obtained. The motion vector is decided by the distance between this search point and the current MB.

For H.264/AVC, the difference from other video coding standards in IME is the Variable Block Size Motion Estimation (VBSME). As shown in Fig. 2, there are 7 sub-macro-block sizes in H.264/AVC, including 16×16 , 16×8 , 8×16 , 8×8 , 8×4 , 4×8 , and 4×4 . And totally, it has 41 partition modes for one MB, which enables very precise segmentation of moving regions [2], [4]. Hence, the coding performance is greatly improved [3], [4].

VBSME achieves higher compression ratio, however, the computational cost and required bandwidth is very huge. According to the instruction profile, an H.264/AVC encoder requires 315 Giga-instructions per second (GIPS) compu-

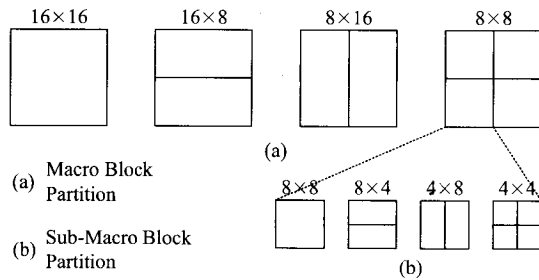


Fig. 2 Variable block size in H.264.

tation and 471 Giga-bytes per second (GBPS) memory access to code a CIF 30-fps video. And IME occupies about 74.29% (234 GIPS) computation and 77.49% (365 GBPS) memory access requirement of the whole encoder [7]. Compared with the previous standards, the IME of H.264/AVC is almost ten times more complex than that in the MPEG-4.

In order to achieve real time video encoding, especial for high resolution video, the hardware design of IME for H.264/AVC video encoder becomes very important. There are many hardware designs were proposed, such as [6]–[8], [11]. Among these designs, 2-D SAD tree IME architecture [6] is the most high performance and popular used in H.264/AVC encoder. However, the memory size of original 2-D SAD tree IME is large, and also the bandwidth requirement is very huge, which limit its performance and efficiency.

In this paper, an optimized 2-D SAD tree IME architecture is proposed to reduce on-chip memory size and bandwidth requirement. The performance is also improved by optimizing adder tree data path. The rest of the paper is organized as follows: The 2-D SAD tree IME architecture and the existing problems for this architecture are introduced in Sect. 2. The proposed optimized 2-D SAD tree IME architecture and proposed search pattern are presented in Sect. 3. The experimental results and comparison are shown in Sect. 4. Finally, conclusion is given in Sect. 5.

2. Original 2-D SAD Tree IME Architecture

2.1 2-D SAD Tree IME Architecture

2-D SAD tree IME is a widely used high performance architecture which is proposed by Chen in [6]. Figure 3 shows the top level modules in IME design. Current MB buffer stores 16×16 pixels of current MB (Cur.Pixel). Reference MB systolic array stores 16×16 pixels of reference MB (Ref.pixel), which represents one search point. Reference MB systolic array (RMSA) loads one row or one column of 16 reference pixels each clock cycle. At the same time, one row or one column of 16 pixels in RMSA are shifted out. 2-D SAD tree is used to calculate SADs value for each search point. All of the 41 sub-block SADs for one search point are calculated in one clock cycle. Search Window (SW) SRAM stores all of the reference pixels in search window of reference frame.

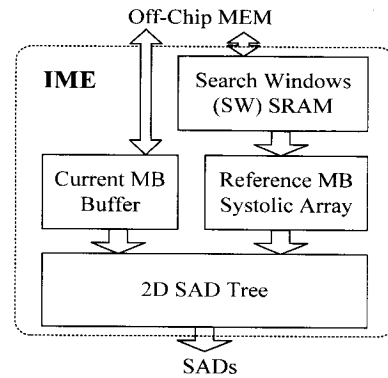


Fig. 3 Original IME architecture.

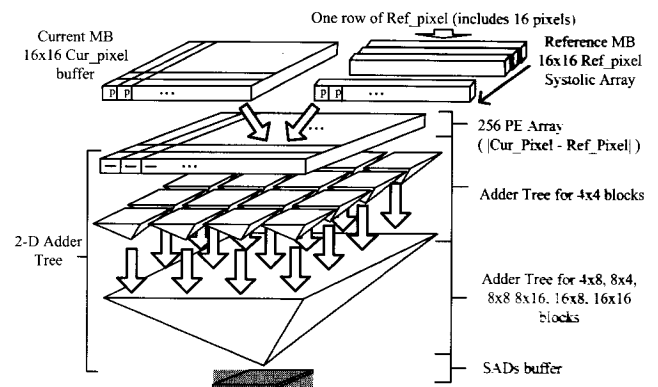


Fig. 4 Original 2-D SAD tree architecture.

Figure 4 shows the 2-D SAD tree architecture. The input data are 16×16 Cur.Pixel and 16×16 Ref.pixel. Residues are generated in “256-PE Array” and then summed up by 2-D adder tree. The small sub-blocks’ SAD can be reused for big one. For example, 4×4 SAD can be reused for 4×8 SAD (Two 4×4 makes up one 4×8). The output is all of 41 sub-blocks’ SADs for current search point.

Figure 5(a) shows the procedure of original 2-D SAD tree IME design. Firstly, all of the reference pixels in search window are loaded from Off-chip memory to SW SRAM, and the current MB is also loaded from Off-chip memory. Secondly, 16 clock cycles are needed to load 16 rows of Ref.pixel from SW SRAM to RMSA. Thirdly, the integer motion estimation is performed with a certain search pattern.

Figure 5(b) shows the search pattern used in 2-D SAD tree IME design, which is proposed in [6]. Firstly, the search direction is downward (①). One row of Ref.pixel is read from SW SRAM to RMSA, as shown in Fig. 5(c). Secondly, when the search point reaches the bottom of search range, the search direction changes to rightward (②). At this step, one column of Ref.pixel is read from SW SRAM, as shown in Fig. 5(d). Thirdly, the search direction changes to upward (③), and when it reaches top margin, the direction also changes to rightward (④). This procedure is continued until all of the search points are scanned.

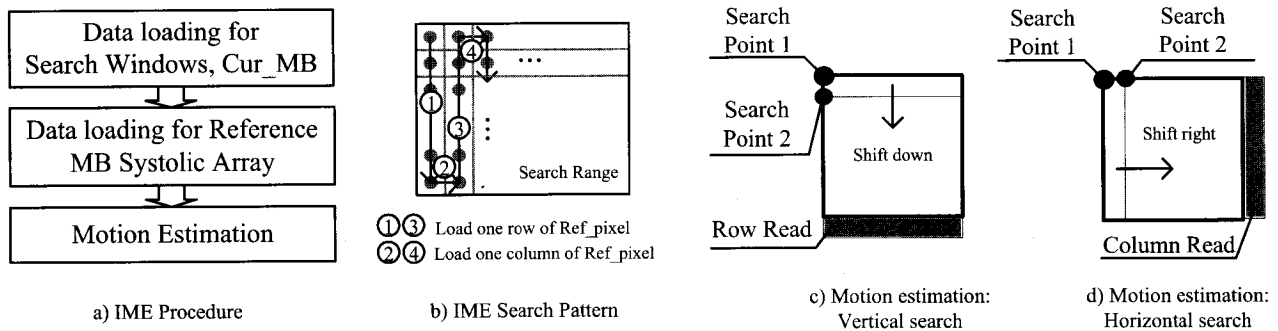


Fig. 5 Procedure of original 2-D SAD tree IME.

2.2 Problems Analysis

2-D SAD tree IME architecture achieves very high performance for VBSME. However, for real hardware implementation, there are some problems in this architecture which limits its performance and efficiency.

Memory Size & Access Problem

As discussed in last sub-section, in order to do motion estimation, all of the pixels in the search range should be loaded in SW SRAM in advance. While the search range increasing, the memory size will be increased squarely. Therefore, it needs huge on-chip memory.

For memory access, there are two kinds of memory access in 2-D SAD tree IME design: row access (Fig. 5(c)) and column access (Fig. 5(d)). However, SW SRAM can only supports row access. Column access should be realized by multiple row accesses. For H.264/AVC, the MB size is 16x16. In this way, one column access needs 16 rows access to finish. Thus, column access needs 16 clock cycles while row access only needs one cycle. Moreover, since only one pixel of each row is needed for column access, other pixels in the same row are useless in this step. In other words, most part of memory bandwidth is wasted.

Bandwidth Utilization Problem

There are two kinds of memory bandwidth in this IME design: Off-chip bandwidth and On-chip bandwidth. Off-chip means the bandwidth between IME and Off-chip memory. On-chip means the bandwidth between SW SRAM and RMSA. Off-chip bandwidth depends on the data reuse strategy and on-chip memory size. It has been discussed in many papers, and Tuan summarized these strategies in four levels in [9].

For on-chip bandwidth, it highly depends on memory partition of SW SRAM and search pattern. As discussed in Sect. 2.1, row pixels access (Row Read) is the main access type for IME search pattern. The neighboring search points in vertical direction share same IO ports while accessing memory (Suppose that the pixels are stored in SRAM row by row). However, for horizontal neighboring search points, the IO ports become different, which induces memory bandwidth waste.

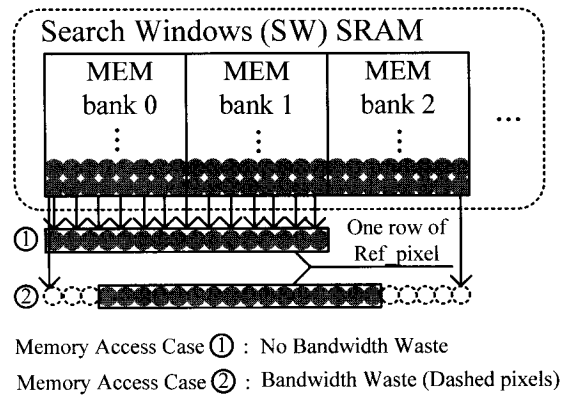


Fig. 6 Memory bandwidth analysis.

Figure 6 shows an example of bandwidth waste for IME. There are several physical memories used for SW SRAM (MEM bank 0, 1, 2 ...). Each MEM bank stores 8 pixels row by row. If search window's width is 47, totally 6 MEM banks are needed to store whole SW. In Fig. 6, it shows 2 cases of memory access: In case 1, it loads one row of Ref_pixel, which is exactly aligned to memory boundary (Only MEM bank 0 and 1 are used). In this case, the bandwidth is fully used. However, in case 2, it is not aligned to memory boundary (MEM bank 0, 1 and 2 are used), part of pixels (dashed pixels) are not used for IME even they have been read out from MEM banks. In this case, the bandwidth is wasted. While decreasing the number of MEM banks (Each MEM bank has larger bit-width), the wasted bandwidth is increased more. Increasing the MEM bank number is also not a good idea, it will greatly increase the silicon area and also the power consumption.

Some researchers proposed segmentation-free access SRAM [10] to fully use bandwidth. However, it still needs to load whole search window to on-chip memory. Furthermore, it requires custom design for SRAM, and the maximum frequency of SRAM is greatly reduced due to the complex decoders for horizontal/vertical access.

3. Proposed 2-D SAD Tree IME Design

3.1 Proposed 2-D SAD Tree IME Architecture

The proposed IME architecture is shown in Fig. 7(a). Different from Fig. 3, the proposed IME design uses a Reference Pixel Stack (RPS) SRAM instead of SW SRAM, and this stack only exchange data with RMSA. It is used to store the pixels overflowed from RMSA, which can be reused for next search point. It is useful for data-reuse in level A and level B. In order to achieve data-reuse in level C and level D, a pixel cache is used for pixel access from Off-chip memory.

Figure 7(b) shows the proposed 2-D SAD tree architecture with optimized data path. Two levels of registers (Gray part) are used to shorten data path. Carry-save adders are used in the first level adder tree (for 4x4 adder tree). As a result, the original long critical path is cut to 3 short paths. The maximum frequency of 2-D SAD tree is greatly improved.

Figure 7(c) shows the memory organization for Reference MB construction module. RMSA is a 16x16 reference pixel register array which supports shift-right, shift-left and shift-down. RPS SRAM is used to store overflowed column data (15 pixels, which can be reused later) while search path moving left or right. The left stack and the right stack are actually one same physical dual-port SRAM. The IO register buffer is used to store external data temporary.

3.2 Proposed IME Search Pattern

The proposed search pattern for IME is shown in Fig. 8. Figure 8(a) is the overview of this pattern. The detailed searching process is given as following:

Initial Step (Fig. 8(b))

In this step, the first row of search points is loaded from Off-IME memory, and RMSA is used as a transpose buffer to store the pixel data to RPS SRAM. As shown in the figure, firstly, the external data is loaded row by row and shifted into the right most strip of RMSA (the area with bias in Fig. 8(b)). Secondly, while the strip is fully filled, then RMSA does left-shifting, and pushes overflowed columns of pixel into the RPS SRAM. Multiple of columns can be loaded together, which depends on the bit-width of memory interface.

Motion Estimation Step (Figs. 8(c), (d), (e)):

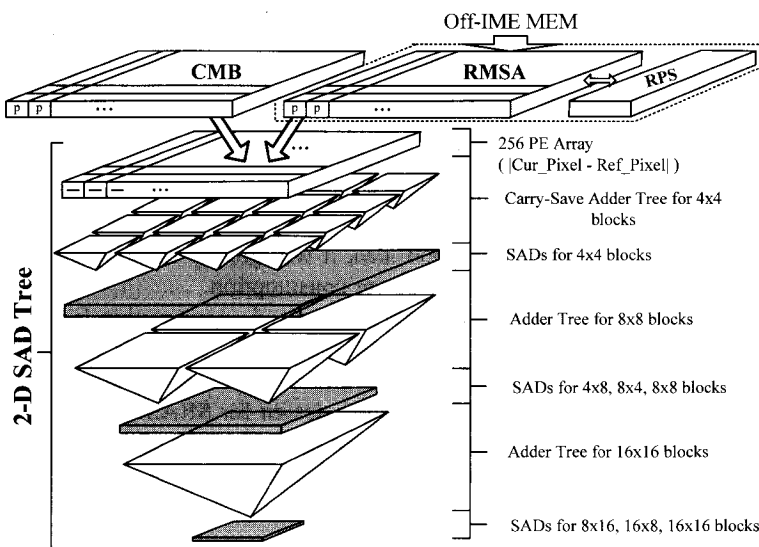
As shown in Fig. 8(a), there are 5 steps in motion estimation:

Step (1) (Fig. 8(c)): In this step, the search points are scanned leftward. Columns of reference pixels are needed to update RMSA. As RPS SRAM has already held 15 pixels which can be reused for new search points, only one pixel should be loaded from outside. For memory bandwidth utilization's consideration, we load multiple pixels, and store them into IO register buffer firstly. Once a new pixel is required, it can be easily loaded from this register buffer. The pushed-out pixels will be re-pushed in to the RPS SRAM for future data-reuse.

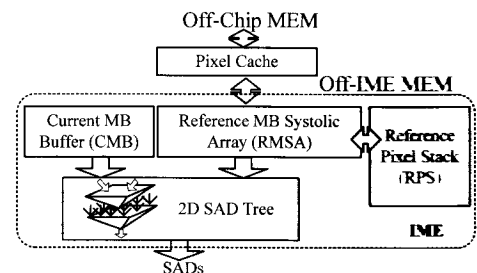
Step (2) (Fig. 8(d)): The search direction is downward in this step. One row of Ref. pixel (16 pixels) need to be loaded from Off-IME memory to update RMSA. There is no data exchange with RPS SRAM.

Step (3) (Fig. 8(e)): Search points are scanned rightward and columns of pixels are needed to update RMSA in this step. The memory access is similar as step (1).

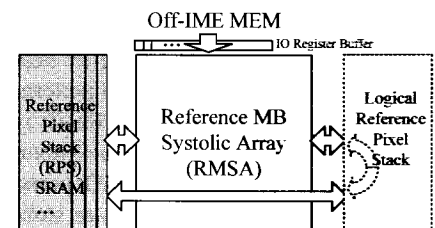
Step (4) and step (5) are similar as step (2) and step (1).



b) Proposed 2-D SAD Tree architecture



a) Proposed IME architecture



c) Reference MB construction Module (RMSA+RPS)

Fig. 7 Proposed optimized 2-D SAD tree IME architecture.

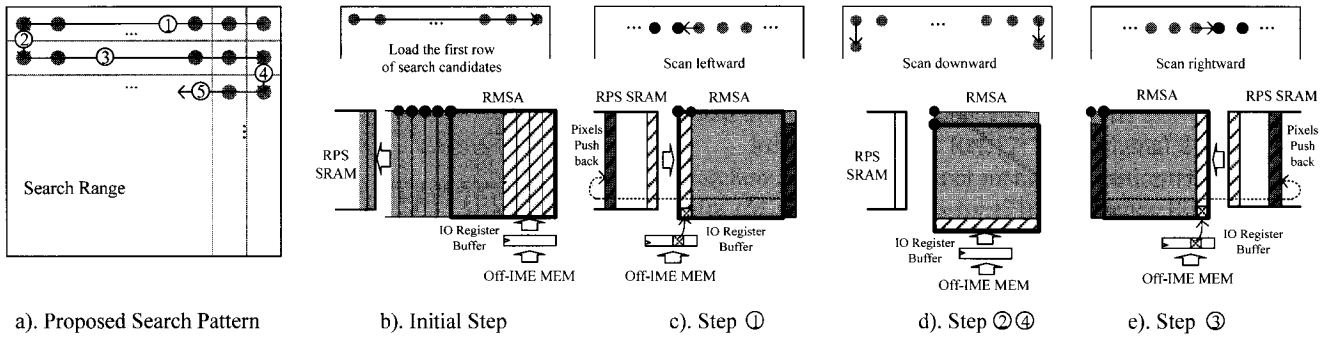


Fig. 8 Proposed search pattern for IME.

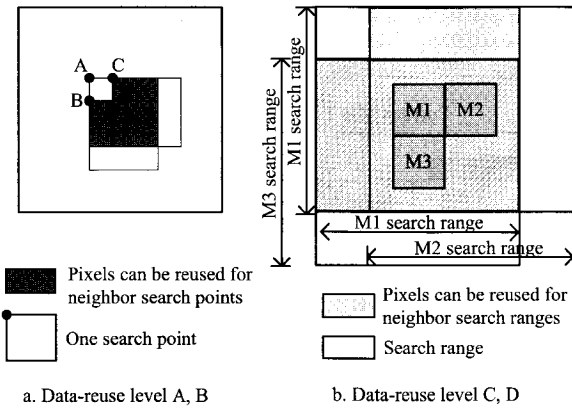


Fig. 9 Data-reuse strategies.

3.3 Memory Access in Our Design

There are three types of memory access in our design: Off-chip memory access, Off-IME memory access and On-IME memory access. Off-IME and On-IME can be considered as on-chip memory access.

Off-chip memory access is the memory access between pixel cache and Off-chip memory, such as DDR memory. The pixels are loaded into cache block by block with a specific block size. Off-IME memory access is memory access between pixel cache and IME module. As shown in Fig. 8, all of the memory access can be realized by row access. There is no bandwidth waste in our design. On-IME memory access is the memory access between RMSA and RPS SRAM. For On-IME memory access, only push and pop operations are used, and all of the pixels are read/write column by column. The bandwidth of RPS SRAM is also fully used.

3.4 Data-Reuse Strategy in Our Design

Data-reuse is very important to reduce Off-chip memory bandwidth. Tuan proposed four levels strategies in [9]:

Level A and level B achieve data-reuse among neighbor search points. As shown in Fig. 9(a), while doing motion estimation in search range, there are many overlapped pixels

for each search point. If these pixels are stored in on-chip memory temporarily, it is no need to access Off-chip memory for next search point. In this way, the Off-chip bandwidth can be saved. Similarly, level C and level D achieve data-reuse in search range level. As shown in Fig. 9(b), the overlapped search range can be reused for M1, M2 and M3.

For our IME design, combining with our proposed IME search pattern, RPS SRAM perfectly supports level A and level B data-reuse. In order to further achieve level C and D data-reuse, a pixel cache is used, as shown in Fig. 7(a). This pixel cache is very flexible for further data-reuse optimization, the levels of data reuse only depends on the size of cache.

4. Comparison & Experimental Results

4.1 On-Chip Memory Comparison

On-chip memory size sensitive IME

Low cost IME with small on-chip memory. Only level A and B data-reuse strategy can be used. For original 2-D SAD tree IME, the on-chip memory is SW SRAM which is used to store whole search window's pixels. The size of SW SRAM can be represented as:

$$M_{SWSRAM} = (m + 15) \times (n + 15) \times 8 \text{ (bit)} \quad (2)$$

m is the width of search range, n is the height of search range. For our proposed IME, it is not need to preload whole search window into on-chip memory. Only one row of search points is required to be stored in the RPS SRAM. The RPS SRAM size is:

$$M_{RPSRAM} = (m - 1) \times 15 \times 8 \text{ (bit)} \quad (3)$$

Similarly, m is the width of search range. Figure 10 shows the memory size comparison of original IME design (SW SRAM) and our proposed IME design (RPS SRAM). The search range is set as a square range (n equals m). From this figure, it can be seen that our proposed IME design saves many on-chip memory, especially for large search range.

Off-chip bandwidth sensitive IME

In order to save Off-chip bandwidth, level C or level D data reuse strategies should be used. The original IME design

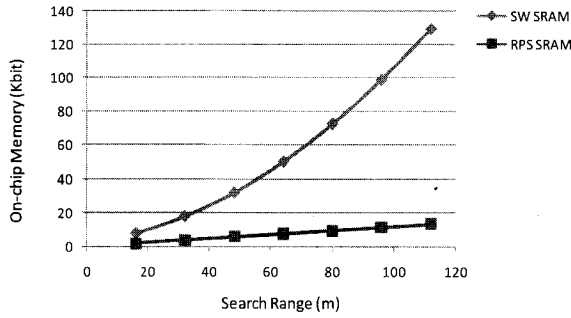


Fig. 10 On-chip memory size comparison.

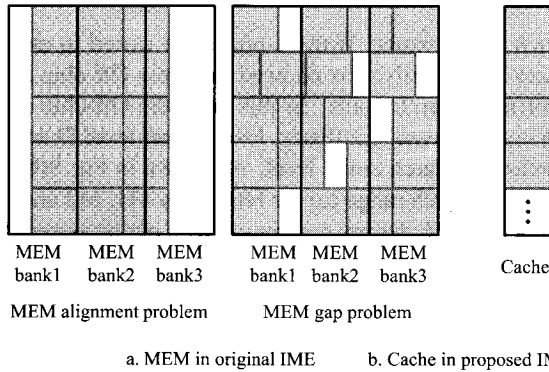


Fig. 11 On-chip memory organization comparison.

uses SW SRAM, which includes several MEM banks, to support data-reuse level A, B and also level C, D. There are two problems in this IME design: MEM alignment problem and MEM gap problem. As shown in Fig. 11(a), while search range is not exactly aligned to MEM bank boundary, part of MEM is wasted. While adopting data-reuse level D strategy, it will cause many blank gaps in MEM, as shown in Fig. 11(b).

In contrast, our design uses two on-chip memories for data-reuse: RPS SRAM for level A, B, and Pixel Cache for level C, D. The benefit is that there is no waste for on-chip memory, and the hardware implementation becomes much easier.

4.2 Memory Bandwidth Utilization Comparison

There are two kind of bandwidth: On-chip bandwidth and Off-chip bandwidth. For Off-chip bandwidth, it depends on data-reuse strategy. The bandwidth utilization is the same for original 2D SAD tree IME and our design under the same data-reuse level.

For on-chip bandwidth, as discussed in Sect. 2.2, the original 2D SAD tree IME waste much bandwidth. The data read out from MEM bank can't be totally used for motion estimation. In contrast, for our design, the on-chip bandwidth, which includes On-IME and Off-IME, is 100% used. There is no bandwidth waste. In order to compare the bandwidth utilization, a benchmark of D1 resolution (720 × 480) video with frame rate of 30 fps was used in our experiment. The

Table 1 On-chip Bandwidth utilization comparison.

IME Designs	Number of MEM Banks	Bit-width for each MEM	Bandwidth Utilization (On-IME)
Original 2-D SAD tree IME	1*	632-bit	20.3%
	3	128-bit	55.9%
	6	32-bit	70%
	16**	8-bit	100%
Proposed 2-D SAD tree IME	1	120-bit	100%

*: Same architecture as Reference [6], implemented by this paper.
 **: Same architecture as Reference [7], implemented by this paper.

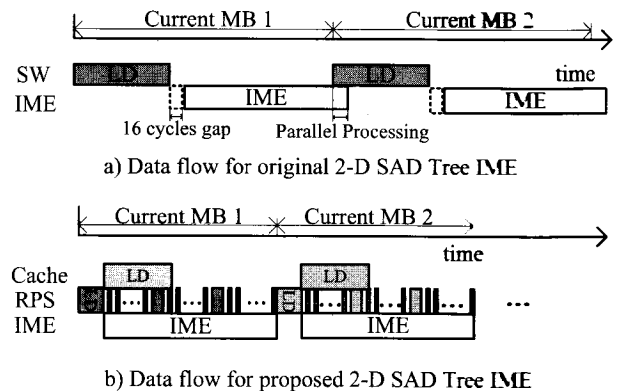


Fig. 12 Data flow comparison.

search range is set to [±32, ±32].

The comparison of on-chip memory bandwidth utilization is shown in Table 1. For original IME, the bandwidth utilization depends on memory partition and IO width. In this table, we listed 4 example implementations. For worst case, which only using 1 MEM bank (contains 79 pixels in a word, or 632 bit/word), the bandwidth utilization is only 20.3%. In order to achieve 100% utilization, it should at least use 16 MEM banks. In contrast, for our proposed design, we use Pixel cache and RPS SRAM to achieve 100% bandwidth utilization.

4.3 Dataflow Comparison

Figure 12 shows the data flow for original 2-D SAD tree IME and our proposed IME. In original one, all of the data should be loaded before doing IME. There are 16 cycles' gap between LD (Loading Data) and IME, which is used to fill pixels into RMSA in the initial step. The parallel processing of IME and LD depends on the memory organization for SW SRAM.

For our proposed one, most of the memory access is parallel processed with motion estimation. Pixel cache and RPS SRAM can be parallel accessed in IME procedure. The gap between IMEs becomes much smaller, which means our

Table 2 Throughput on each part of proposed IME.

Data-reuse	On-chip memory size	Off-chip throughput	Off-IME throughput	On-IME throughput
Level A+B	8Kb RPS	253MB/s	-	2.5GB/s
Level A+B+C	8Kb RPS + 40Kb Cache	51.2MB/s	253MB/s	2.5GB/s
Level A+B+C+D	8Kb RPS + 371Kb Cache	10.4MB/s	253MB/s	2.5GB/s

Table 3 Implementation and comparison.

Ref	Tech	Area (Gates)	Freq	Supported Resolution & Data-reuse	Year
[6]	0.18 μ m	88.6 K + 64 Kb SRAM	110 MHz	720×480 @ 10 fps, Level C [± 64 , ± 32]	2006
[7]	0.18 μ m	131.2 K + 64 Kb SRAM	66 MHz	352×288 @ 30 fps, Level C [± 32 , ± 16]	2007
[8]	0.18 μ m	179 K + 64 Kb SRAM	167 MHz	720×480 @ 30 fps, Level C [± 32 , ± 32]	2008
This paper	0.18 μ m	118 K + 8 Kb SRAM	200 MHz	720×480 @ 30 fps, Level B [± 32 , ± 32]	2010
		118 K + 48 Kb SRAM		720×480 @ 30 fps, Level C [± 32 , ± 32]	
		118 K + 379 Kb SRAM		720×480 @ 30 fps, Level D [± 32 , ± 32]	

design achieves much higher parallelism and saves many clock cycles compared to original IME design. Moreover, the memory access operation is averagely distributed into whole time pieces. For some high latency system, it is very useful.

4.4 Hardware Implementation and Comparison

In order to compare the hardware cost and on-chip memory size in the similar specification, we designed an IME which supports D1 size (720 × 480) video coding. The maximum frame rate of our design achieves 30 fps, and the maximum search range is [± 32 , ± 32].

Table 2 shows the throughput of each part in our IME design under different data-reuse strategies. The definition of each part's throughput can be found in Sect. 3.3. From this table, Pixel cache size and Off-chip throughput are highly related to data-reuse strategy. Since RPS SRAM is within IME design, the size of RPS is only depends on search range, and the throughput of Off-IME and On-IME are only depend on video resolution, reference frame number and search range size.

Using TSMC 0.18 μ m CMOS standard cell library and Synopsys Design Compiler Tools, the ASIC implementation

results and comparison are shown in Table 3. JM [5] software is used to generate test pattern and function verification data. The total hardware cost for our design is 118K Gates, and the maximum frequency is 200 MHz.

Reference [6] and Ref. [7] uses original 2-D SAD tree architecture with different memory partition (1 MEM bank for [6], 16 MEM banks for [7]). Reference [8] is the latest reconfigurable IME design uses reconfigurable architecture, which is different from 2-D SAD tree architecture. The supported resolution shown in Table 3 includes three parameters: Resolution (Width × Height), Frame rate (Frame-Per-Second, fps) and Search range ([\pm Horizontal, \pm Vertical]). Moreover, all of the referred IME designs only support up to level C data-reuse. Our design supports up to level D data-reuse.

Compared with others' design, our proposed lowest cost IME design requires much smaller on-chip memory (saves 87.5% under level B data-reuse strategy), and the performance is about 1.5 times of original one [6] (30 fps versus 10 fps, Half search range). Our design provides most efficiency for Integer Motion Estimation in H.264/AVC.

5. Conclusion

In this paper, we proposed an optimized 2-D SAD tree IME architecture. New hardware architecture and new search pattern are proposed to reduce on-chip memory size and improve memory bandwidth utilization. The data path is also optimized by using carry-save adder and inserting register buffer to achieve higher frequency. The experiment results show that, for D1 resolution video encoder with search range [± 32 , ± 32], our proposed design saves 87.5% on-chip memory. It shows great advantage for H.264/AVC video encoder design.

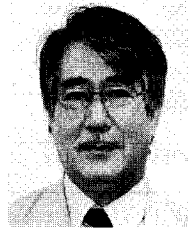
Acknowledgement

This work was supported by CREST, JST.

References

- [1] 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, first quarter, 2004.
- [2] T. Wiegand, G.J. Sullivan, G. Bjntegaard, 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] Iain E.G. Richardson, *H.264 and MPEG-4 Video Compression, Video coding for next-generation multimedia*, John Wiley & Sons, 2003.
- [4] ITU-T, "Advanced video coding for generic audiovisual services," <http://www.itu.int/rec/T-REC-H.264>, 2003. J. Clerk Maxwell, *A Treatise on Electricity and Magnetism*, 3rd ed., vol.2, pp.68–73, Oxford, Clarendon, 1892.
- [5] JM 11.0 source code, <http://iphome.hhi.de/suehring/tml/>
- [6] 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, March 2006.

- [7] T.C. Chen, Y.H. Chen, S.F. Tsai, S.Y. Chien, and L.G. Chen, "Fast algorithm and architecture design of low-power integer motion estimation for H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol.17, no.5, pp.568–577, 2007.
- [8] Y.B. Fan, T. Ikenaga, and S. Goto, "Reconfigurable variable block size motion estimation architecture for search range reduction algorithm," *IEICE Trans. Electron.*, vol.E91-C, no.4, pp.440–448, April 2008.
- [9] 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 Trans. Circuits Syst. Video Technol.*, vol.12, no.1, pp.61–72,
- [10] J. Miyakoshi, Y. Murachi, T. Ishihara, H. Kawaguchi, and M. Yoshimoto, "A power- and area-efficient SRAM core architecture with segmentation-free and horizontal/vertical accessibility for super-parallel video processing," *IEICE Trans. Electron.*, vol.E89-C, no.11, pp.1629–1636, Nov. 2006.
- [11] Y. Murachi, J. Miyakoshi, M. Hamamoto, T. Iinuma, T. Ishihara, F. Yin, J. Lee, H. Kawaguchi, and M. Yoshimoto, "A sub 100 mW H.264 MP@L4.1 integer-pel motion estimation processor core for MBAFF encoding with reconfigurable ring-connected systolic array and segmentation-free, rectangle-access search-window buffer," *IEICE Trans. Electron.*, vol.E91-C, no.4, pp.465–478, April 2008.

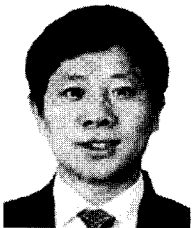


Satoshi Goto was born on January 3rd, 1945 in Hiroshima, Japan. He received the B.E. and M.E. degree in Electronics and Communication Engineering from Waseda University in 1968 and 1970, respectively. He also received the Dr. of Engineering from the same university in 1981. He is IEEE fellow, member of Academy Engineering Society of Japan and professor of Waseda University. His research interests include LSI system and Multimedia System.



coding and associated VLSI architecture.

Yibo Fan received the B.E. degree in electronics and engineering from Zhejiang University, China in 2003, M.S. degree in Microelectronics from Fudan University, China in 2006, and Ph.D. degree in engineering from Waseda University, Japan in 2009. From 2009 to 2010, he worked as an Assistant Professor in Shanghai Jiaotong University. And currently, he is the Assistant Professor in Department of Microelectronics of Fudan University. His research interesting includes information security, video



signal processing, and communication systems. Prof. Zeng is the Chair of Design-Contest of ASP-DAC 2004 and 2005, also the TPC member of several international conferences such as ASCON 2005 and A-SSCC 2006, etc.

Xiaoyang Zeng received the B.S. degree from Xiangtan University, China in 1992, and the Ph.D. degree from Changchun Institute of Optics and Fine Mechanics, Chinese Academy of Sciences in 2001. From 2001 to 2003, he worked as a post-doctor researcher at the State-Key Lab of ASIC & System, Fudan University, P.R. China. Then he joined the faculty of Department of Micro-electronics at Fudan University as an professor. His research interests include information security chip design, VLSI