

# A HARDWARE-EFFICIENT ARCHITECTURE FOR MULTI-RESOLUTION MOTION ESTIMATION USING FULLY RECONFIGURABLE PROCESSING ELEMENT ARRAY

Xianghu Ji<sup>1</sup>, Chuang Zhu<sup>1</sup>, Huizhu Jia<sup>1</sup>, Xiaodong Xie<sup>1</sup>, Haibin Yin<sup>1,2</sup>

<sup>1</sup>Nat'l Engineering Lab for Video Technology, Sch'l of EECS, Peking University

<sup>2</sup>Sch'l of EE, China JiLiang University  
{xhji, czhu, hzjia, xdxie, hbyin}@jdl.ac.cn

## ABSTRACT

Integer motion estimation (IME) for block-based video coding presents a significant challenge in external memory bandwidth, data latency, and circuit area with the increase of coding complexity and video resolution. To conquer these problems, this paper proposes a hardware-efficient VLSI architecture for multi-resolution motion estimation algorithm (MMEA) based on fully reconfigurable processing element (PE) array. On-chip storage and PE array are carefully designed to support parallel computation and hardware resource sharing. In addition, low data latency is obtained by arranging internal logics in parallel according to the data dependency. As a result, our design can support real time processing of 1080P@30fps with 2 reference frames and a search range of 256x192 and it is implemented under SMIC 0.18- $\mu$ m CMOS technology with 920K logic gates and 192 KB SRAMs. Compared with previous work, our design can achieve the best performance-price rate benefiting from the proposed re-configurable PE array.

**Index Terms**—architecture, re-configurable, multi-resolution motion estimation, video coding

## 1. INTRODUCTION

Today, to provide higher image quality and more vivid perception for users, new video coding technologies and higher video resolution are both required in telecom and broadcasting system which bring great challenges for the video coding. For the block-based video coding system, Integer motion estimation (IME) can remove most of the temporal redundancy. IME is the most complex module which contributes nearly 70% computation of the whole video encoder [1]. The new techniques of variable block sizes (VBS) and multiple reference frames (MRF) [2] in IME increase the computation complexity even more.

Compared with the previous standards, the IME of H.264/AVC is almost ten times more complex than that in MPEG-4[1]. Furthermore, with the increased video resolutions, real-time implementation of the IME will become more challenging. Hence, hardware acceleration is a must [1].

With the benefits of microelectronic technology, several IME VLSI architectures with high throughput are proposed [3][4]. Nevertheless, further IME algorithm and architecture optimization are desired to achieve optimal balance among rate distortion performance, throughput and hardware resources. A hardware-oriented multi-resolution motion estimation algorithm (MMEA) was brought out in the work of Yin et al [5], which can achieve approximately 0.1dB PSNR degradation in average compared with the full-search block matching (FSBM) algorithm.

In this paper, a hardware-efficient MMEA VLSI architecture is designed and implemented based on the fully reconfigurable PE array. The proposed architecture can make full use of PEs between each search level and achieve a optimal balance between the hardware resource and throughput.

The remainder of this paper is organized as follows. Section 2 gives a brief introduction to the hardware-oriented multi-resolution motion estimation algorithm (MMEA). In Section 3, an overall VLSI architecture is developed and a fully reconfigurable PE array is proposed based on the SAD decomposition. Section 4 shows the implementation results as well as the comparisons with other works. Finally, some concluding remarks are given in Section 5.

## 2. HARDWARE-ORIENTED MMEA

In this section, we will introduce the three-level multi-resolution motion estimation algorithm. The adopted MMEA searches for the best integer MVs using a hierarchical strategy. Fig.1 gives a brief description of the multi-resolution frame structure for the three-level MMEA.

This work was supported by the National Basic Research Program (973) of China (No. 2009CB320903) and NSFC 60802025.

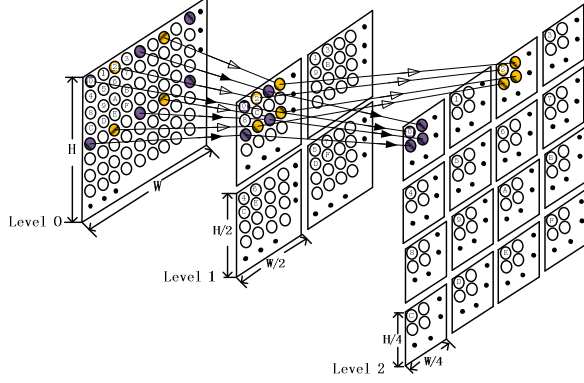


Fig.1. multi-resolution frame structure

$W$  and  $H$  are the width and the height of the frame respectively. MMEA consists of 3 resolution levels. Level 0 is the bottom level. The 4:1 direct down-sample is performed at each level to form its upper level. As a result, level 0 is sub-divided into four level 1 sub-windows. Similarly, level 1 is sub-divided into sixteen level 2 sub-windows. The three-level MMEA is performed from the coarsest 16:1 direct down-sampled level 2 to the finest unsampled level 0 in each sub-frame. And the original micro-block is down-sampled in the same way into three resolutions in each LEVEL.

The overall search process can be divided into three levels, as presented in Fig.2.

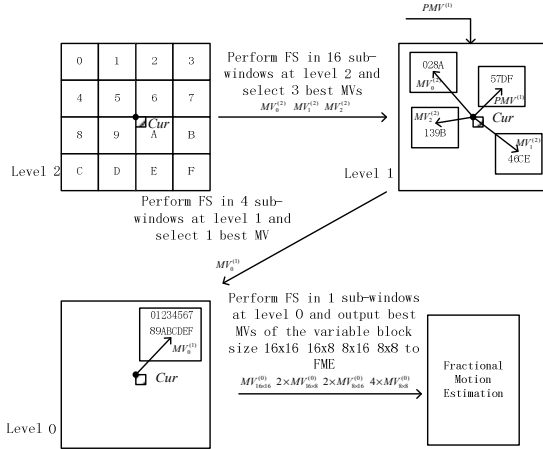


Fig.2. three-level MMEA search procedure

A detailed search procedure at each level will be described below.

a) *Search at Level 2:* Suppose the size of the whole search window is  $[-srx, srx] \times [-sry, sry]$ . Therefore, the size of the each sub-window at level 2 is  $[-srx/4, srx/4] \times [-sry/4, sry/4]$ . In order to accelerate the search speed and improve the search performance, the search window at level 2 is divided into 16 sub-areas mapping to the 16 sub-windows, as described in Fig.2. 16-way parallel full search is performed in each sub-area with RDO-based matching

criterion. At the end of the search procedure, sixteen candidate MVs with the minimum cost in each sub-window are found. From the 16 candidates, three MV candidates denoted as  $MV_0^{(2)}, MV_1^{(2)}, MV_2^{(2)}$  at the minimum cost are selected.

b) *Search at Level 1:* Prediction of motion vector (PMV) is taken into consideration at level 1. PMV, together with three MV candidates that are searched at level 2, will be adopted as the center point of each search window at level 1. Searches are performed around the four candidates with the purpose of finding an optimal MV candidate for the search at level 0. Search windows centered at  $MV_0^{(2)}, MV_1^{(2)}, MV_2^{(2)}$  and  $PMV$  are mapped to the sub-windows at level 1. Similarly, full search is performed in each search window with a search range of  $[-srx^{l1}, srx^{l1}] \times [-sry^{l1}, sry^{l1}]$  at level 1. Hence, four MV candidates are obtained at the end of the search procedure with only one candidate MV denoted as  $MV_0^{(1)}$  at the minimum cost selected from the four MV candidates.

c) *Search at Level 0:* In the proposed three-level MMEA, variable block size motion estimation is performed at level 0. The type of VBS includes  $16 \times 16, 16 \times 8, 8 \times 16, 8 \times 8$ . And their SADs all can be derived from four  $8 \times 8$  SADs [7]. The search window mapping to level 0 is centered at  $MV_0^{(1)}$  with search range of  $[-srx^{l0}, srx^{l0}] \times [-sry^{l0}, sry^{l0}]$  at level 0. Full search is adopted for MMEA at level 0. The VBS SADs can be calculated simultaneously. Hence, optimal VBS MVs denoted as  $MV_{16 \times 16}^{(0)}, 2 \times MV_{16 \times 8}^{(0)}, 2 \times MV_{8 \times 16}^{(0)}, 4 \times MV_{8 \times 8}^{(0)}$  can be obtained at the same time at the end of search at level 0.

Detailed analysis on the rate distortion performance of the three-level MMEA can be found in the previous work [5].

### 3. HARDWARE ARCHITECTURE DESIGN

#### 3.1. Overall VLSI Architecture

We proposed a VLSI architecture based on the three-level MMEA introduced in section 2. Fig.3 shows the block diagram of the proposed architecture. It consists of three main function modules with the corresponding control modules (level2\_ctrl, level1\_ctrl, level0\_ctrl, search window management), calculation module (PE array, MVs gen, cost compare) and storage module (search window ram array, 5x4 systolic array).

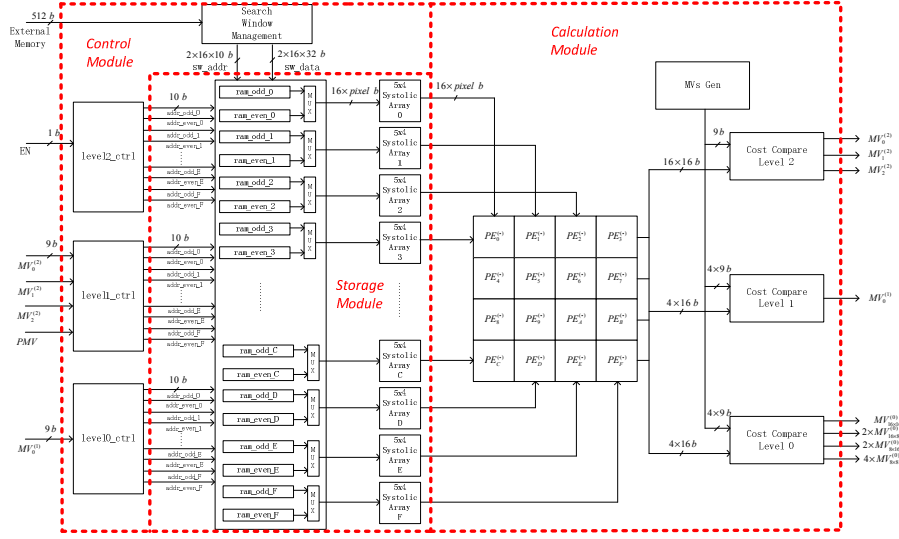


Fig.3. Block diagram of the proposed architecture based on three-level MMEA

### 3.2. On-chip Storage Structure

The candidate searching will move downward, upward or rightward in each level. Meanwhile, the start point of the searching will be varied in different conditions. And low latency and hardware cost are both required to be constrained in our design. So we need a flexible storage structure to support the high memory bandwidth and efficient data reuse. As is stated in Section 2, in the proposed three-level MMEA, search window is down-sampled into 16 sub-windows at level 2. In our design, 16 sub-windows are mapped into 16 memory units. And in order to achieve a high data bandwidth for calculation module, the sub-window is segmented into two types of region which are even-odd interleaved with four pixel columns in each group. So each memory unit is divided into two types of RAMs corresponding to the region's type in each sub-window, namely  $ram\_even_n$  and  $ram\_odd_n$ , as depicted in Fig.4. Here  $even\_col^m$  and  $odd\_col^m$  denote  $m$ th group in the corresponding region. The size of each RAM is

$$\left(\frac{2 \times srx + 128}{16}\right) \times \left(\frac{2 \times sry + 32}{4}\right) \text{ words.}$$

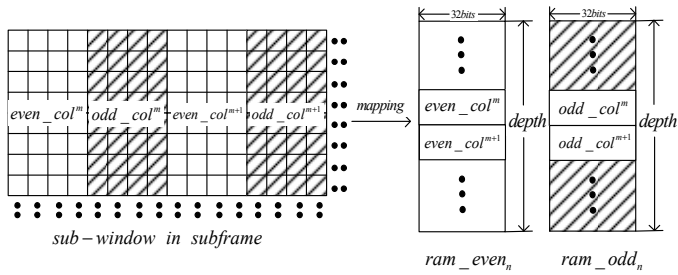


Fig.4. On-chip memory organization

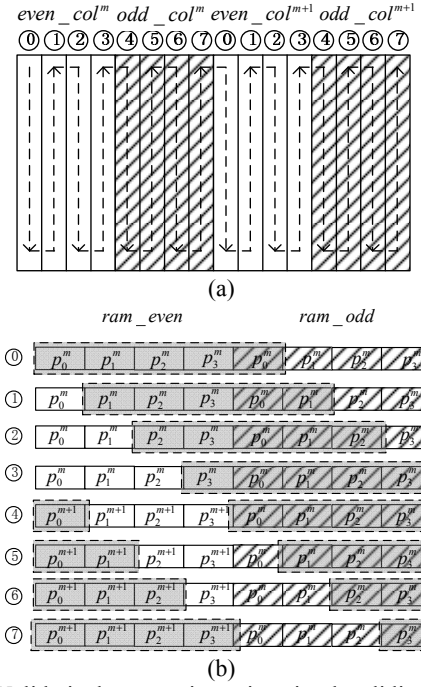


Fig.5. Valid pixels generation using circular sliding window method. (a) sub-window search process in sub-frame. (b) circular sliding window.

First, all pixels are grouped together according to their regions' type and stored in each corresponding RAM simultaneously in the mapping order (from left to right), as described in Fig.4. Then valid pixels which are needed in the SAD calculation can be obtained by the combination of the data from  $ram\_even_n$  and  $ram\_odd_n$ .

Here, the scan pattern presented in Fig.5 (a) is adopted in our proposed architecture. Circular sliding window method as shown in Fig.5 (b) is proposed to generate the valid pixels during the search process.  $p_k^m$  denotes the  $k$ th pixel column in the  $m$ th group.

### 3.3. Fully Re-configurable Parallel PE Array Structure

A large search range in a search process can guarantee the coding performance in the IME algorithm but meanwhile it will significantly increase the computational complexity. So, high throughput is recommended in the design of IME architecture. Simply multiplying PE arrays for the search in different levels will dramatically increase the hardware resource consumption. Therefore, flexible parallelism for the PE array is designed in order to maximize the resource reuse. Our structure can achieve nearly 100% PE reuse though different configuration at each level. According to the three-level MMEA algorithm introduced in Section 2, all computation of SADs at each level can be decomposed as the combination of four pixels SAD calculation at Level 2 which is described in Fig.6.

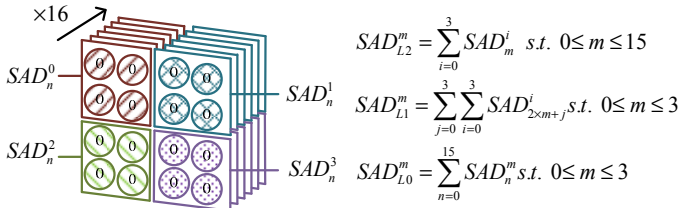


Fig.6. SAD decomposition

Here  $SAD_n^{(*)}$  handles the pixels mapped to the same sub-window.  $SAD_{L2}^{(*)}$ ,  $SAD_{L1}^{(*)}$ ,  $SAD_{L0}^{(*)}$  denote the SADs computed at each level.

At Level 2, sixteen candidate MVs are searched simultaneously corresponding to the sixteen sub-windows. At Level 0, four 8x8 block-sized motion searches are required to be performed in order to support VBS motion estimation.

Hence, in our design the basic processing element (PE) is the four-pixel SAD unit, as shown in Fig.7.(a). With this basic unit, each level can be easily implemented by re-configuration based on the proposed decomposition method. The proposed fully re-configurable VLSI architecture of PE array can be illustrated in Fig.7.(b). There are totally  $64 \times 2$  parallel four-pixel PEs in this architecture. Every four PEs mapped to support the same sub-window forms a PE group. Each PE group is connected to a 5x4 systolic array. Different SAD calculation can be implemented by the different configuration of the PE group at each level. In the process of IME, a row of reference pixels are read from search window memory and shifted into the 5x4 systolic array. Reference pixels can move upward, downward and rightward. Current and reference pixels are fetched into the PE array simultaneously to compute the SAD.

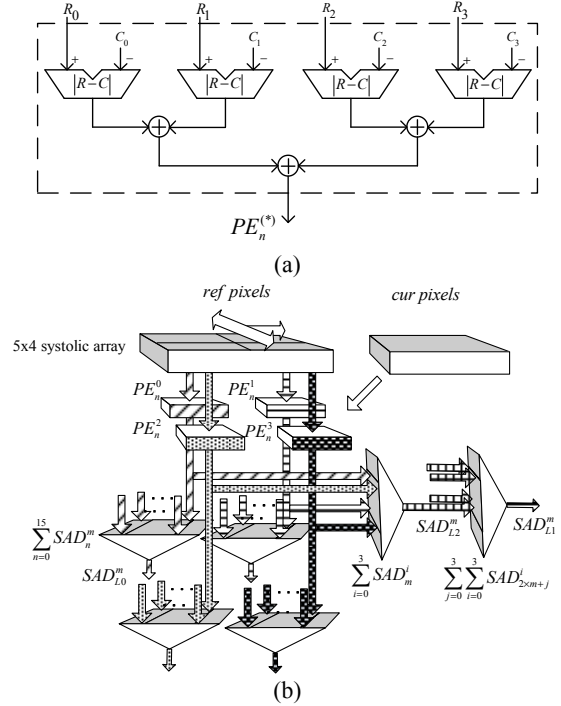


Fig.7. (a) Basic four-pixels PE. (b) Fully re-configurable PE array structure.

### 3.4. Processing Flow and Data Latency

The processing flow of the proposed three-level architecture is shown in Fig.8. At the beginning of coding each frame, the frame level parameters are configured by MCU through system bus as shown in Fig.8. Frame level parameters include height and width of the video frame, search range of each level, enabled VBS modes, forward/backward frame availability, etc. After the frame level parameters are loaded, MCU will start the encoder pipeline controlled by the MB controller as shown in Fig.8. The MB controller starts at idle. It will be activated by the “start” signal from the MCU. Then the MB controller will send its “command” to every module in the pipeline through a local bus as shown in Fig.8. The “command” includes MB level parameters which include the information about the current MB position, MB boundary availability, etc. IME top controller will start search after receiving the “command” from the MB controller. A level controller as shown in Fig.8 will generate the address in the current valid search window called logic address and shift information which describes the searching direction for the MV candidate. The output of each level controller is multiplexed by a MUX as shown in Fig.8 allowing hardware resource reuse. In order to support data reuse scheme that brings significant increase of complexity in the search window management but simplify the address generation logic, a window address mapping module is designed to perform the transformation from the logic address in the valid search window to the physical address in the RAMs. Shift information will be sent to a delay chain

to synchronize with the pixel processing. Finally, SAD and MVD cost will be sent into the cost compare tree. The optimal MV candidates will be selected by the RDO-based matching criterion. And the search is performed in a sequential order from the higher level to the lower level as described in Section 2.

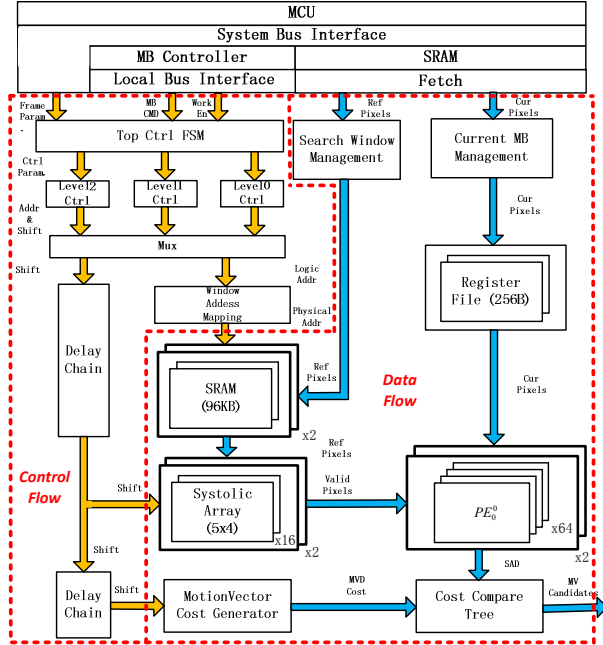


Fig.8. Process flow of three-level MMEA architecture.

In order to get low data latency, internal logics are arranged as parallel as possible on the premise of data dependency. Detailed cycle consumptions of proposed MMEA architecture are illustrated in Fig.9.

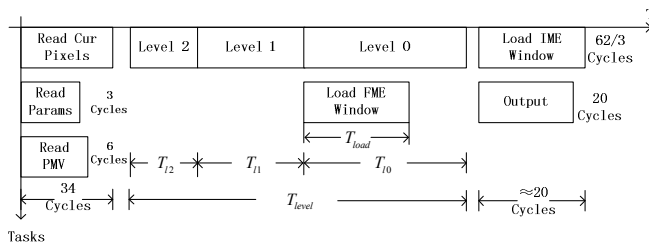


Fig.9. Detail cycle consumptions.

Here,  $T_{12}$ ,  $T_{11}$ ,  $T_{10}$  are cycle consumptions for each level. And  $T_{load}$  is the cycle consumption in loading search window for FME.

Because FME window loading operation can be performed at the end of Level 1, it can be performed in parallel with the Level 0 search with the constraint of  $T_{load} \leq T_{10}$  [8]. Therefore, in our proposed MMEA architecture, the cycle consumptions of the three levels can be described as follows:

$$T_{12} = \frac{2 \times srx / 4}{4} \times \frac{2 \times sry / 4}{4} \quad (1)$$

$$T_{11} = (2 \times srx^{l1} + 1) \times (2 \times sry^{l1} + 1) \quad (2)$$

$$T_{10} = (2 \times srx^{l0} + 1) \times (2 \times sry^{l0} + 1) \quad (3)$$

$$T_{level} = T_{12} + T_{11} + T_{10} \quad (4)$$

Here  $T_{level}$  is the total cycle consumption of the three levels.

As HF2V3 scan mode level C+ data reuse scheme which requires zigzag coding pattern is adopted in our design to reduce external memory, IME loading operation will occur every three MBs. So the cycle consumption in loading IME window is 62/3 in average.

From the results of the previous work [5], in order to get a better rate-distortion performance, the search size of the three levels is :

$$\left\{ \begin{array}{l} \text{The whole search window: } srx = 128, sry = 96 \\ \text{Search window at level 1: } srx^{l1} = 7, sry^{l1} = 6 \\ \text{Search window at level 0: } srx^{l0} = 10, sry^{l0} = 7 \\ \text{Load window for FME: } srx^{load} = 20, sry^{load} = 12 \end{array} \right.$$

Based on the data latency analysis above, we can get the total data latency  $T_{total} = 756$ .

#### 4. IMPLEMENTATION AND COMPARISONS

In this section, we will give out the implementation results of the proposed MMEA architecture and the comparisons with other works.

Our proposed MMEA architecture is implemented in SMIC 0.18μm CMOS technology. The total logic gate count is 920K and 192KB on-chip memory size with maximum operation frequency of 200MHz. Our design can support real-time encoding of 1080P@30fps videos with two reference frames and the maximum search ranges are ±128 pixel horizontally and ±96 vertically.

First, circuit is shared at each searching level as a result of the fully reconfigurable PE array which determines the final gate count consumption. Only 64×2 four-pixels PE units are employed in our architecture. Therefore, 920K is totally enough for our proposed two reference MMEA architecture.

Second, all internal modules are arranged as parallel as possible on the premise of data dependency. The major data latency is caused by the searching process. Hence, parallel SAD computation and reasonable search range are used to reduce the data latency.

Third, our work uses HF2V3 scan mode level C+ data reuse scheme to reduce external memory bandwidth. And it leads to large on-chip memory consumption. So our work needs large memory size compared with [1], [3].

**Table 1** Cost comparison with previous arts.

| Designs                | Proposed       | Huang [4]  | Liu[5]            | Chen[1]              | Deng[7]  |
|------------------------|----------------|------------|-------------------|----------------------|----------|
| Video Spec.            | 1080P@30fps    | 720P@30fps | 1080P@30fps       | 720P@30fps           | SD@30fps |
| Ref. Number            | 2              | 4          | 1                 | 1                    | 1        |
| Search Range           | 256×192        | 128×64     | 196×128           | 128×64               | 65×65    |
| Number of PEs          | 64×2(4-pixels) | N/A        | 2048              | 128×8                | 16×16    |
| Data Latency(Cycles)   | 756            | N/A        | 960               | 1536                 | 5216     |
| On Chip Memory(KB)     | 96×2           | 34.72      | 40<br>(dual port) | 13.71<br>(dual port) | 62       |
| Technology(μm)         | 0.18           | 0.13       | 0.18              | 0.18                 | 0.18     |
| Gate Count(K)          | 460×2          | 992.8      | 486               | 305                  | 210      |
| Working Frequency(MHz) | 200            | 108        | 200               | 108                  | 260      |
| R (Mspps)              | 24065          | 3539       | 6142              | 885                  | 205      |
| HER                    | 25             | 3.6        | 12.6              | 2.9                  | 0.98     |

For the comparison of different ME architectures, we introduce the hardware efficiency ratio [6] (HER), which can be expressed by the ratio of through-put rate R and required silicon area A.

$$HER = \frac{R}{A} \quad (5)$$

For the IME architectures, R can be described by mega search points per second (Mspps). The mentioned IME designs all have the similar rate distortion degradation compared with FSBM. So we use their equivalent FSBM to simplify the calculation of R. And silicon area A is evaluated by gate count. Hence, HER represents mega search points per second per thousand gates.

Table 1 shows the total implementation cost of the proposed design and the comparison with previous arts. Among all architectures, the proposed architecture can provide the highest through-put, meanwhile, it has the best hardware efficiency ratio benefiting from the fully re-configurable PE array.

## 5. CONCLUSION

In this paper, a hardware-efficient MMEA VLSI architecture is designed based on the fully re-configurable PE array. The On-chip storage structure is designed to meet challenges of high memory bandwidth and efficient data reuse. Fully re-configurable PE array is proposed for highly parallel SAD computation and hardware resource sharing. According to the implementation result, 920K gate count is required for real time coding 1080p@30fps with 2 reference frames under the operation frequency of 200MHz.

## 12. REFERENCES

[1] T.-C. Chen, S.-Y. Chien; Y.-W. Huang, C.-H. Tsai, C.-Y. Chen, T.-W. Chen, L.-G. Chen, "Analysis and architecture design of an HDTV720p 30 frames/s

H.264/AVC encoder," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 6, pp. 673–688, Jun. 2006.

[2] Wiegand T., Sullivan G.J., Bjontegaard G., Luthra, A., "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[3] Y.-W. Huang, T.-C. Chen, et al, "A 1.3 TOPS H.264/AVC single-chip encoder for HDTV applications," in *IEEE ISSCC Dig.Tech. Papers*, pp128-129, Feb.2005.

[4] Z.Y. Liu, Y. Song, M. Shao, S. Li, L.F. Li, Ishiwata, S., Nakagawa, M., Goto, S., Ikenaga T., "HDTV 1080P H.264/AVC encoder chip design and performance analysis," *IEEE J. Solid-State Circuits*, vol. 44, no. 2, 816 pp. 594–608, Feb. 2009.

[5] H. B. Yin, H. Z. Jia, H. G. Qi, X. H. Ji, X. D. Xie, W. Gao, "A Hardware-Efficient Multi-Resolution Block Matching Algorithm and Its VLSI Architecture for High Definition MPEG-Like Video Encoders" *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 9, pp. 1242–1254, Jul. 2010.

[6] L. Deng, W. Gao, M. Z. Hu, Z. Z. Ji, "An efficient hardware implementation for motion estimation of AVC standard," *IEEE Trans. Consumer Electron.*, vol. 51, no. 4, pp. 1360-1366, Nov. 2005.

[7] C. Y. Chen, C. T. Huang, Y. H. Chen, L. G. Chen, "Level C+ Data Reuse Scheme for Motion Estimation With Corresponding Coding Orders," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 4, pp. 553–558, Apr. 2006.

[8] H. B. Yin, L. Deng, H. G. Qi, W. Gao, "VLSI Friendly ME Search Window Buffer Structure Optimization and Algorithm Verification for High Definition H.264/AVS Video Encoder," in *Int. Conf. Multimedia and Expo*, pp. 1098-1101, Jun. 2009.