High Throughput VLSI Architecture for Multiresolution Motion Estimation in High Definition AVS Video Encoder

HaiBing Yin*[ab], Honggang Qi[b], Hao Xu[b], Don Xie[b], Wen Gao[b]
[a]Dept. of Information Engineering, China Jiliang Univ. Hangzhou, China;
[b]National Engineering Laboratory for Video Technology, Peking Univ. Beijing, China

ABSTRACT

This paper proposes a hardware friendly multiresolution motion estimation algorithm and VLSI architecture for high definition MPEG-like video encoder hardware implementation. By parallel searching and utilizing the high correlation in multiresolition reference pixels, huge throughput and computation in motion estimation due to large search window are alleviated considerably. Sixteen way parallel process element arrays with configurable multiplying technologies achieve fast search with regular data access and efficient data reuse. Also, the parallel arrays can be efficiently reused at three hierarchical levels for sequential motion vector refinement. The modified algorithm achieves a good balance between complexity and performance. Also, the logic circuit and on-chip SRAM consumption in the VLSI architecture are moderate.

Keywords: Multiresolution motion estimation, VLSI, Architecture, Video coding

1. INTRODUCTION

AVS is a MPEG-like audio and video standard of China. Its video part (AVS-P2) was finalized and accepted as the national standard in 2006. AVS video standard achieves equivalent coding performance compared with H.264/AVC with lower complexity and small quality loss. The industrialization for the AVS standard is being on and leaded by the AVS industry alliance.

Dedicated AVS video encoder chip is highly desired for consumer applications such as DTV and PVR for AVS standard industrialization. Motion estimation (ME) is the most complex module in MPEG-like video encoder. Real-time ME implementation in high definition (HD) video encoder is challenging due to not only large search window (SW) to cover, bus also new tools such as variable block size motion estimation (VBSME), multiple reference frames, and fractional pixel motion estimation. Cost-efficient ME engine is highly desired for HD AVS video encoder [1].

Full search block matching (FSBM) algorithm is widely used for ME hardware design due to superior performance and high regularity. However, these advantages are challenged in HD cases due to the large SW. Directly multiplying process element arrays with redundant structure was widely adopted to alleviate the throughput problem in FSBM based ME architecture [1]. As a result, the hardware cost is high and unacceptable. Many fast ME algorithms were proposed in the literature, but they are usually ill-suited for VBSME hardware implementation due to performance degradation, complex control or irregular memory access. Hierarchical ME algorithms such as three-step search (TSS), new TSS, and four step search are well-suited for VLSI implementation with high regularity and fast search speed. However, they suffer from considerably serious performance degradation. As far as HD video encoder is considered, multiresolution ME algorithm (MMEA) is a better solution for VLSI implementation to cover the large SW with good balance between performance and complexity. This paper focuses on cost-effective VLSI architecture combined with MMEA tailor and optimization.

The rest is arranged as follows. Hardware oriented MMEA are reviewed and proposed in section 2. The proposed VLSI architecture based on MMEA is proposed in section 3. Simulation and conclusion are drawn in section 4.

2. HARDWARE ORIENTED MMEA

2.1 Challenge Analysis on MMEA

In typical three-level MMEA [2], hierarchical refinements are performed from the coarsest level to the finest level successively. Although complexity and throughput are greatly reduced in MMEA, the resulting performance degradation is not negligible due to downsampling. Multiple candidate refinement centers are widely combined with MMEA to avoid

being trapped into local minima. This measure improves the accuracy to some extent indeed. But, the advantages are challenged if rate distortion optimization (RDO) based VBSME and high throughput in HD cases are both considered.

First, VBSME contribute to the performance superiority of H.264 and AVS remarkably. How to achieve efficient VSBME combined with MMEA is a very important problem. In blocks with size no larger than 8x8, their downsampled versions at coarse levels are very small. For example, at the coarsest level only 4 pixels attend in SAD (sum of absolute difference) calculation in 8x8 blocks. Too small pixels in small blocks at the coarsest level result in SAD incredibility. Thus, VSBME is ill-suited to be implemented at coarse levels. In our MMEA, VBSME is performed only at the finest level restricted within a local search window (LSW) centered about the winner motion vector of the middle level. The position and size of LSW is very important to sustain the superiority of VBSME.

Second, huge throughput in ME is the largest challenge in HD cases. The cycles consumption in MMEA based IME engine will be very high if no parallelism is adopted. That is ill-suited for cost-effective hardware implementation. Thus, multiplying the process element (PE) arrays is indispensible for real-time pipelining. Simple multiplying PE arrays results in dramatically increased circuit area and complex data control. Thus, more efficient parallelism of PE arrays is highly desired.

Third, RDO based ME is recommended in AVS and H.264 although it is not forced. A weighted SAD cost function WSAD including SAD and motion vector (MV) coding bit consumption are jointly considered using Lagrange optimization theory. In MMEA, the MV coding bit measure at coarse levels is not credible relatively. This perturbation aggravates the uncertainty of WSAD function at the coarsest level, possibly resulting in being trapped into local minimum.

Lin et al. proposed a parallel MMEA [4] in which three hierarchical levels were simultaneously searched with parallel PE arrays associated with individual on-chip search window buffers. The inspiring fast search speed is achieved at cost of performance degradation in the sequences with complex motion. More importantly, data access between SDRAM and on-chip search window buffers of three levels become complex and irregular because the finest level is centered about a varying predictive MV. Other MMEA architecture all performs MV refinement from the coarsest to the finest level sequentially. The later kind is also adopted in this work, and measures are taken to face three major challenges analyzed above.

2.2 The proposed MMEA

The proposed three-level MMEA is performed from the coarsest level to the finest level sequentially. Direct down-sampling is used instead of low-pass filtered downsamping [3] for data and circuit reuse among adjacent levels. 256 pixels in the original MB and all reference pixels in the whole SW are decomposed into three resolutions and 16-way interlaced groups. The original MB is taken as example for downsampling illustration here.

The undownsampled 256 pixels in the original MB (the finest level L_0) are shown in Fig.1-(a). They are 4:1 downsampled into four 8x8 blocks (the middle level L_1) indexed by m and n. In order to simplify illustration, the pixels in four 8x8 blocks are respectively marked using different symbols:×(mn=00),●(mn=01),▲(mn=10), and ■(mn=11). Similarly, each 8x8 block at level L_1 is 4:1 downsampled into four 4x4 subblocks (the coarest level L_2) indexed by p and q. The pixels in four 4x4 subblocks of each 8x8 block are marked using red, blue, green and black colors respectively. As a result, the original MB at level L_0 is downsampled into sixteen interlaced subblocks marked using different symbols and colors. The three-level downsampling and the indices (m, n, p, q) are shown from Fig.1-(a) to Fig.1-(e). Similarly, the whole reference SW is also downsampled into sixteen interlaced reference sub-SW (SSW) indexed by the indices (m, n, p, q).

The proposed three-level MMEA is illustrated in Fig.2 and described as follows. Suppose the whole integer pixel SW is [-SRx, SRx] x [-SRy,SRy], our target SW is 256x192 (SRx=128 and SRy=96). The small SW [-32,32]×[-32,32] is used for example here due to display resolution limitation. MV refinement in integer pixel ME (IME) is achieved by three successive hierarchical stages described as follows.

First, FSBM is performed at IME stage 1 to check all candidate MVs at level L_2 shown using black points in Fig.2-(a) to cover the whole SW. To accelerate the search speed, 16-way parallel motion searches are employed using 16 interlaced downsapled pixel samples. The 16-way parallel searches are performed by 16-way process element arrays (PEA) PEA_{mnpq}. In each PEA, there are sixteen process elements to perform SAD calculation of 4x4 pixels in each subblock. All candidate MVs at level L_2 (block points) are divided into 4x4 subareas indexed by $mnpq$ associated with the indices

in Fig.1-(b) to (e). The PEA_{mnpq} is employed to implement motion matching for the subarea indexed by *mnpq*. The 16 PEA modules can achieve the throughput of 16 candidate MVs in each cycle at level L_2. In order to accelerate the search speed, each PEA module PEA_{mnpq} can be multiplied cloned to achieve fast convergence and improve data reuse further. Suppose each PEA_{mnpq} are cloned and multiplied by T way parallel structure, the throughput at level L_2 can achieve 16T candidate MVs in each cycle. The parallelism intensity T is configurable according to the image resolution and the system clock frequency.

Second, MV refinement at level L_1 is performed at IME stage 2. Three winner MVs obtained at level L_2 are kept as the following refinement centers. There are strong spatiotemporal correlations in the motion field, thus we can estimate a predictive MV for the current MB according the MVs of the adjacent MBs coded using the MV correlation. The predictive MV is used as the fourth refinement center to compensate for the disfunction of level L_2 search due to downsampling. IME stage 2 is illustrated in Fig.2-(b), in which four-way full searches at level L_1 are performed within a local small SW with size of $[-SRx_{L1}, SRx_{L1}]$ x $[-SRy_{L1},SRy_{L1}]$ centered about four center MVs by employing four PEA subsets (PEAS) $PEAS_{mn}$ respectively. Similarly, T-way doubled structure is also applied in $PEAS_{mn}$ at level L_1. The 16T PEA modules are combined to 4T PEAS modules resulting in the throughput of 4T candidate MVs in each cycle at level L_1.

Third, VSBME is performed at IME stage 3 at level L_0 only within a well-selected local SW (LSW) with size of $[-SRx_{L0},-SRx_{L0}]$ x $[-SRy_{L0},-SRy_{L0}]$. The center of LSW is determined by successive MV refinements at level L_2 and L_1. Although VSBME is performed within LSW instead of the whole SW, the resulting performance degradation is negligible due to the high MV correlation existing in the different size blocks of the same MB if the LSW size is large enough [8]. The 16T PEA modules are simultaneously employed to construct T way PEAS arrays (PEASA) achieve the throughput of T candidate MV in each cycle at level L_0.

The cycle consumptions of three levels in the proposed MMEA are listed as follow.

$$cycle_{\mathrm{IME}} = \frac{(2\times SRx)\times(2\times SRy)}{16\times16\times\mathrm{T}} + \frac{(2\times SRx_{L1})\times(2\times SRy_{L1})}{\mathrm{T}} + \frac{(2\times SRx_{L0})\times(2\times SRy_{L0})}{\mathrm{T}} \quad (1)$$

In our MMEA, the SW size parameters are as follows: SRx =128, SRy = 96, $SRx_{L1} = SRy_{L1} = 8$, and $SRx_{L0} = SRy_{L0} = 16$. The candidate MVs are assigned to T way parallel structures at three levels, the mapping relationship and the scan order with efficient data share are given in Fig.3. The total cycles consumed for each MB IME are approximately 736 and 368 in the case of T=2 and T=4 respectively for example. This quick search speed is very important and meaningful for low-power hardware implementation of video encoder with HD resolution above 1080P.

Moreover, on-chip SRAM consumption for reference SW buffer is also reduced. In traditional architecture, dual-port SRAM is adopted to implement the SW buffer for data share between IME and FME. In our architecture, the reference pixels in the LSW at level L_0 instead of the whole SW are simultaneously transferred to a double-buffered LSW buffer for FME during level L_0 search stage. As a result, single-port SRAM is used for the whole SW buffer in our architecture instead of dual-port SRAM. Dual-port SRAM consumes doubled logic gates compared with single-port SRAM. Thus, single-port SRAM structure SW is highly preferred.

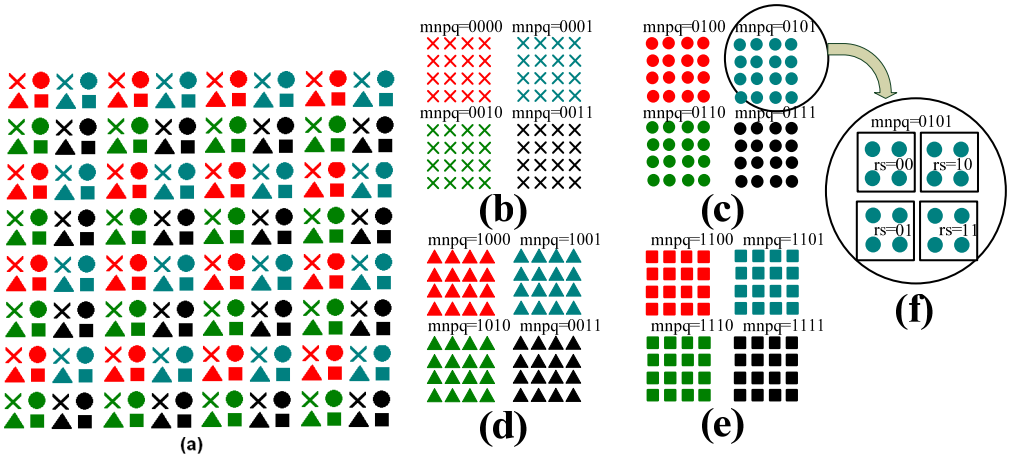

Figure1. The relationship among 16-way pixel samples of three levels.

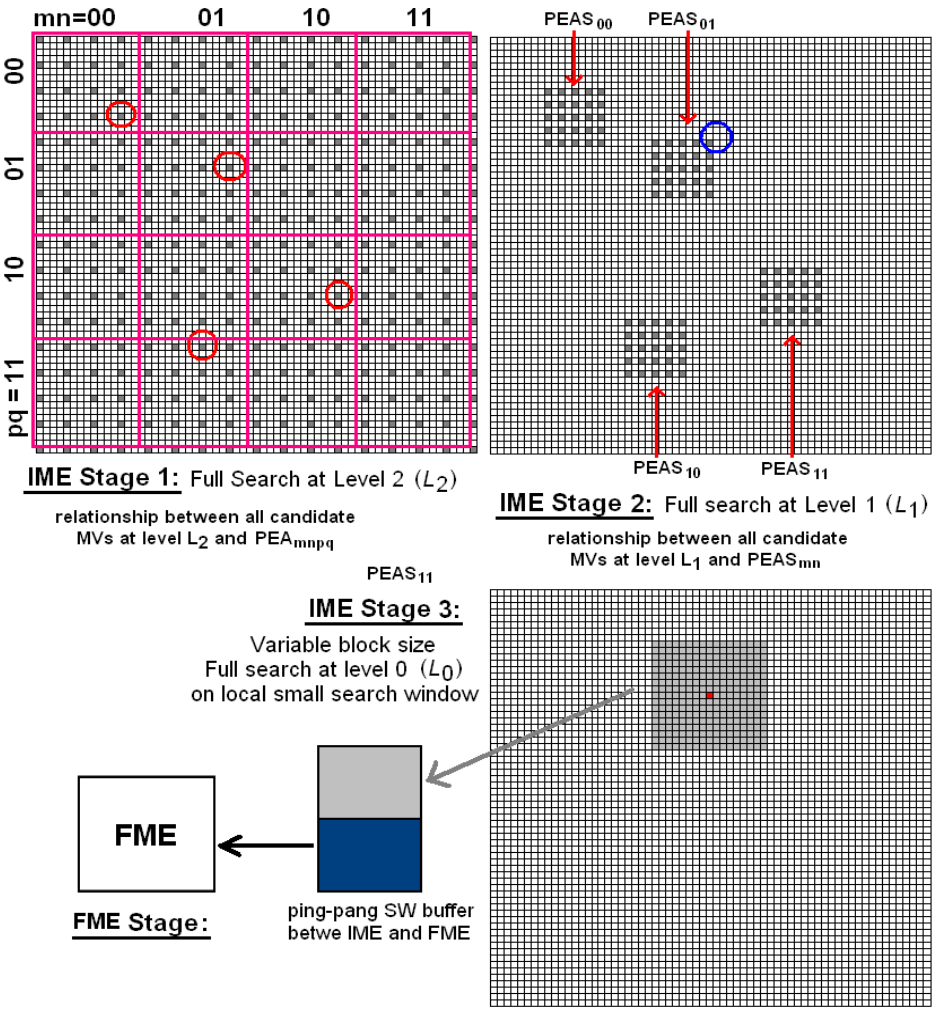

Figure 2. Three level hierarchical integer MMEA.

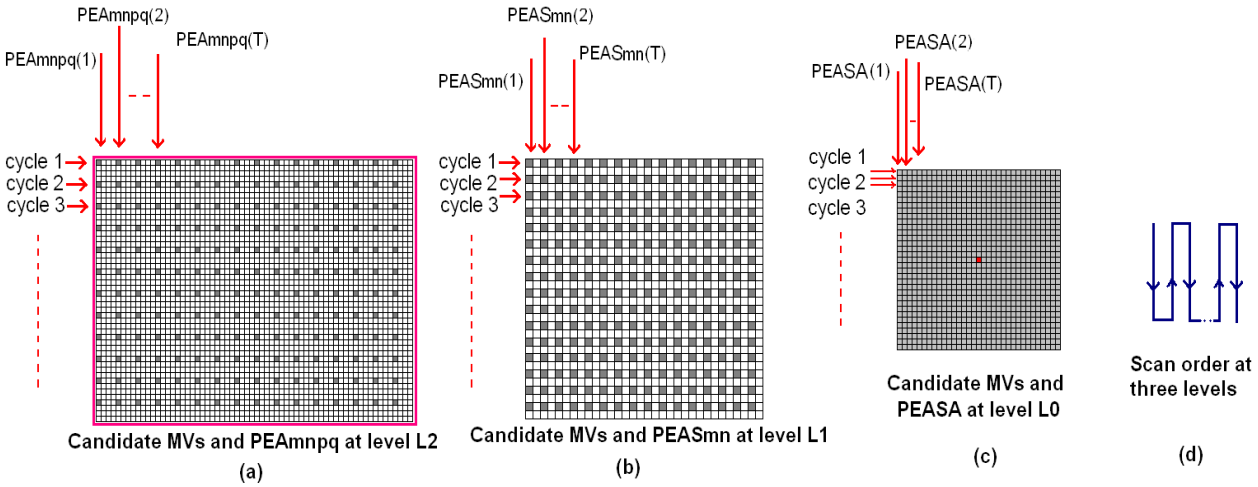

Figure 3. The candidate MVs and their corresponding hardware search units.

3. THE PROPOSED VLSI ARCHITECTURE

Three level MV refinements are performed successively. To improve the hardware utilization, we propose a highly reused PE array structure to achieve 100% hardware reuse at adjacent three level stages. The basic unit for motion estimation is process element (PE), and each PE performs SAD calculation for one pixel. There are totally 256T parallel PEs divided into 16 groups named PE arrays (PEA) indexed by mnpq and t. Each basic PEA is labeled as PEAmnpq(t), mnpq is the sownsapling pixel indices, and t is the index of the parallel structure cloned and multiplied (t = 1, 2, ..T). The VLSI architecture of PEAmnpq(t) is given in Fig.4. The task of PEAmnpq(t) is to calculate SADmnpq for the 4x4 block indexed by t shown in Fig.3 and mnpq shown form (b) to (e) in Fig.1, which is 16:1 downsampled from the finest level L_0. So, 16T parallel PEA modules are adopted for full search at level L_2.

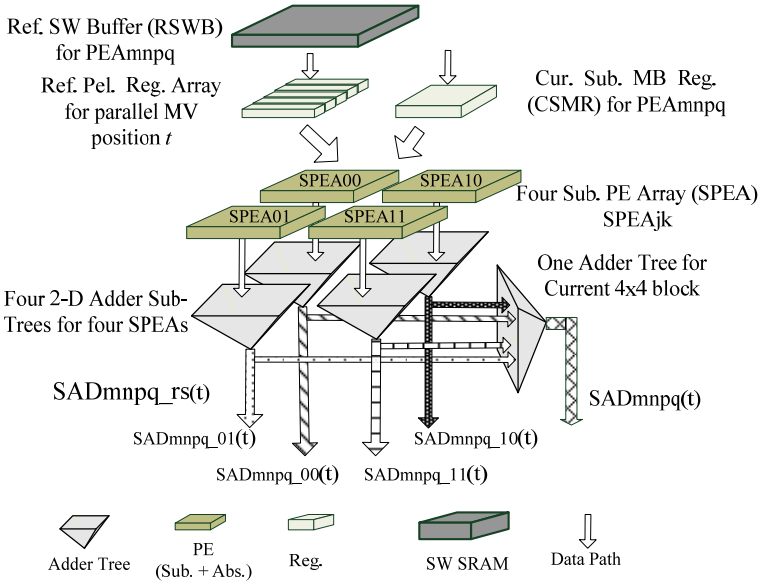

Figure 4. The proposed micro-architecture of PEAmnpq(t).

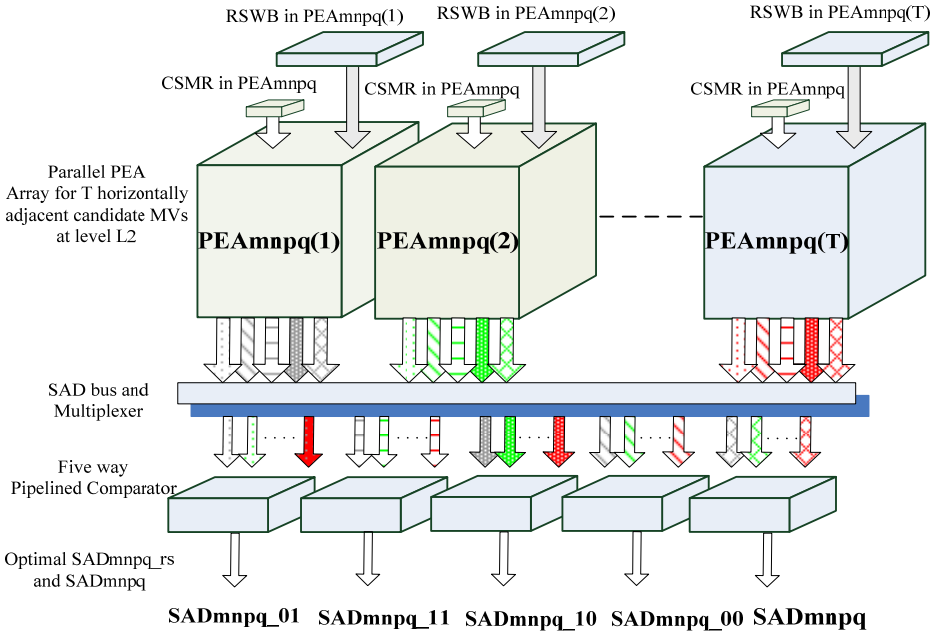

Figure 5. The architecture of T-way parallel PEAmnpq for level L_2 Full Search.

The smallest MB partition mode for VSBME is 8x8 in AVS. Although H.264 supports the smallest block with size of 4x4, only blocks with size of no smaller than 8x8 are supported in typical H.264 video encoder VLSI architecture in the HD cases [3]. The resulting performance degradation is negligible due to lower possibility that nontranslational motion occurs in 8x8 blocks in HD cases. Thus, the proposed architecture is also well-suited for H.264 if only blocks no smaller than 8x8 are adopted.

To fully utilize the PEA structure for VSBME at level L_0, we employ four subblock PE array (SPEA) indexed by rs in every PEA. The SPEArs is used to calculate SADmnpq_rs of 2x2 pixels shown in Fig.1-(e) and Fig.4. The 16 pixels in the current original 4x4 block mnpq are stored in the *Cur. Sub. MB Reg.* (CSMR). The 16:1 downsampled luminance reference pixels are stored in the *Ref. SW buffer* (RSWB). The *Ref. Pel. Reg. Array* with size of 5x5 pixels is employed to load reference pixels from RSWB for four SPEA modules in the current PEA instantaneously. Left, right, and up shift operations are supported in the *Ref. Pel. Reg. Array* for data share among the adjacent MVs. All MVs at level L_2 are searched by 16T-ways parallelism as shown in Fig.3-(a). The architecture of T-way parallel PEAmnpq for level L_2 full search is shown in Fig.5. In each cycle, one locally optimal MV with the smallest WSADmnpq (SADmnpq plus the MV coding bit cost) is selected from T horizontal adjacent candidate MVs as shown in Fig.3-(a), and all locally optimal MVs are compared in a pipelining manner to obtain the final optimal MV at level L_2.

At the level L_1 stage, four PEA modules (PEAmn00, PEAmn01, PEAmn10, and PEAmn11) are combined to implement one PEA subset (PEAS) PEASmn(t) as shown in Fig.6. PEASmn(t) is employed to calculate the SADmn for 8x8 blocks at level L_1 as shown in Fig.1. As a result, 16T parallel PEA modules are mapped to 4T way PEAS modules to achieve 4T-way parallel search at level L_1 shown in Fig.3-(b). Similarly, SADmn_rs is calculated by SAD reuse in PEASmn(t) for VSBME implementation desired at level L_0. The proposed architecture of PEASmn for level L_1 local full search is shown in Fig.7. In each cycle, one locally optimal MV with the smallest WSADmn (SADmn plus the MV coding bit cost) is selected from T horizontal adjacent candidate MVs at level L_1 as shown in Fig.3-(b), and all locally optimal MVs are compared in a pipelining manner to obtain the final optimal MV at level L_1.

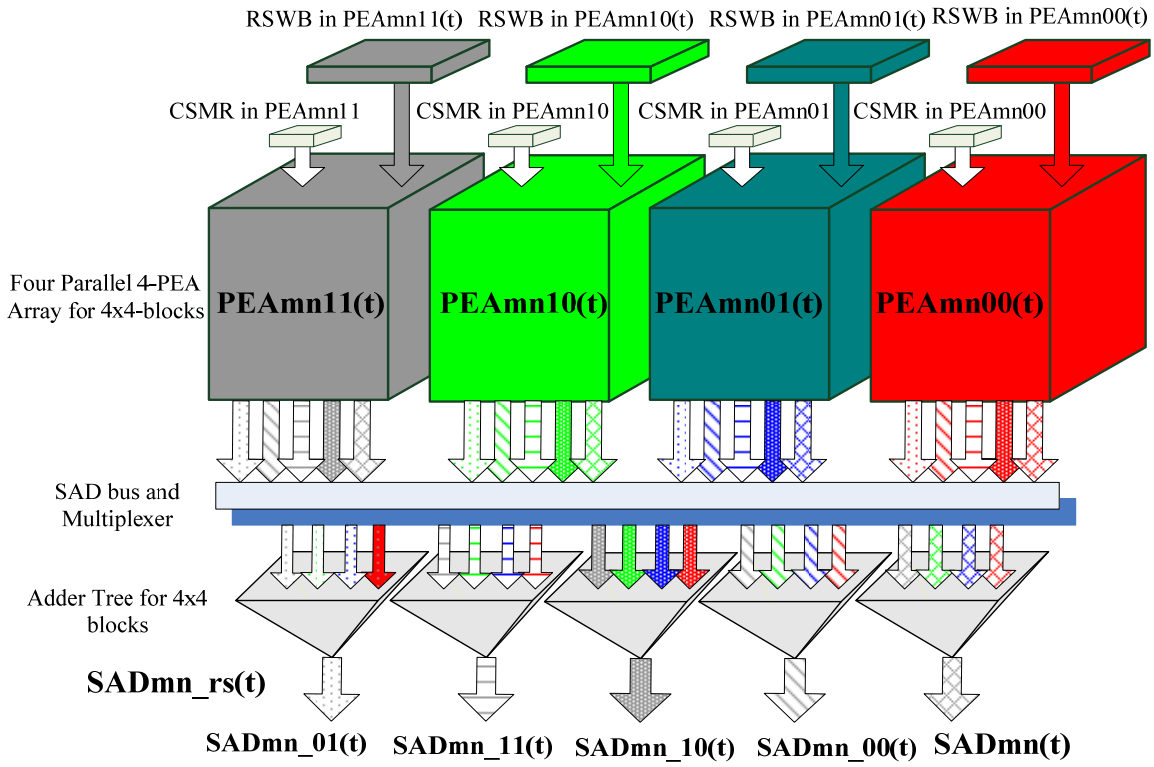

Figure 6. The proposed mirco-architecture of PEASmn(t)

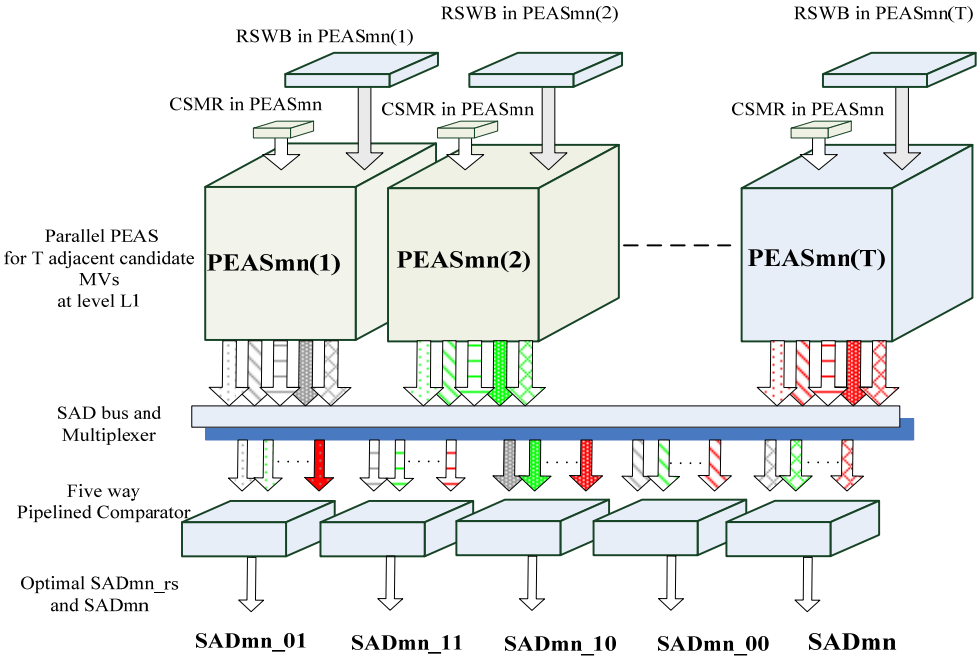

Figure 7. The proposed architecture of PEASmn for level L1 local Full Search.

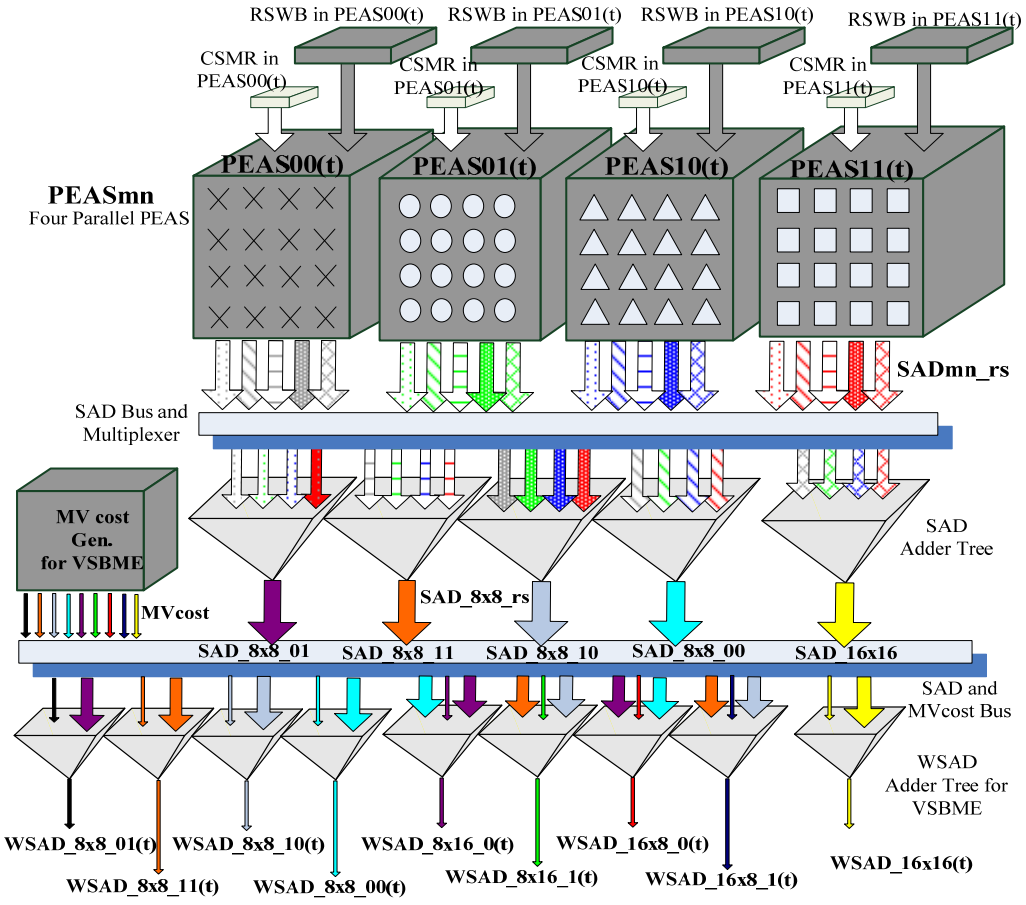

Figure 8. The PEAS Array (PEASA) PEASA(t) Structure for VSBME at level L0.

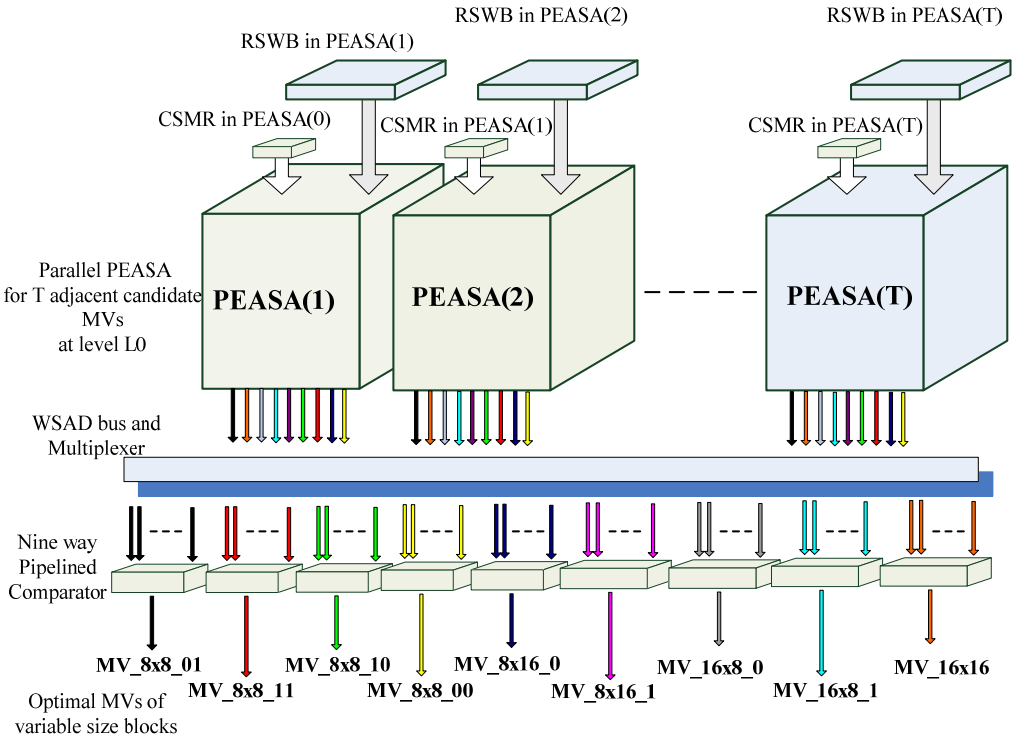

Figure 9. VSBME at level L0.

At the level L_0 stage, four PEAS modules ($PEAS_{00}$, $PEAS_{01}$, $PEAS_{10}$, and $PEAS_{11}$) are combined to form the PEAS Array (PEASA) to calculate the SAD of different blocks for VSBME implementation. The structure of PEASA(t) is shown in Fig.8. The MV cost Gen. calculates the MV costs for all blocks with different block sizes. SAD_{mn_rs} are summed up by the SAD adder tree to obtain $SAD_{\_8x8\_rs}$. Then, $SAD_{\_8x8\_rs}$ are reused and summed up with the MV costs by the WSAD adder tree to obtain the weighted SAD (WSAD) of different size blocks. Here, WSAD is just the criterion MEcost mentioned in section 2.1 for RDO based ME. The optimal integer MVs of all MB partition modes are finally selected by the 9-parallel WSAD comparator array. Local full search based on the proposed PEASA architecture is shown in Fig.9 for level L_1 VBSME implementation. In each cycle, one locally optimal MV with the smallest WSAD is selected from T horizontal adjacent candidate MVs at level L_0 as shown in Fig.3-(c), and all locally optimal MVs are compared in a pipelining manner to obtain the final optimal MV of the blocks with different size at level L_0.

4. SIMULATION RESULT AND CONCLUSION

The hardware friendly motion estimation algorithms such as full search block matching (FSBM) [1], NTSS, MMEA [2], MMEA [3], and their architectures are used as references. Identical SW and coding parameters are used in all reference algorithms for fair comparison. The AVS Jizhun profile with rate distortion optimized VBSME is used for performance evaluation. Eight 720P sequences *city*, *Spincalendar*, *Sailormen*, *Crew*, *Syclists*, *Optis*, *Harbour*, and *Night* with diffenet motion characteristics are used for simulation, and the integer pixel SW is [-128,128]× [−96,96]. The PSNR results of the proposed MMEA versus the anchor FSBM [1] are given in Fig.10-(a) and Fig.10-(b).

According to Fig.10, the PSNR degradation of the proposed MMEA versus FSBM is small. The average PSNR degradation is smaller than 0.1dB. The worst case occurs in the *Spincalendar* sequence with complex motion and camera circumrotation, and the worst loss is approximately 0.15dB.

The proposed IME architecture was implemented by Verilog-HDL language and synthesized by Design complier with SMIC 0.18μm 1P6M standard cell library. Table 1 shows the total hardware cost of the proposed IME design and comparison to other typical reference designs. The throughput (reference frame number, search range, inter modes supported), PE number, gate count, SRAM consumption for IME and FME, external memory bit width, and frequency are also taken as comparison factors.

According to the data in table 1, the MMEA [4] is designed for small size video coding application with search range 32x32. The architectures in FSBM [1] and reference [2] also consume huge PE units although only baseline H.264 with only one reference frame is supported. Inspiring on-chip memory and throughput were achieved with 128 bit external memory bus in the three-level parallel MMEA based IME architecture [4]. However, these two advantages are challenged by performance degradation in the case of sequences with highly regular motion and irregular data share between IME and FME. It was reported that 90% data share was achieve in [4]. However, the missing in the case of direct mode in B frame would be relatively high in main profile H.264 and Jizhun profile AVS. As a result, the data share between IME and FME and bandwidth control for external memory become complex. Multiple reference frame and data share between IME and FME were not considered in MMEA [3].

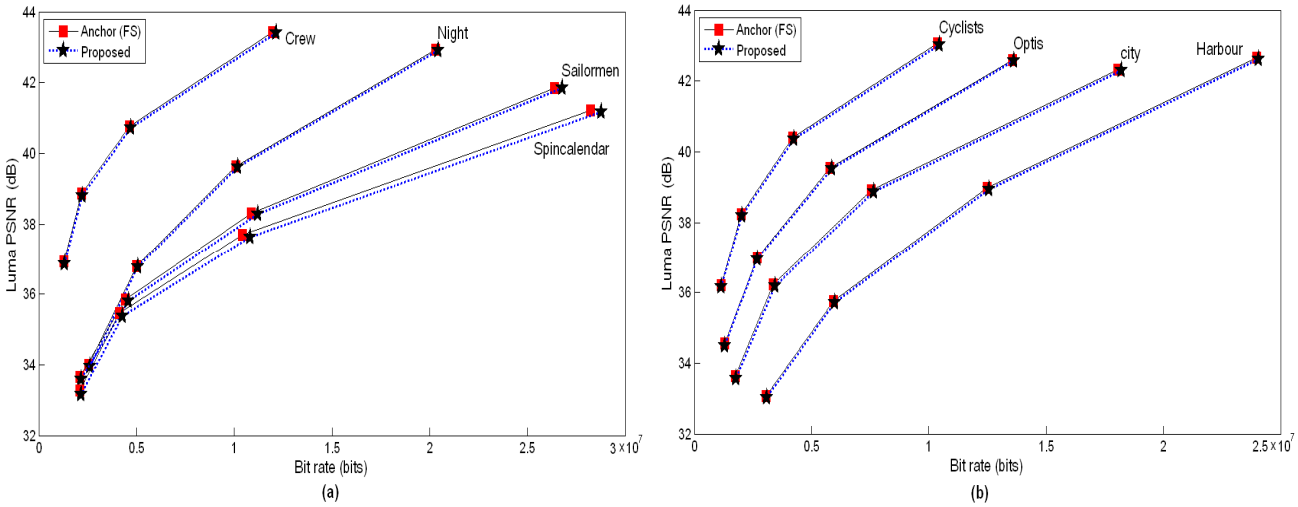

Fig.10. The PSNR curves of the proposed MMEA versus FSBM.

Based on hardware-oriented algorithm optimization on MMEA, the proposed IME architecture achieve relatively better trade off among the multiple factors including gate consumption, SRAM consumption, throughout, and memory bandwidth. Main profile H.264 and Jizhun profile with B frame and two reference frames in P frames are supported in the proposed architecture. 512T PE units are enough for two reference frame motion estimation achieving the throughput of 1472, 736, and 368 cycles per MB in the case of T=1, T=2, and T=4. Due to local buffer management for the adjacent PEAmnpq(t) are identical, the *Ref. Pel. Reg. Array* modules in each PEAmnpq(t) can be integrated for data share due to the identical data flow control, and all top level control and buffer management is identical. Thus, the additional circuit consumption increase due to T is also moderate. As a result, only 130K, 190K, and 250K gate is enough for one reference frame motion estimation in the case of T=1, T=2, and T=4, and correspondingly doubled gate is enough for main profile H.264 and jizhuan profile AVS with B frame support.

More inspiringly, efficient on-chip SRAM structure is adopted in the proposed architecture for data share between IME and FME. Single-port SRAM is used for on-chip SW buffer instead of dual-port SRAM for IME. Dual-port SRAM is only used for the LSW buffer to achieve efficient data share between IME and FME. Thus, the SW buffer SRAM consumption is saved up to almost 50% [8], which is very inspiring for HD video encoder chip especially for AVS bidirectional ME support.

Table 1. Hardware Cost Comparision on IME Cycle, PE Number and SW Memory.

Symbol	Wu [5]	Lee [6]	Chen[1]	Lin[4]	Liu [3]	Proposed
Video Spec.	CIF @ 30fps	720x480@30fps	720p@30fps	1080p@30fps	1080p@30fps	1080p@30fps
Algorithm	MMEA	MMEA	FSBM	MMEA	Subsampling	MMEA
Profile (frame type)	Baseline(P)	Baseline (P)	Baseline (P)	Baseline (P)	Baseline (P)	Main (P, B) Jizhun AVS
Ref. frame number	1	1	1	1	1	2
Search Range	32 × 32	128 × 128	128 × 64	256 × 256	196×128	256×192
Inter Modes	N/A	All	All	All	8×8, 16×8, 8×16, 16×16	8×8, 16×8, 8×16, 16×16
No. of PEs	50	64/320	2048	728	2048	512 (T=1) 1024 (T=2) 2048 (T=4)
Gate count (k)	59	N/A	305	213.7	486	260 (T=1) 380 (T=2) 500 (T=4)
SRAM for IME (kb)	1.3	N/A	13.71(Dual Port)	5.95(Dual Port)	40 (Dual Port)	159.5 (Single port)
SRAM for FME(kb)	N/A	N/A	13.82(Dual Port)	N/A	40.8 (Dual Port)	17(Dual port)
Throughput (cycles/MB)	495	375	1536	256	960	1472 (T=1) 736 (T=2) 368 (T=4)
Frequency (Mhz)	153	16	108	128.8	200	360 (T=1) 180 (T=2) 100 (T=4)
DRAM bit width	N/A	N/A	N/A	128	512	64
Technology	0.18 CMOS	N/A	0.18 CMOS	0.13 CMOS	0.18 CMOS	0.18 CMOS

REFERENCES

1. T.C. Chen et al, "Analysis and Architecture Design of an HD720p 30 Frames/s H.264/AVC Encoder," IEEE Trans. Cir. Syst. Video Tech., vol. 16, no. 6, pp. 673-688, June 2006.
2. B. C. Song et al, “Multi-resolution block matching algorithm and its VLSI architecture for fast motion estimation in a MPEG-2 video encoder,” IEEE Trans. CSVT vol. 14, no. 9, pp.1119-1137, 2004.
3. Zhenyu Liu,Yang Song, Satoshi Goto etc. HDTV 1080P H.264/AVC Encoder Chip Design and Performance Analysis, IEEE Journal of Solid-state Circuits, Vol.44, no.2, Feb,2009.
4. Yu-Kun Lin, Chia-Chun Lin, Tzu-Yun Kuo, and Tian-Sheuan Chang, A Hardware-Efficient H.264/AVC Motion-Estimation Design for High-Definition Video, IEEE Trans. Circuits and System—I: Regular Papers, Vol. 55, no. 6, July, 2008.
5. Bing-Fei Wu, Hsin-Yuan Peng, and Tung-Lung Yu, Efficient Hierarchical Motion Estimation Algorithm and Its VLSI Architecture, IEEE Trans. Very Large Scale Integration Systems, Vol.16, no. 10, pp:1385 – 1398, 2008.
6. J. H. Lee and N. S. Lee, “Variable block size motion estimation algorithm and its hardware architecture for H.264/AVC,” in Proc. IEEE Int. Symp. Circuits Syst., May 2004, vol. 3, pp. 741–744.
7. Calculation of Average PSNR Differences between RD-Curves ITU-T VCEG, 2001, Proposal VCEG-M33.
8. H. B. Yin et al, VLSI Friendly ME Search Window Buffer Structure Optimization and Algorithm Verification for High Definition H.264/AVS Video Encoder, IEEE Conference on Multimedia and Expo (ICME), Cancun, Mexico, June 28-July 3, 2009.