2012 International Conference on Solid-State and Integrated Circuit (ICSIC 2012)
IPCSIT vol. 32 (2012) © (2012) IACSIT Press, Singapore

Efficient Architecture for Variable block size Motion Estimation of H.264 Video Encoder

P.Muralidhar[1 +], C.B.Rama Rao[2], I.Ranjith Kumar[3]
Department of ECE, National Institute of Technology Warangal, India

Abstract. This paper proposes an efficient architecture for implementation of variable block size motion estimation (VBSME). The architecture is able to perform a full motion search on integral multiples of 4x4 blocks sizes. To use the architecture, each 16x16 macroblock of the source frames should be partitioned into sixteen 4x4 non-overlapping subblocks, called primitive subblocks. Proposed architecture based on Propagate Partial SAD and SAD Tree architectures. The motion estimation of all the primitive subblocks are performed in parallel. Because these primitive subblocks can be used to form the 41 subblocks of different sizes specified by the H.264, the VBSME processing element is employed to concurrently compute the sums of absolute differences (SADs) of all the 41 subblocks from the SADs of the primitive subblocks. This new architecture has lower latency and higher throughput over other exiting VBSME architectures. The Architecture was designed using Synopsis DC(design compiler) 0.18 μm CMOS technology**,** synthesis results shows that the architecture can work at 53 frames/s for 1280x720 resolution SDTV video application.

Keywords: H.264,MotionEstimation,Full-Search Block matching(FSBM),VLSIArchitecture,Microblock,Sum of absolute difference(SAD),

1. Introduction

In general video coding systems, motion estimation (ME) can efficiently eliminate temporal redundancy between adjacent frames. At the same time, ME is also regarded as a vital component in a video coder as it consumes a large amount of computation resources. Especially, in the video coding standard H.264, ME accounts for most of the complexity of the encoder for its seven different block sizes and 1/4-pel accuracy fractional pixel search. Therefore, simplifying the ME process is essential for real-time applications.

In order to reduce the computational complexity of the Full search block matching (FSBM) motion estimation, several fast block-matching algorithms, such as three step search [1], new three-step search algorithm [2], the novel four-step search algorithm [3], one-dimensional search algorithm [4], the cross search algorithm [5], the diamond search algorithm [6],hexagon-based search algorithm [7] and block-based gradient descent search algorithm [8] etc., have been proposed. All these algorithms are not optimal in the sense that instead of exhaustive search, only some fixed positions are searched, based on the predictions of motion. Any error in motion prediction may lead to wrong motion vectors, resulting in poor peak signal-to-noise ratio (PSNR) of the motion-compensated frame. Experimental results show that the full search algorithm is very greedy in computing times, but always gives the best results in compression ratio and PSNR.

The Variable Block Size Block Matching Algorithms (VBSBMA) are effective, their computational complexities may be very high. Many VLSI architectures have been proposed for the VBS-BMAs to reduce the computational complexities. However, some of the architectures [3], [7], [8] do not incorporate the capabilities of processing all the block sizes specified by the H.264. In [9], the flexibilities of the one-dimension (1D) processing element (PE) array are exploited for the realizations of H.264 VBSBMAs. Nevertheless, the employment of 1D PE array may result in high latency and low efficiency. The objective of this paper is to present an efficient VLSI architecture for hardware realizations of full-search H.264 VBSBMA. The architecture attains low latency, low power and high throughput while supporting all the block sizes 4x4,

[+] Corresponding author.
E-mail address: pmurali_nitw@yahoo.co.in

4x8, 8x4, 8x8, 16x8, 8x16, and 16x16 specified by H.264. There are sixteen modules and one variable block size motion estimation (VBSME) logic in this architecture. We use each module for the block-matching operations of a different primitive sub block. Each module is a cascade of 1D systolic array [2],

which attains high throughput and high flexibility with 100 % PE utilization. The sums of absolute differences (SAD) of all the primitive sub blocks are computed by the 16 modules in parallel. We then use the VBSME logic to concurrently identify the best-matching block to each of the 41 sub blocks from the SADs of the primitive sub blocks.

Whereas conventional 1D or 2D [5] architectures process only one motion vector (MV) within a macro block, our new architecture can process 41 MVs in less number of clock cycles. By taking the advantage of high throughput, the Architecture is allowed to reduce the clock period subject to a constraint on frame size and frame rate.

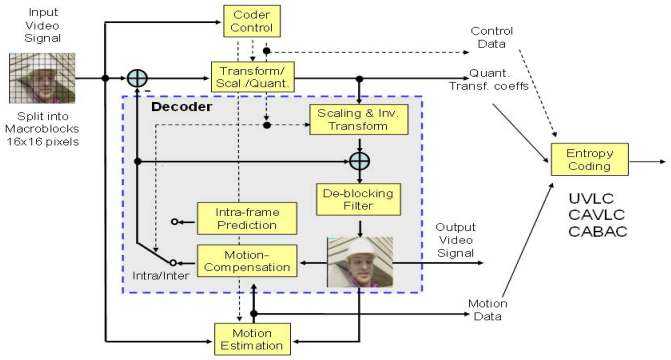

Fig 1.H.264 Encoder

2. Hardware Architecture

In this section , we first presents the system overall structure of our VBSME hardware architecture . Second we describe the processing element (PE) and adder-tree optimization method and memory organization is illustrated.

2.1 System architecture

Fig. 2 shows the proposed architecture which contains a PE 4x4 array for SAD computation and two local memories for current block and search area. The data of search areas and current blocks can be inputted into the PE array through the port from the memories for search area and current block. Since the 4x4 block is the smallest block in the MB partitions, 16 SADs of 4x4 blocks in one MB can be computed first through 16 PE 4x4 units and then the SADs of other sub-blocks and MB can be obtained by the 8X8 Mode Calculator(4x8,8x4,8x8) and 16x16 Mode calculator (16x8,8x16,16x16).

The 16 processing elements (PE) in one PE 4x4 unit can compute synchronously 16 absolute values of difference between the candidate pixels and the current pixels of one 4x4 block. The 4x4 block's SAD can be obtained by accumulating the 16 absolute values of difference in the SAD adder. This structure can calculate the SAD for one search point in one clock cycle using 16 PE Array 4x4 modules. Moreover, this architecture can make it possible to shift reference pixel data to top, bottom, right or left by connecting PE Array of top, bottom, left or right. Fig.13. shows the PE4x4 array. As shown in the figure 3.1 for every vertical change of search area data the internal data moves in either left or right direction. Similarly for horizontal change of search area data the internal data moves in either up or down shown in figure 3.2.

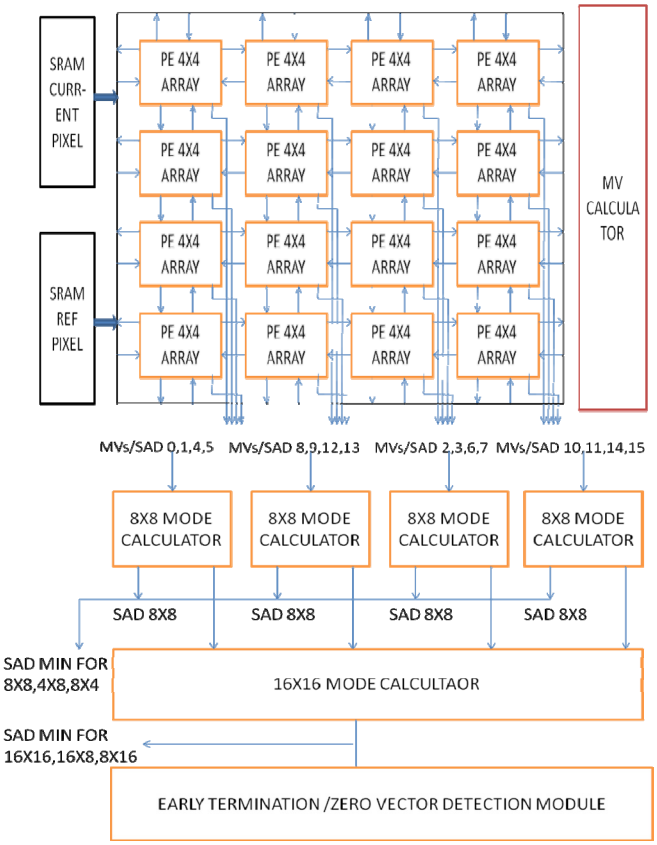

Fig 2.System architecture

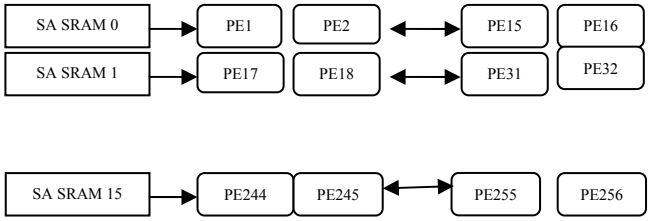

Fig 3.1 Data moment for every vertical change in search memory (both in up left right direction)

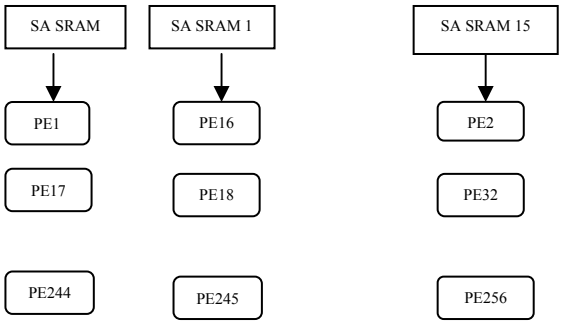

Fig 3.2 Data moment for every horizontal change in search memory (both in bottom and up direction)

It takes only one cycle to compute the absolute differences for each search position so finding best matching block for a search region of 32x32 performed in 279 clocks plus 1 clock (If we add one more adder tree delay) plus 15 clocks(Initial latency to load) .So therefore for search region of 48x48 It takes 1040 clocks (1024+1+15) to search the best matching block. In order to move the reference pixel data in all the directions we need to have 4 control signals each for one direction.

The 16 Search area SRAMS outputs only one pixel at a time. Though the placing of search 16 SRAMS occupies more space it makes the search area organization more feasible and very simple to get the required pixel from the search memory.

2.2 Circuit optimization for PE and adder Tree

The pipeline circuit and structure can be optimized to reduce the system latency and the hardware cost. In PE circuit design, another trick is applied. In theory, $|C - R|$ is equal to $|R - C|$, but the latter is preferred in this hardware Implementation. During the ME processing, the data of current pixels are constant most of the time and reference pixel data is used to change more frequently. So when performing /R – C/, the inverter which does not come in the critical path dynamic power consumption is zero. 'Row Adder Tree'[6] which include the propagated in partial SAD (PSAD I) from the previous row and absolute difference of current and reference pixels.

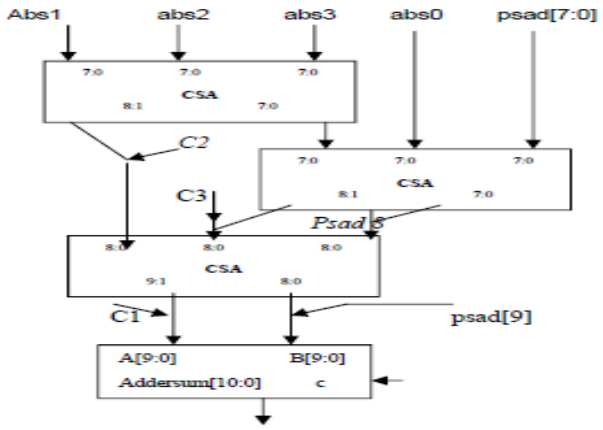

Fig.4 Optimized adder tree structure

2.3 Search Window Memory Organization:

Memory organization is another important issue in motion estimation process. We implemented a snake scan[5] to read the pixels from the current memory. Depending on the position of the offset pixels from the previous pixels the data is moved in the PE array. It is the responsibility of the controller to generate the necessary control signals needed to the movement of data in PE array.

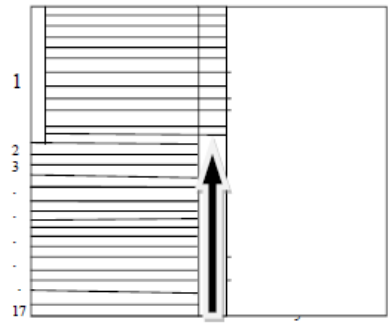

Fig. 5 : Search memory scanning order

3. Implementation

The proposed architecture is modeled using VHDL and synthesized by Synopsys Design Complier with 0.18um standard cell library. The design has a search range of 32x32 pixels and total area occupied is 179k um ² and runs at maximum clock frequency of 58.13 MHz . The design specifications achieved is shown in tTable1 and performance analysis is shown in Table 2

Algorithm	Full search
No.of PEs	16x16 Array
Search range	32x32
Block size	16x16,16x8,8x16,8x8,4x8,8x4,4x4
Technology	180nm
Max frequency	58.13MHz
Throughput	56000 MB/ Sec

Table 1: Design specifications

Critical path	17.22 ns
Critical path clk period	17.91 ns
Total area(μm²)	179K
Operating conditions	5V
Total dynamic power	180 mw

Table2: Performance analysis

Let M be the number of sub blocks in a macro block. Therefore, M=41 for the VBSBMA and M=1 for fixed block size. Latency (T) is the number of clocks required to identify the best MV in all the M subblocks. Throughput (S) of the architecture as the number of best MVs produced per clock cycle.

Table3 shows the latency, throughput and number of PEs, for previous works and proposed architecture. Compared to previous works the proposed architecture takes 305 cycles for processing of each Macro block (MB).Hence throught put achieved is 56000 MB/sec.With this throughput we can achieve SDTV 1280x720 @ 53 fps

Architecture	**Vos[7]**	**Yap[8]**	**Ours**
Number of PEs	256	16	16x16 PE Array
Latency(T)	5376	4096	305
Throughput(S)	1/256	41/4096	41/305
Block size supported	16x16, 8x8, 4x4(Masking)	All Available sizes	All Available sizes

Table3:Comparisions for Block size N=16,search size p=8

4. Conclusion

In this paper, we presented high performance hardware architecture for real time implementation of Partial propagate SAD based motion estimation for H.264 / MPEG4 part 10 video coding. The proposed architecture is implemented in VHDL. The VHDL RTL code is verified to work at 58.13 MHz in a Synopsys design compiler. The ASIC implementation can able to process 56k MBs in one clock cycle which is more than sufficient for SDTV frames at 53 fps.

5. References

[1] Y. W. Huang, T. C. Wang, B. Y. Hsieh, and L. G.Chen.Hardware architecture design for variable block size motion estimation in mpeg-4 avc/jvt/itu-t h.264. In *Proceedings ofISCAS 2003*, volume 2, pages 796–799, May 2003.

[2] 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 Circuits and Systems I*, 53(3):578–593, March 2006.

[3] Z. Y. Liu, Y. Song, T. Ikenaga, and S. Goto. A fine-grain scalable and low memory cost variable block size motion estimation architecture for h.264/avc. *IEICE Transactions on Electronics*, E89-C(12):1928–1936, December 2006.

[4] Theepan Moorthy and Andy "A scalable computing and memory architecture for variable block size motion estimation on FPGAs" IEEE 2008.

[5] Minho Kim, Ingu Hwang, "A Fast VLSI Architecture for Full- search Variable Block Size Motion Estimation in MPEG-4 AVC/H.264" .IEEE 2005

[6] Zhenyu Liu, Yiqing Huang, Yang Song, Satoshi Goto, and akeshi Ikenaga"Hardware-Efficient Propagate Partial SAD Architecture for Variable Block Size Motion Estimation in H.264/AVC

[7] L. de Vos and M.Schobinger "A VLSI architecture for flexible block matching processor", IEEE Trans. Circuits and systems for video technology vol 5,pp.417-428,,vol.51,1995.

[8] S.Y.Yap and J.V.McCanny, "A VLSI architecture for variable block size video motion estimation", IEEE Trans. Circuits and systems ,pp384-389, vol 51, 2004.