

# A Modified Hardware-Efficient H.264/AVC Motion Estimation Using Adaptive Computation Aware Algorithm

Dr.S.Rajkumar

Associate Professor/ ECE Dept, Arulmigu Meenakshi Amman College of Engineering, Vadamavandal (Near Kanchipuram)

**Abstract-** In order to increase transmission efficiency of the real world video sequences, Motion estimation (2) plays an vital role. An improved version of the reconfigurable block motion estimation algorithm (3) is proposed in this paper. The new algorithm uses a small cross-shaped search patterns to speed up the motion estimation of stationary and quasi- stationary blocks. Also we propose a pipelining method for SAD unit to minimize clock delays with minimum area overhead. Our approach increases speed and enhance the throughput for Codec design. We propose a new method “Block Motions matching technique (BMM)” (2) where compression takes place at both Spatial and Temporal domain. In BMM, images are sub divided into micro blocks of 16x16 matrices and it is checked with nearby blocks. Also this method is applied for video compression techniques. The advantage of BMM over existing system is that it compresses block level compression instead of pixel level compression that improves execution speed and adapt for fast processing.

**Index Terms-** Field-programmable gate array (FPGA), H.264/AVC, motion estimation, multipath search algorithm very large-scale integration (VLSI) architecture, video coding.

## I. INTRODUCTION

A video signal represented as a sequence of frames of pixels contains vast amount of redundant information that can be eliminated with video compression technology enhancing the total transmission and hence storage becomes more efficient. To facilitate interoperability between compression at the video producing source and decompression at the consumption end, several generations of video coding standards have been defined and adapted by the ITU-G and VCEG etc... Demand for high quality video is growing exponentially and with the advent of the new standards like H.264/AVC it has placed a significant increase in programming and computational power of the processors.

In H.264/AVC, the motion estimation (3) part holds the key in capturing the vital motion vectors for the incoming video frames and hence takes very high processing at both encoder and the decoder. Motion estimation techniques form the core of H.264/AVC video compression and video processing applications. It extracts motion information from the video sequence where the motion is typically represented using a motion vector (x, y). The motion vector indicates the displacement of a pixel or a pixel block from the current location

due to motion. This information is used in video compression to find best matching block in reference frame to calculate low energy residue to generate temporally interpolated frames. It is also used in applications such motion compensated de-interlacing, video stabilization, motion tracking etc...

Due to the great innovation of display and information technology, the stringent requirement of data capacity is drastically increased in human life. This trend makes a significant impact on storage and communication evolution. The data compression technique is extensively applied to offer acceptable solution for this scenario, some images like satellite images or medical images have very high resolution. Such high resolution images have large file size and computation time required to process such high quality images is more. Hence compression of images and video has become need of hour. The image can be compressed using lossy or lossless compression techniques. In the lossy image compression technique, the reconstructed image is not exactly same as the original image. The lossless image compression can remove redundant information and guarantee that the reconstructed image is without any loss to original image. Different image compression techniques (5) are suggested by the researchers, but the technique with high data compression with low loss is always preferred. Because of the advancement in Internet, world has come very close and can afford and avail the services such as medical, tourism, education etc., remotely. Data compression is the key in giving such fast and efficient communication. It has made large impact on service sector to provide best services to all sections of society. High code efficiency is measurement parameter for performance of data compression system.

## II. MODIFICATION IN THE EXISTING SYSTEM

In existing system an adaptive computationally scalable ME algorithm and its hardware architecture are proposed. The ME algorithm employs a two-level hierarchical search to support wider search ranges. At the fine level, the algorithm checks MVs taken from previously coded neighboring macro blocks and selects between three strategies to adapt to local motion activity. The estimation can be terminated at any point, which enables the encoder to trade the number of search points for the compression efficiency. The hardware architecture applies a novel dataflow where the interpolation of fractional positions follows the coarse-level FS and precedes the fine level ME (1). Motion estimation and compensation are performed simultaneously as all residuals forwarded to the computation of block difference measure are

buffered ready for the next coding steps. Instead of the selection of the best MV, the architecture selects the set (at least eight) of candidate MVs to forward them to the rate-distortion (RD) analysis to select the best partition mode and MV(s). This approach allows more compression-efficient coding. Additionally, the architecture (8) supports the compensation for both inter- and intra-predictions. Hence, there is no need to employ separate resources to compute residuals. The order of checked MVs is not constrained and can be adapted according to a desired search strategy.

### III. THE PROPOSED SYSTEM

In this motion estimation, search patterns have a large impact on the searching speed and quality of the performance. Based on motion vector distribution characteristics of real world video sequences, we propose a new cross-diamond search (NCDS) algorithm (6) using cross search patterns before large/small diamond search patterns in this paper. In multipath search pattern we use NCDS rather than DS. Because NCDS employs halfway technique to achieve significant speedup on sequence with (quasi-) stationary blocks. NCDS employs modified partial distortion criterion (MPDC) (4), which results in fewer search points with similar distortion. NCDS provides faster searching speed and smaller distortions (5) (7) than other popular fast block-matching algorithms.

#### 2.1. Multipath search algorithm

Multipath search (MPS) is a computationally scalable ME algorithm, which exploits spatial correlations between MVs and selects the search strategy according to estimated block motion activity and available computational resources. The initial MV is selected from the prediction set that contains MVs of left, upper left, upper, and upper right neighboring macro blocks, and the middle of the search area. The point that gives the smallest SAD is used as the starting SP. In the algorithm description, SP is considered as an MV checked for a  $16 \times 16$  luma macro block and a RP. Although MPS can be applied to wide search ranges, the cost of hardware resources would be significant. Therefore, it is better to use a hierarchical search to narrow the MPS range.

#### 2.2. Strategy Selection

The search strategy following the prediction set is selected on the basis of the estimated motion activity measured as the standard deviation (Std. Dev) of MVs of spatially neighboring MBs with respect to their median. Based on experiments, the Std. Dev threshold value is set to three to distinguish between high- and moderate/low- motion activity. For the sake of its wide range, three step search (TSS) is selected to track high- motion activity MBs. For the rest of MBs, the diamond search (DS) algorithm (1) (2) is selected. However, since the large diamond search pattern (5) used by DS is rather sparse, kite- cross diamond search (KCDS) (6) is employed when the number of

SPs remaining after the evaluation of the prediction set (*Number of SPs*) for Integer-Pel ME (IPME) is smaller than 10 (actually 26 if Fractional-Pel ME is also taken into account).

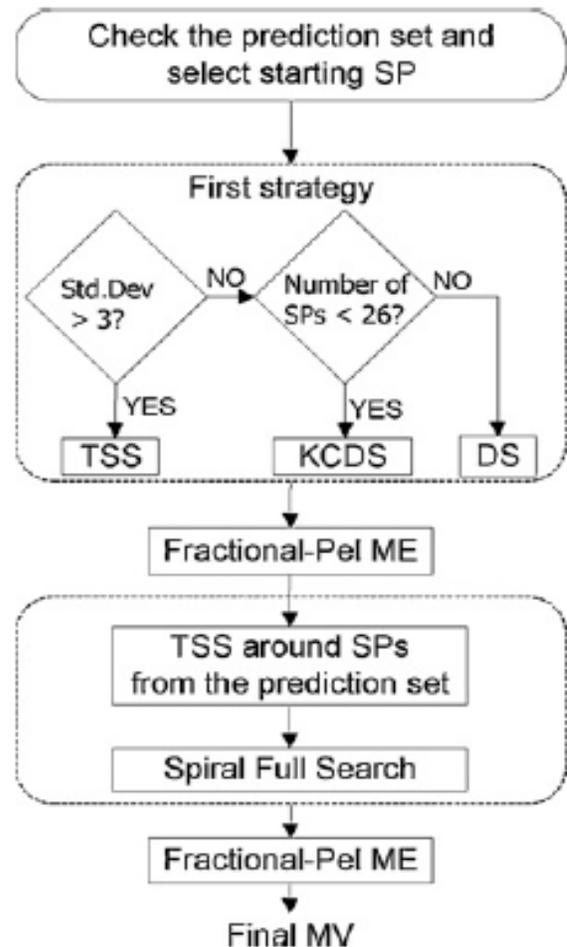
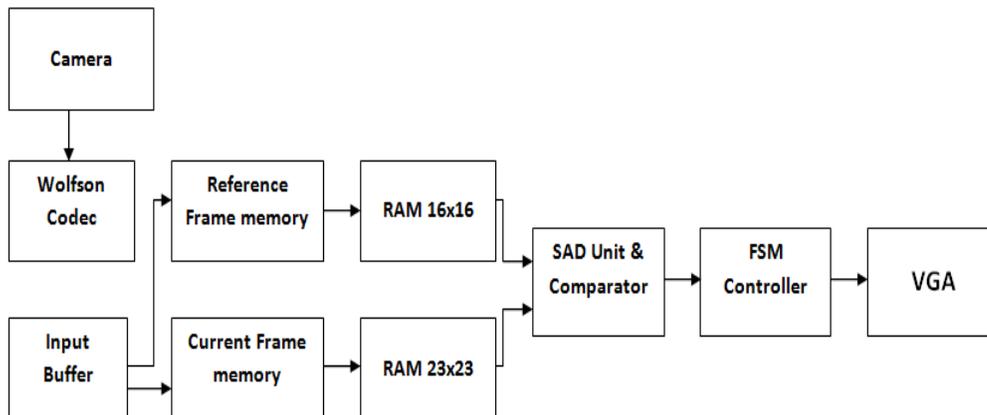


Fig 1: strategy selection

KCDS uses a denser search pattern than DS and that is better suited to track small/moderate motion when theMPS procedure embeds some basic search strategies. Integer-Pel ME distinguished by dashed boxes can be interrupted at any SP to perform Fractional-Pel ME. If TSS is selected as the first strategy, it is not repeated later for the same search center for High-motion activity of MBs. For the rest of MBs, the new cross diamond search (NCDS) (6) algorithm is selected. However, since the large diamond search pattern used by DS is rather sparse, kite- cross diamond search (KCDS) (5) (6) is employed when the number of SPs remaining after the evaluation of the prediction set (*Number of SPs*) for Integer-Pel ME (IPME) is smaller than 10 (actually 26 if Fractional-Pel ME is also taken into account). KCDS uses a denser search pattern than NCDS and, thus, its number of SPs is small.

#### IV. MODIFIED HARDWARE ARCHITECTURE



**Fig 2: Reconfigurable motion estimation.**

In improved version of the reconfigurable motion estimation algorithm we propose the new diamond cross search algorithm (6) that uses a small cross-shaped search patterns to speed up. Also we propose a pipelining method for SAD unit to minimize clock delays with minimum area overhead.

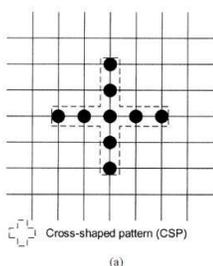
#### 4.1. CDS Algorithm

##### 4.1.1 Cross-Diamond Searching Patterns

The DS algorithm uses a large diamond-shaped pattern (LDSP) (1) and small diamond-shaped pattern (SDSP) (2), as depicted in Fig. 1. As the motion vectors distribution possesses over 96% CCB characteristics in the central 5 DCB areas, an initial CSP, as shown in Fig1, is proposed as the initial step to the DS algorithm, and is termed the CDS algorithm.

##### 4.1.2 The CDS Algorithm

CDS differs from DS by: 1) performing a CCB CSP in the first step and 2) employing a halfway-stop technique for quasi-stationary or stationary candidate blocks. Below summarizes the CDS algorithm (7).



**Step (i) Starting:** A minimum BDM is found from the nine search points of the CSP located at the center of search window. If the minimum BDM point occurs at the center of the CSP, the search stops. Otherwise, go to Step (ii).

**Step (ii) Half-diamond Searching:** Two additional search points of the central LDSP closest to the current minimum of the central CSP are checked, i.e., two

of the four candidate points located at. If the minimum BDM found in previous step located at the middle wing of the CSP, i.e., or, and the new minimum BDM found in this step still coincides with this point, the search stops. (This is called the second-step stop Otherwise; go to Step (iii).

**Step (iii) Searching:** A new LDSP (9) is formed by repositioning the minimum BDM found in previous step as the center of the LDSP. If the new minimum BDM point is still at the center of the newly formed LDSP, then go to Step (iv) (Ending); otherwise, this step is repeated again.

**Step (iv) Ending:** With the minimum BDM point in the previous step as the center, a new SDSP is formed. Identify the new minimum BDM point from the four new candidate points, 3 which is the final solution for the motion vector.

The proposed CDS algorithm is compared against five traditional BMAs: FS, 3SS, 4SS, N3SS, and DS, in four aspects. They are:

1) Average number of search points per block and its speedup ratio with follows SAD16 which performs the SAD on one respect to the FS;

$$SAD(V_i) = \sum_{x=0}^M \sum_{y=0}^N |S_i(x, y) - S_{i-1}(x + dx, y + dy)|$$

2) Average MAD per pixel;

3) Average distance from the true motion vector per block;

4) Probability of finding the true motion vector per block.

The “true” motion vectors are regarded as those found in FS. The first two aspects provide the prediction quality and searching speed improvement. The last two methods show how far from and the percentage of finding the true motion vectors, but they are independent of the first two aspects. That means that a motion vector far away from the optimal could even give better quality within the search area.

#### 4.2 Pipelining method for SAD

The most commonly used matching criterion is the sum of absolute differences (SAD), which is chosen for its simplicity and ease of hardware implementation (9). In these fast algorithms (10), only selected subsets of search positions are evaluated using SAD. As a result, these algorithms usually produce sub-optimal solutions but the computational saving over FS is significant. When it comes to hardware implementation on the other hand, the number of SAD calculations is not the only criterion for the choice of a motion estimation algorithm. Other criteria, such as algorithm regularity, suitability for pipelining and parallelism, computational complexity and number of gates which directly affect power consumption and cost of hardware, are also very important. Due to these reasons, there have been several implementations of the full search and hierarchical search which are very regular.

For an  $M \times N$  block, where  $Sl(x,y)$  is the pixel value of frame  $l$  at relative position  $x,y$  from the macro block origin and  $V_i = (dx, dy)$  is the displacement vector, SAD can be computed as row of an macro block ( $16 \times 1$ ). By iteration or parallel execution of the SAD16 operation (8), the complete SAD for the  $16 \times 16$  macro block can be performed

## V. RESULT ANALYSIS

### 5.1 Area & Power Analysis Report:

SAD value computed between current reference frames shows complete execution to identify compression vectors. There are 9 pixels are read from current memory block and 9 from reference memory block. It SAD value is executed using comparator. Based on SAD point value (7), FSM controller moves centre point pixel axis to minimum SAD point. This process is executed for the complete frame and motion vectors are loaded into vector memory. Similarly the synthesis result for proposed area and power analysis result were generated using Cyclone II EP2C35F672C6 family (10) and in the area analysis, my logic utilizes 181 logical elements which is 1% of total available LE's and other parameters are total registers 114 and total pins 425/475(89%). In power analysis, Total Thermal Power dissipation is 176.70mW out of which core dynamic thermal power dissipation is 9.41mW, core static thermal power dissipation is 80.15mW & input/output thermal power dissipation is 87.14mW.

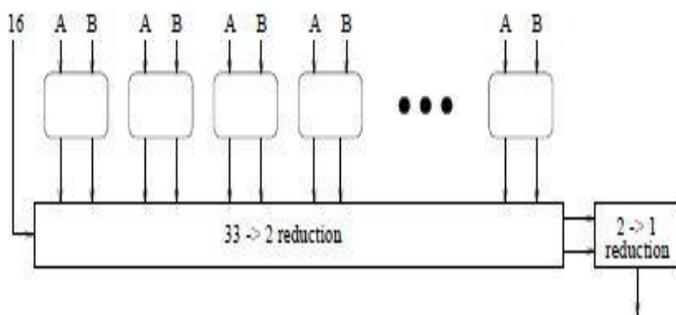
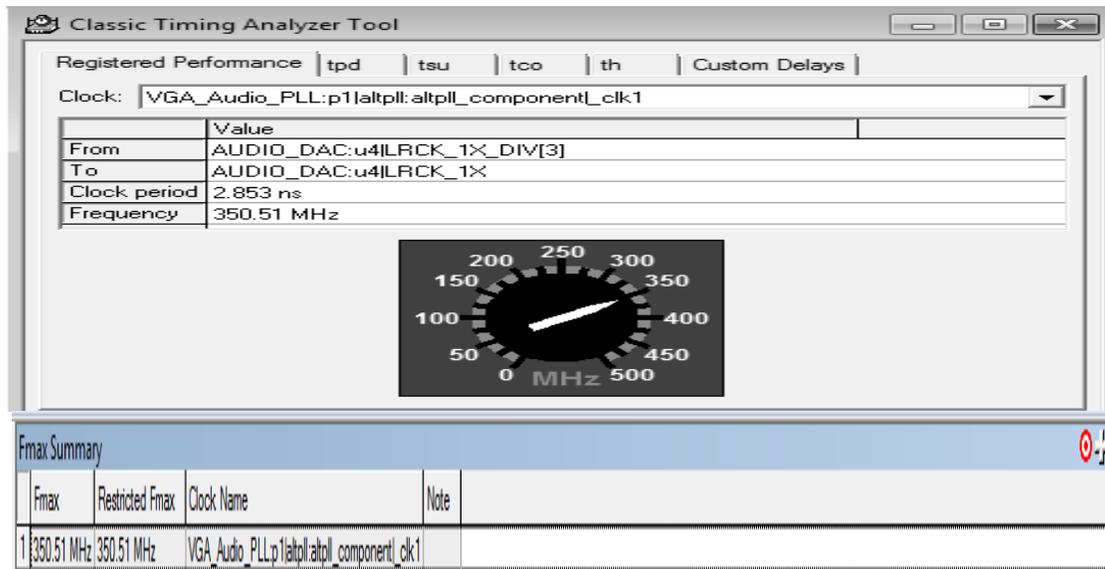


Fig 3: parallel processing of SAD values

Flow Summary	
Flow Status	Successful - Thu Apr 10 10:44:40 2014
Quartus II Version	9.0 Build 132 02/25/2009 SJ Web Edition
Revision Name	DE2_Default
Top-level Entity Name	DE2_Default
Family	Cyclone II
Device	EP2C35F672C6
Timing Models	Final
Met timing requirements	Yes
Total logic elements	181 / 33,216 (< 1%)
Total combinational functions	181 / 33,216 (< 1%)
Dedicated logic registers	114 / 33,216 (< 1%)
Total registers	114
Total pins	425 / 475 (89%)
Total virtual pins	0
Total memory bits	0 / 483,840 (0%)
Embedded Multiplier 9-bit elements	0 / 70 (0%)
Total PLLs	1 / 4 (25%)

PowerPlay Power Analyzer Summary	
PowerPlay Power Analyzer Status	Successful - Wed Apr 16 20:11:26 2014
Quartus II Version	9.0 Build 132 02/25/2009 SJ Web Edition
Revision Name	DE2_Default
Top-level Entity Name	DE2_Default
Family	Cyclone II
Device	EP2C35F672C6
Power Models	Final
Total Thermal Power Dissipation	176.70 mW
Core Dynamic Thermal Power Dissipation	9.41 mW
Core Static Thermal Power Dissipation	80.15 mW
I/O Thermal Power Dissipation	87.14 mW
Power Estimation Confidence	Low: user provided insufficient toggle rate data

### 5.2 Timing analysis report:



The Timing analysis report shows, **Clock period : 2.853 ns & Frequency : 350.51 MHz**

The timing analysis of the method minimizes clock delays with minimum area overhead. My approach increases the design speed and enhances the throughput for Codec design

## VI. CONCLUSION

The proposed architecture implements combination of the hierarchical search and the MPS adaptive search algorithm, allowing more compression-efficient coding. The design needs the lowest gate count among the referred designs. The memory consumption is relatively high, especially when the number of RPs increases. In this project to speed up the search, a new cross diamond search algorithm (6) based different Evolutionary Computing Techniques is proposed. Traditional fast block matching algorithms are easily trapped into the local minima resulting in degradation on video quality to some extent after decoding. Since these Evolutionary Computing Techniques are suitable for achieving global optimal solution. Because NCDS employs half way technique to achieve significant speedup on sequence with (quasi-) stationary blocks. NCDS employs modified partial distortion criterion (MPDC), which results in fewer search points with similar distortion. NCDS (2) provides faster searching speed and smaller distortions than other popular fast block- matching algorithms and also we use pipelining method for SAD unit to minimize clock delays with mi

## REFERENCES

- [1] Shan Zhu and Kai-Kuang Ma, "A New Diamond Search Algorithm for Fast Block-Matching Motion Estimation, IEEE Transactions on Image processing, vol. 9, no. 2, February 2000".

## AUTHORS

- [2] Chun-Ho Cheung and Lai-Man Po "A Novel Cross- Diamond Search Algorithm for Fast Block Motion Estimation IEEE transactions on circuits and systems for video technology, vol. 12, no. 12, December 2002".
  - [3] Ching-Yeh Chen, Yu-Wen Huang, Chia-Lin Lee, and Liang-Gee Chen, "One-Pass Computation-Aware Motion Estimation With Adaptive Search Strategy IEEE transactions on multimedia, vol. 8, no. 4, august 2006"
  - [4] Haibing Yin, Huizhu Jia, Honggang Qi, Xianghu Ji, Xiaodong Xie, and Wen Gao, "A Hardware-Efficient Multi-Resolution Block Matching Algorithm and Its VLSI Architecture for High Definition MPEG-Like video Encoders IEEE TRANSACTIONS on circuits and systems for video technology, vol. 20, no. 9, September 2010".
  - [5] Stephan Wong, Stamatis Vassiliadis, and Sorin Cotofana "SAD implementation in FPGA hardware", Zhu and K. K. Ma, "A new diamond search algorithm for fast block matching motion estimation," in Proc. Int. Conf. ICICS, 1997, pp. 292–296.
  - [6] C. W. Lam, L. M. Po, and C. H. Cheung, "A novel kite-cross-diamond search algorithm for fast block matching motion estimation," in Proc. IEEE ISCAS, vol. 3, May 2004, pp. 729–732.
  - [7] L. K. Liu and E. Feig, "A block-based gradient descent search algorithm for block motion estimation in video coding," IEEE Trans. Circuits Syst., Video Technol., vol. 6, no. 4, pp. 419–422, Aug. 1996.
  - [8] M. Jakubowski and G. Pastuszak, "An adaptive computation-aware algorithm for multi-frame variable block-size motion estimation in H.264/AVC," in Proc. Int. Conf. SIGMAP, 2009, pp. 122–125.
  - [9] S. Warrington, S. Sudharsanan, and W.-Y. Chan, "Architecture for multiple reference frame variable block size motion estimation," in Proc. IEEE ISCAS, May 2007, pp. 2894–2897.
  - [10] L. Zhang and W. Gao, "Reusable architecture and complexity controllable algorithm for the integer/fractional motion estimation of H.264," IEEE Trans. Consumer Electron., vol. 53, no. 2, pp. 749–756, May 2007.
- First Author – Dr.S.Rajkumar, Associate Professor/ ECE Dept, Arulmigu Meenakshi Amman College of Engineering, Vadamavandal (Near Kanchipuram), rajakumar.elango@gmail.com**