Indian Journal of Science and Technology, *Vol 9(32), DOI: 10.17485/ijst/2016/v9i32/89148, August 2016*

ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Pipelined Architecture for Motion Estimation in HEVC Video Coding

P. Jayakrishnan* and Harish. M. Kittur

School of Electronics Engineering, VIT University, Vellore - 632014, Tamil Nadu, India;
pajayak@gmail.com, kittur@vit.ac.in

Abstract

Background: High Efficiency Video Coding (HEVC) is the latest video coding standard for video coding. Motion estimation is one of important stages in video coding. In HEVC, the conventional macroblocks are replaced by coding tree units (CTU). **Method/Statistical Analysis:** The maximum block size is increased from 16×16 to 64×64 in HEVC makes motion estimation more complex. An efficient hardware architecture ha to be designed for real time implementation of motion estimation. In motion estimation, sum of absolute difference (SAD) is calculated between current block and reference blocks present in two different video frames and minimum of SAD gives motion vector of the block. **Findings:** This paper presents design of pipelined SAD architecture for efficient SAD calculations. The proposed design is used in diamond search algorithm to carry out the motion estimation for evaluating the performance of the design. The proposed architecture employs pipelining at 8×8 block. The proposed design is implemented in TSMC 90 nm technology and operates at maximum frequency of 486 MHz. **Applications/Improvements:** The design achieves 78% reduction in power compared to the previous design and around 50 % increase in operating frequency.

Keywords: H.264/AVC, H.265/HEVC, Motion Estimation, Motion Vector, Sum of Absolute Difference

1. Introduction

HEVC (High Efficiency Video Coding) developed under joint collaboration of ITU-T VCEG and ISO/IEC MPEG, together under the name JCT-VC (Joint Collaborative Team on Video Coding) is the latest video coding standard. It is targeted to double the video compression efficiency to that of its predecessor H.264/AVC. It can achieve 50% bit rate saving compared to H.264/MPEG-4 AVC for the same video quality[1]. Encoding a frame in HEVC is performed in coding units (CU) and transform units (TU). A frame is divided into CU, with each CU size varying from 8×8 to 64×64 pels. The maximum size of CU is 64×64 in HEVC which adds complexity in design of ME unit. Motion estimation (ME) is one of vital stage many video coding standards and it is more time consuming part in video coding. ME need an additional attention in design to reduce the encoding time. ME is the process of matching the current block with the multiple blocks present in reference frame and find the best match with least error and corresponding motion vector (MV). As shown in the figure.1, in each level of CU, HEVC supports 8 different ways of partitioning the inter prediction unit. At depth 0, 2N=64 and depth 1, 2N = 32 and so on. Partitions (a) to (d) are called symmetric modes and (e) to (h) are called asymmetric partition modes (AMP)[2].

Sum of Absolute Difference (SAD) is used as cost function measure in motion estimation. The calculation of SAD from the current and reference block is performed using (1)

$$SAD = \sum_{i=1}^{M} \sum_{j=1}^{N} | CB(\mathrm{i}, \mathrm{j}) - RB(i+x, j+y) | \quad (1)$$

Where, CB represents the current block pixels and RB represents the reference block pixels. The design of SAD architecture is vital in ME design. HEVC supports larger CU, which increases the complexity of SAD and needs to have efficient architecture which higher degree of parallelism.

We can find many design of SAD units with different levels of parallelism. P. Nallurietal, presented a 1-stage parallel SAD architecture[3] and SAD of four 4×4 blocks

*Author for correspondence

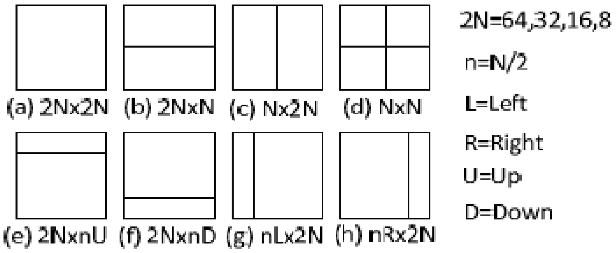

Figure.1. Inter prediction unit partition sizes in HEVC.

processed in parallel. The design are proposed which is specific to ASIC as well as FPGA. P. Nallurietal, proposed a 2-stage parallel SAD architecture[4] and SAD of four 8×8 blocks processed in parallel. In both the implementations, SAD of larger blocks is computed using the SAD of smaller block size in non-parallel approach and as the number of parallel stages is increased; there is an increase in input data lines. The parallel architecture[5] by Zhenyu Liu et al., calculates 64×64 block from SADs of 4×4 block. We propose a six stage pipelined SAD architecture block size 64×64. The SAD blocks uses SAD values of smaller block sizes to calculate larger block SAD. The computation time of absolute difference, we propose pipelining at 8×8 block and reduces the overall computation time required for SAD. In this paper Diamond Search algorithm is implemented using pipelined SAD architecture.

2. Pipelined SAD Architecture

The proposed ME design computes Motion Vectors (MV) from SAD between the reference and current blocks. The design performs the computation basic block size of 8×8 and larger block sizes are computed from basic blocks. In HEVC, the block size varies from 64×64 pixels to 8×8.The proposed SAD unit uses pipelining to provide high speed operation.

Six stage pipelined SAD architecture is represented in Figure. 2. At each clock cycle, the SAD unit is fed with pixel values of 8×8 block from memory buffer. The first stage of SAD unit performs absolute difference of each block and make use 64 Absolute Difference (AD) unit. Consecutive five stages are made of 4:2 CSA trees that results in SAD of 8×8 block. The pipelining has six stages and it takes six clock cycles to compute the SAD of 8×8 block.

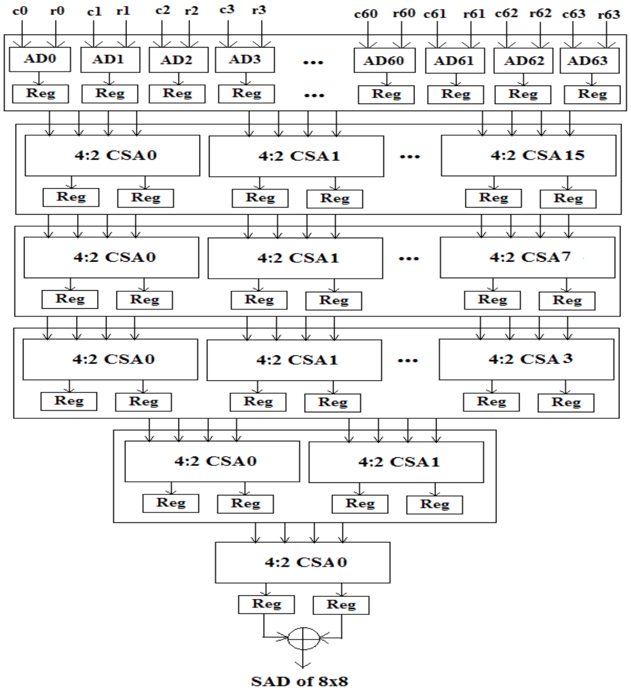

Figure 2. Pipelined SAD unit.

2.1 Absolute Difference Circuit

There are many ways to compute the absolute difference. The fastest way is to use two's complement as shown in eqn (2),

$$\left|X-Y\right|=\begin{cases}X+Y'+1, if \;\; MSB=1\\ X'+Y+1, if \;\; MSB=0\end{cases} \tag{2}$$

Where, MSB is the most Significant bit, X' and Y' are the complements of X and Y, respectively. In a two's complement addition, the MSB is 1 only if X is greater than Y. Hence the AD circuit for one pixel can be designed using the circuit shown in Fig. 3

2.2 Adder Circuit

The consecutive pipeline stages will perform the addition of these absolute differences to obtain the SAD of 8×8 block. The adder circuit used here is 4:2 Carry Save Adder (CSA). This adder adds 4 inputs simultaneously and gives two output i.e. sum and carry. After adding sum and carry the final sum is calculated.

The architecture of 4:2 CSA is shown in figure. 4. It uses slices of 3:2 CSA as main building blocks, which is highlighted in figure. 4. A slice consists of two 3:2 CSA blocks.

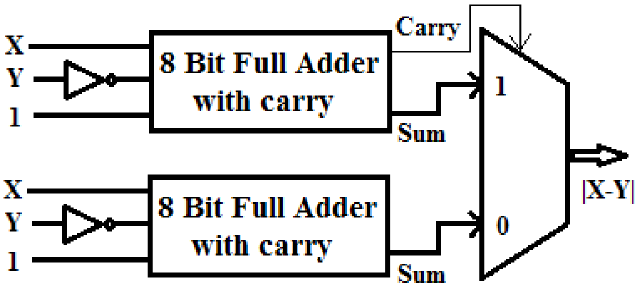

Figure.3. AbsoluteDifference Circuit.

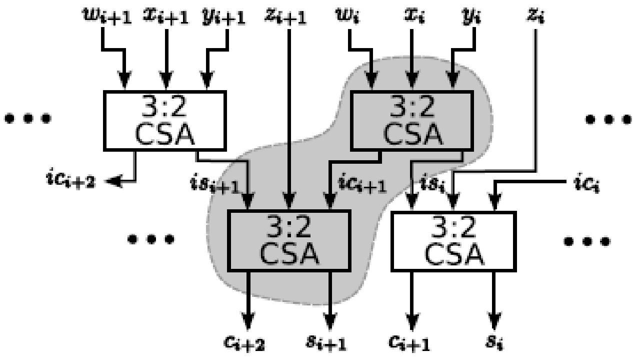

Figure.4. 4:2 compressor implementation using 3:2 CSA.

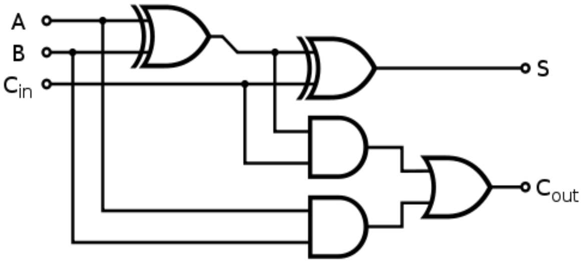

Figure 5. 3:2 CSA.

Block diagram of 3:2 CSA is shown in figure. 5, which is nothing but simple full adder circuit. A simple full adder takes 3 bits as inputs and results in 2 bit output as sum and carry. A slice in figure. 4 makes use of two 3:2 CSA unit, where the total inputs are 4. In figure 4 consider the inputs wi, xi, yi, zi+1 in the highlighted slice, which produces the sum si+1 and Ci+2. The 4:2 CSA makes use of 8 slices.

3. Diamond Search Algorithm

To perform motion estimation, usually exhaustive search or also referred as full search is used which provides better quality than any other searching algorithm. But full search algorithm make exhaustive search in reference frame which is time consuming one, there are several fast search algorithm is used with compromise in quality. We have chosen diamond search algorithm[6] to evaluate the performance of pipelined architecture as it is a simple and iterative algorithm and will be very good test model for a pipelined architecture.

3.1 Large Diamond Search (LDS) Algorithm

LDS finds total nine SADs for nine different reference blocks as shown in figure. 6 where dark circle indicates the position of reference block i.e. {(0,0), (1,1), (-1,1), (-1,-1), (1,-1), (2,0), (0,2), (-2,0), (0,-2)}. These nine SADs are compared together to give minimum SAD. The best matching block whose SAD with current block is taken as center block (0,0) and again large diamond search is repeated until the best matching comes to be center block.

3.2 Small Diamond Search (SDS) Algorithm

On completion of LDSP, with the output of LDSP, similar search is performed with five reference blocks which are {(0,0), (0,1), (-1,0), (0,-1), (1,0)} as shown in figure 7. Similar to LDS the search in small diamond search is also iterative until the center block becomes the best match.

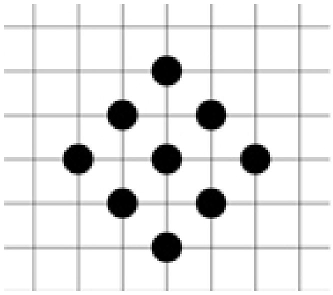

Figure 6. Large Diamond Search.

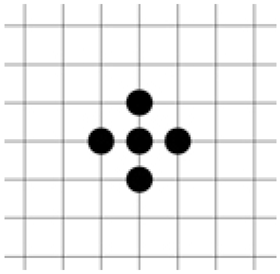

Figure 7. Small Diamond Search.

For SAD unit calculates the SAD of given block depending upon control unit output. Control unit will decided that how many cycles are needed to calculate the corresponding SAD. Figure 2 indicates the pipelined architecture of SAD which gives 8x8 SAD at output.Using this architecture remaining SADs are calculated. For example 16x16 block consists of four 8x8 blocks. So to calculate SAD of 16x16, we need to add SAD of 8x8 block four times. So SAD unit will give SAD for 8x8, 16x16, 32x32 and 64x64.

3.3 Control Unit

To carry out the diamond search algorithm a control unit is designed, to perform the iteration in the algorithm. Control unit consists of three inputs i.e. Depth, current block, and reference block. Depth decides the size of the block. Depth 0 means block size is 64×64, depth 1 means block size is 32×32, depth 2 means block size is 16×16 and depth 3 means block size is 8×8. Current block and reference block consist of pixels for 8x8 block. The block diagram of diamond search algorithm implemented using pipelined architecture is given in Figure.8.

Here SAD of larger block sizes are calculated with a use accumulator which receives the instruction from control unit based on depth of the block. Comparator which compares all the SADs and compute the minimum SAD which is used for choosing the best matching block. During the iteration comparator gives the minimum SAD value to the control unit, such that the control unit choose the centre block for the next iteration

4. Synthesis Results

The proposed architectures was designed in Verilog HDL and synthesized in cadence RTL compiler using TSMC 90nm technology library. The synthesis results are shown in table 1 below. The results shows that proposed architecture provides high speed operation compared with previous SAD architectures with maximum operating frequency of 486MHz.

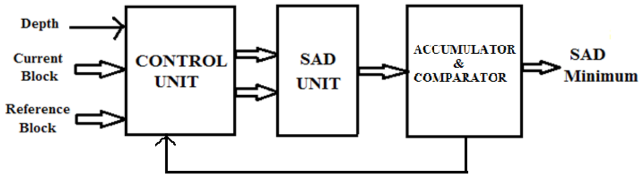

Figure 8. Block diagram of motion estimation.

Table 1. Synthesis results of the proposed architecture

Process	TSMC 90nm
Frequency	486 MHz
Power	51.605 mW
Area	45167 um^2
Gate Count	6551

Table 2. Comparison of proposed parallel search architecture with previous works

	[3]	[5]	**Proposed**
Process	FPGA Xilinx Virtex 5	TSMC 180nm	TSMC 90nm
Max. Frequency (MHz)	171.9	204.8	486
Power (mW)	136.18	241.43	51.605

Table 3. Clock cycles and PSNR of different video sequences using diamond search algorithm

Video Sequence And Format	**Diamond Search using Pipelined SAD architecture**	
	Clock cycles / frame	**PSNR**
Caltrain_CIF (352×288)	252208	32.1040
Foreman_CIF (352×288)	251680	33.1802
flower_CIF (352×288)	251936	30.8813
Beauty_HD (1080×1920)	3605752	31.2993

A comparison between the proposed design and previous SAD designs are presented in Table 2. From the table it can be observed that, the proposed design achieves 78% less power and around 50% higher operating frequency.

The maximum operating frequency is 486 MHz, compared to the frequency obtained in[3,5] the proposed design achieved high clock rate, this is because the design has six pipelined stages and it will take 8 clock cycles to compute SAD of one 8×8 block. But if processed in large number, this design proved to be efficient; table 3 gives average number of clock cycles required to process one frame

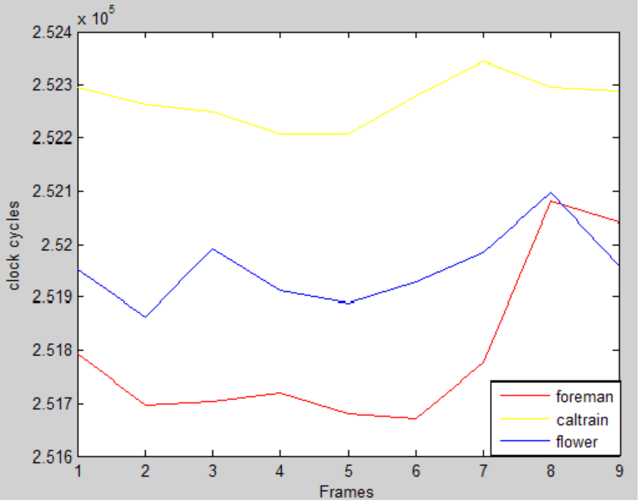

Figure 9. Frame wise clock cycles for CIF video sequences.

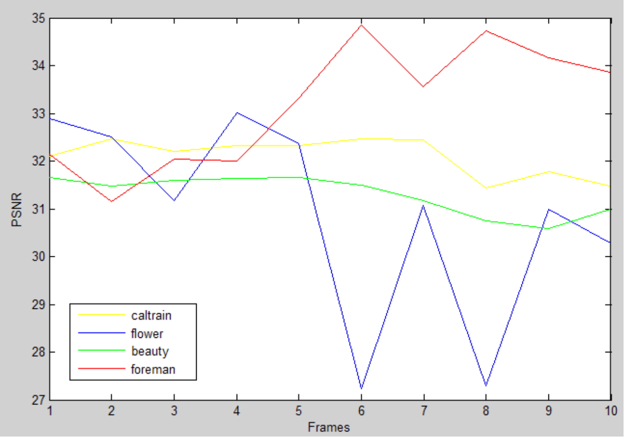

Figure 10. PSNR for video sequences using diamond search algorithm.

for the set of video sequences. Figure 9 shows the plot of number of clock cycles taken to code the different video sequences. Figure 10 shows the plot of PSNR values of different video sequences coded using the proposed design. sequences.Table 3 and figure.10 gives the PSNR achieved for different video sequences using diamond.

5. Conclusion

The paper presents efficient and high speed SAD calculation architecture for real time applications. Due to use of pipelining, this architecture gives good performance as compared with other architectures. The proposed architecture calculates 8×8 SAD first and reuses it to calculate remaining SADs. The design is synthesized in cadence RTL compiler using TSMC 90nm technology library. The result shows that maximum operating frequency is 486 MHz. This SAD unit is a basic processing unit which can be utilized implementation of any motion estimation algorithm.

6. References

1. Sullivan GJ, Ohm J, Han W-J, Wiegand T. Overview of the High Efficiency Video Coding (HEVC) Standard. IEEE Trans Circuit and System for Video Tech. 2012 Dec; 22(12).
2. Kim I-K et al. Coding efficiency improvement of HEVC using asymmetric motion partitioning. IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB). 2012.
3. Nalluri P, Alves LN, Navarro A. A novel SAD architecture for variable block size motion estimation in HEVC video coding. IEEE International Symposium on System on Chip (SoC). 2013 Oct.
4. Nalluri P, Alves LN, Navarro A. High speed sad architectures for variable block size motion Estimation in HEVC video coding. IEEE. 2014.
5. Liu Z, Goto S, Ikenaga T. Optimization of Propagate Partial SAD and SAD tree motion estimation hardwired engine for H.264. IEEE Inter Conf on Compo Des. 2008 Oct. p. 328–33.
6. Zhu S, Ma K-K. A New Diamond Search Algorithm for Fast Block-Matching Motion Estimation. IEEE Transactions on Image Processing. 2000 Feb; 9(2).