Implementation of a Low Latency Motion Estimator for HEVC Encoder on FPGA

Estefania Alcocer, Otoniel Lopez-Granado, Roberto Gutierrez and Manuel P. Malumbres

***Abstract*— HEVC is the latest video coding standard aimed to compress double to that its predecessor standard H.264. Motion Estimation is one of the critical parts in the encoder due to the introduction of asymmetric motion partitioning and higher size of coding tree unit. In this paper, a design for an Integer Motion Estimator of HEVC is presented over specific hardware architecture for real time implementation. The implementation shows a new IME unit supporting asymmetric partitioning mode which significantly reduce the overall motion estimation processing time. The prototyped architecture has been designed in VHDL, synthesized and implemented using the Xilinx FPGA, Zynq-7000 xc7z020 clg484-1. The proposed design is able to process 30 fps at Full-HD and 15 fps at 2K resolution.**

***Keywords*— HEVC, video coding, FPGA, motion estimation.**

I. Introduction

The new High Efficiency Video Coding (HEVC) standard was introduced targeting to double the compression efficiency. It can achieve 50% bit rate saving compared to H.264/MPEG-4 for the same video quality, but Motion Estimation (ME) and Motion Compensation (MC) are the major loads at video encoder. They consume more than 90% of encoding time. The overhead involved is larger compared to H.264 which consequently increases the complexity of HEVC encoder. This is due to several issues such as a large set of Coding Tree Unit (CTU) partitioning modes, and the varying size of Coding Units (CU) in relation to the previous standard H.264 [1].

The coding structure in HEVC consists of CUs with a maximum block size of 64x64 pixels, and each CU can be divided recursively until a block size of 8x8 pixels. The CUs consist of prediction units (Intra or Inter modes) and the size of each prediction unit may vary from the CU maximum size to 4x4 pixels in Intra mode or, 4x8 and 8x4 for Inter prediction, supporting 8 modes of partitioning for each CU depth level as shown in Fig. 1, where 2N = 64 represents depth 0, 2N = 32 depth 1 and so on. Partitions from (a) to (d) are called Symmetric Motion Partitions (SMP), whereas partitions from (e) to (h) as Asymmetric Motion Partitions (AMP). The ME unit is responsible for comparing all prediction units of the current frame with the positions within the search window belonging to the reference frames (past or future) so that the difference between these blocks contains the least residual information [2].

Estefania Alcocer, Otoniel Lopez-Granado, Roberto Gutierrez, Manuel P. Malumbres
Miguel Hernandez University, Elche
Spain
{ealcocer, otoniel, roberto.gutierrez, mels}@umh.es

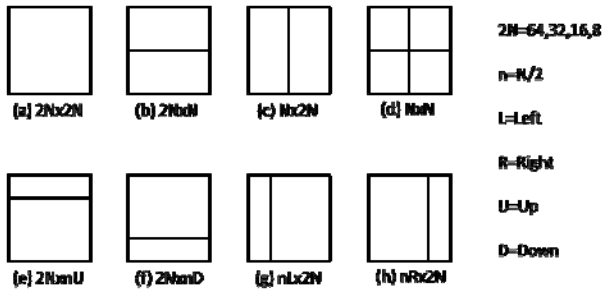


Figure 1. Partitioning sizes for Inter Prediction in HEVC

As mentioned above, ME is the task which has a greater computational cost within HEVC video encoding. Therefore, in this paper a VLSI architecture is proposed for the calculation of ME in a more quickly and accurately way, in order to significantly reduce the computational cost of the entire encoder.

This paper is structured as follows: Section II gives an overview of the proposed design, explaining each of the modules that comprise the hardware SAD unit. In Section III, results in terms of latency, FPGA resource usage, and frame rates of the proposed hardware architecture are shown. Finally, in section IV some conclusions are drawn.

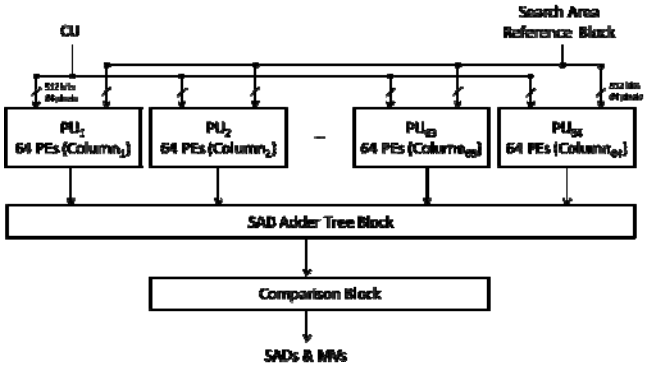
II. Proposed Architecture

The largest CTU size in the proposed architecture is 32x32 by default, instead of 64x64, which reduce the complexity of the ME. This approach strikes a good balance between the hardware resources used and compression quality obtained [3]. In addition, the Full Search (FS) strategy has been chosen for being the most accurate, since thus every current CU is evaluated at every single point of the search area in reference frame [4]. This has the advantage of a computational regularity and excellent video quality output.

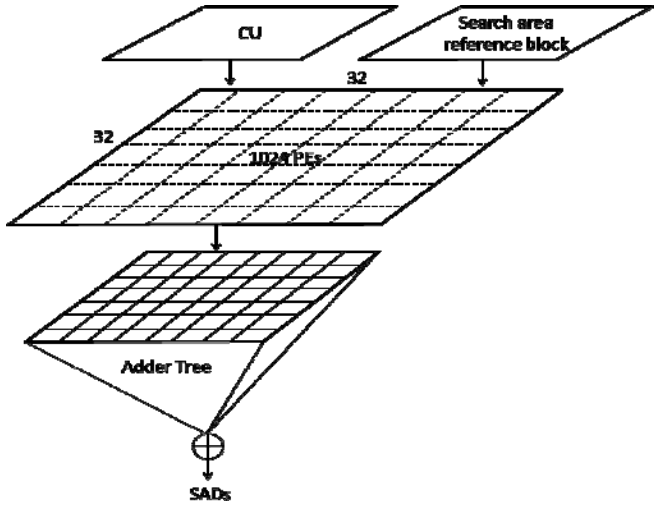
The search area is the region of the reference frame where a full search of the current CU is performed, evaluating the distortion in each pixel position in this area [5]. In this architecture, the size of the search area is defined by the HEVC standard, ie, twice the CU size, in this case being 64x64 pixels.

Our system consists of (a) search area memories for reference frame pixels which are moved from Block-RAMs (BRAMs) to propagation registers for data reuse, (b) memory for current CU pixels, (c) 32 Processing Units (PU), (d) 1024 Processing Elements (PE), (e) a Sum of Absolute Difference Summation (SAD) Tree Block and (f) a comparator block. Each PE computes the distortion of both current and reference pixel. One PU consists of 32 PEs, which calculates the distortion values of a column from a 32x32 block. In each clock cycle, current and reference pixels are inputted to 32 PUs. Therefore, the distortion

values of a 32x32 block are calculated in a clock cycle. The SAD Tree Block (STB) calculates de SAD values of variously size modes using the results derived from PUs of a block of 32x32. Fig. 2 shows the general structure of the architecture proposed, where Fig. 2a shows the hardware blocks of the system and Fig. 2b represents the concept of the implemented SAD unit.



(a) Hardware blocks



(b) SAD unit concept

Figure 2. General structure of architecture proposed

In order to provide a high data reuse, a snake scan order and a reconfigurable data path with 32 propagation registers are adopted. At first step, 'A', a shift register fetches a row of 32 pixels from the BRAM at each clock cycle, setting the propagation of shift registers as downward. After loading 32 rows in the registers (a 32x32 CU), a 32-pixel column is delivered to each PU from both shift registers and current CTU memory bank, to compute the corresponding first SAD. After that, a new 32-pixel row is loaded in the shift registers discarding the first row, so we can proceed to compute the new CU just displaced one pixel in downward direction. From this moment, at each clock cycle we obtain the SAD corresponding to a CU at each search area position. When computing the CU at last position in the downward scanning direction which corresponds to the last 32 rows of the search area, the next position to be evaluated will be the one resulting from shifting one pixel/column to the right, step 'B'. Just after computing this CU location, the memory controller will proceed to compute the CU shifted in the upward direction, step 'C', following the same criteria as in step A. This procedure will iterate until all searching CU positions in the search area have been processed, getting all the SADs for each CU at every clock cycle. Fig. 3 shows the scan order process of the search area described previously.

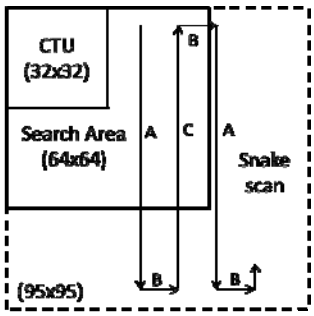


Figure 3. Scan order of reference search area

The STB block obtains the CTU SAD values for each CTU partition from 32x32 (maximum CTU size in our proposal) to 4x8/8x4 as determined by HEVC standard, including both mode partitions AMP and SMP. After receiving from PUs the 32x32 SADs associated to the current search area position, a succession of aggregation stages are performed to compute the corresponding SAD values for all the CTU partitions. In this case, the total number of SADs corresponding to the CUs and its seven partitions within the CTU is 165, where 41 SADs correspond to square partitioning modes (2Nx2N, NxN), 84 SADs as symmetric partitioning modes (2NxN, Nx2N), and 40 SADs as asymmetric partitioning modes (2NxnU, 2NxnD, nLx2N, nRx2N). Table I shows a summary about the number of SADs calculated for each candidate CU in a fixed position of the search area.

TABLE I. TOTAL NUMBER OF SADs FOR EACH 32X32 CTU PARTITION

Partition mode	SADs No.	Partition mode	SADs No.
32x32(2Nx2N)	1	16x16(Nx2N)	8
32x32(2NxN)	2	16x16(NxN)	16
32x32(Nx2N)	2	16x16(nLx2N)	8
32x32(NxN)	4	16x16(nRx2N)	8
32x32(nLx2N)	2	16x16(2NxnU)	8
32x32(nRx2N)	2	16x16(2NxnD)	8
32x32(2NxnU)	2	8x8(2Nx2N)	16
32x32(2NxnD)	2	8x8(2NxN)	32
16x16(2Nx2N)	4	8x8(Nx2N)	32
16x16(2NxN)	8	**TOTAL**	**165**

At each stage, all pairs of consecutive columns/rows are added, reducing to half the width/height of the resulting partition, as shown in Fig. 4. This SAD aggregation process is followed until the last partition size is reached (1x1), ie. the SAD corresponding to the 32x32 partition. At intermediate stages, the SADs of the rest of partitions are stored.

Finally, the comparator block should keep the minimum SAD values for each CTU partition with their corresponding motion vectors (search area locations). So, it will compare all incoming SADs from the adder tree block with the minimum SADs previously found. After comparing the SADs from the last search area location, the minimum SADs

for each partition and the associated Motion Vectors (MVs) are obtained.

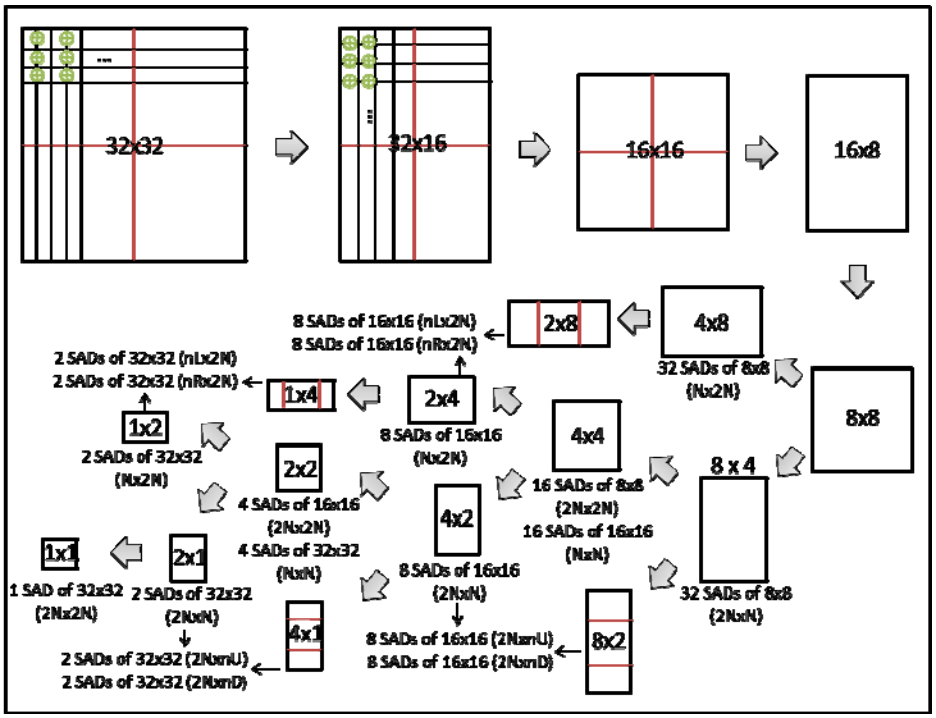


Figure 4. Structure of SAD Tree Block

The proposed system Works in a pipeline mode. The memory read and shift registers use only one clock cycle, once a preload in the shift registers of the 31 first rows of reference pixels corresponding to the first CU of 32x32 pixels has been performed. Another one clock cycle more is needed for the calculation within PUs. The STB requires ten additional clock cycles and the comparison block needs one additional clock cycle. Finally, since the above procedures are performed for each position within the reference search area (64x64 pixels), the proposed architecture needs 4140 clock cycles for processing the Integer Motion Estimation (IME) of all CUs from 32x32 to 8x8 and its partitions.

III. Results

Our proposal has been modelled in VHDL and it has been synthesised, simulated and implemented on the Xilinx FPGA, Zynq-7000 XC7Z020-1CLG484C [7]. The Xilinx tools, ISE Design Suite 14.7, ISIM, and Vivado 2013.4 have been employed to get FPGA results about maximum frequency, latency design and area used.

The aim of this work has been to achieve the lowest possible latency when being obtained the minimum SADs and its associated MVs. In that way, it is made the most of the potential of a reconfigurable device, making a maximum use of available resources in the FPGA.

Therefore, as discussed in Section II, after the necessary clock cycles corresponding to the preload of search area in shift registers, we get 165 SADs corresponding to all CUs and its 7 partitions in a single clock cycle, being the CTU situated into a particular pixel position of the search area. In the next cycle, other 165 SADs corresponding to the next position in the search area are obtained. In this point, a comparison between these last SADs and the previous ones is done in order to keep the minimum SADs and provide its associated MVs.

This procedure is repeated for all positions within the search area. Since in this architecture the CTU size is defined as 32x32 pixels and search range is twice the size of the CTU, in order to go over all the pixels in the search window (64x64), 4096 clock cycles are needed. Therefore, a minimal latency of 4096 cycles is achieved, being able to perform in a single clock cycle the computation of all SADs when the CTU is located in a fixed pixel position of the search area.

The results of the implemented architecture on FPGA show an efficient use of the available area. 40,455 F.F. (Flip Flop) of the 106,400 available and 49,450 of the 53,200 LUTs (Look-up-Table) available are used. In addition, the operating frequency is 140 MHz. Therefore, considering this maximum frequency at which the FPGA can work, the clock cycles needed for the integer motion estimation of a CTU, the proposed architecture is capable of processing 32 fps at Full-HD video resolution and 16 fps at 2K video format.

IV. Conclusion

In this paper a hardware architecture for parallel computing of SADs in HEVC motion estimator has been presented. The design has been prototyping on the Xilinx FPGA, ZC702 (XC7Z020-1CLG484CES).

This architecture is highly efficient in terms of parallelism and computational complexity. The proposed system is able to calculate 165 SAD values for block sizes from 32x32 to 4x8 or 8x4 pixels supporting all partitioning modes, including asymmetric ones, with a low latency of only 4140 clock cycles. The proposed design uses 42.7% of F.F. and 92.9% of LUTs.

With a maximum frequency of 140 MHz, the hardware process is capable of encoding 32 frames per second at Full-HD video formats and 16 fps at 2K, which represents a huge complexity reduction of the HEVC video encoding process, achieving real-time encoding for high definition video contents and beyond.

References

[1] Ahmed Medhat, Ahmed Shalaby, Mohammed S.Sayed and Maha Elsabrouty, "A highly parallel sad architecture for motion estimation in HEVC encoder", in Consumer Electronics (ICCE) 2014 IEEE International Conference on, January 2014, pp. 187-188.

[2] Purnachand Nalluri, Luis Nero Alves and Antonio Navarro," High speed sad architectures for variable block size motion estimation in HEVC video coding", in Image Processing (ICIP) 2014 IEEE International Conference on, October 2014, p. 1233-1237.

[3] Xu Yuan, Liu Jinsong, Gong Liwei, Zhang Zhi and Robert K.F.Teng, "A high performance VLSI architecture for integer motion estimation in HEVC", in IEEE 10th International Conference on ASIC (ASICON), October 2013, pp. 1-4.

[4] Wajdi Elhamzi, Julien Dubois, Johel Miteran and Mohamed Atri, "An efficient low-cost FPGA implementation of a configurable motion estimation for H.264 video coding", Journal of real-time image processing, vol. 9, no. 1, pp. 1930, 2014.

[5] J.Byun, Y.Jung and J.Kim, "Design of integer motion estimator of HEVC for asymmetric motion-partitioning mode and 4k-UHD", Electronics Letters, vol. 49, no. 18, pp. 1142-1143, 2013.

[6] Ching-Yeh Chen, Shao-Yi Chien, Yu-Wen Huang, Tung-Chien Chen, Tu-Chih Wang and Liang-Gee Chen, "Analysis and architecture design of variable block-size motion estimation for H.264/AVC", Circuits and Systems I: Regular Papers IEEE Transactions on, vol. 53, no. 3, pp. 578-593, 2006.

[7] Xilinx Zynq-7000, Zynq-7000 all Programmable SoC Overview, Advance Product Specification - DS190 (v1.2) available on: http://www.xilinx.com/support/documentation/data_sheets/-ds190-Zynq-7000-Overview.pdf, August, 2012

About Author (s):

Estefania Alcocer was born in Bigastro, Spain, in 1986. She received her M.S. degree on telecommunication engineering in 2010 from the Miguel Hernandez University, Elche, Spain, and she joined the GATCOM research group as PhD student in 2012.
Currently, she is assistant professor in the Department of Physics and Computer Architecture at Miguel Hernandez University, Elche since 2012. Her current research activities are related to image processing, the design of FPGA-based systems and video coding.

Otoniel Lopez-Granado received his M.S in Computer Science in 1996. Between 1997-2003 he worked as programmer analyst in an important industrial informatics firm.
In 2003 he joined to the Computer Engineering Departament at Miguel Hernandez University (UMH), Spain, as an assistant professor. Then, he received the phD degree in Computer Science in 2010. In 2012 he was promoted to associate professor. Currently, he leads the GATCOM research group (atc.umh.es) at Miguel Hernandez University. His research and teaching activities are related to multimedia networking (audio/video coding and network delivery).

Roberto Gutierrez was born in Orihuela, Spain, in 1977. He received his M.Sc. degree on telecommunication engineering in 2003, and the Ph.D degree in electronic engineering in 2011, both from the Universidad Politecnica de Valencia, Spain.
He is associate professor in the Department of Communication engineering at Universidad Miguel Hernandez, Elche since 2003. His current research interests include the design of FPGA-based systems, computer arithmetic, VLSI signal processing and digital communications.

Manuel Perez Malumbres received his B.S in Computer Science from the University of Oviedo (Spain) in 1986. In 1989 he joined to the Computer Engineering Department (DISCA) at Technical University of Valencia (UPV), Spain, as an assistant professor. Then, he received the M.S. and Ph.D. degrees in Computer Science from UPV, at 1991 and 1996 respectively.
He is a TC member of IEEE Multimedia Communications Group and associate editor of the Signal, Image and Video Processing journal. He was serving as TPC member of several relevant international Conferences related with his main research interests.
He is author of more than 160 conference and journal publications and several networking books for undergraduate CS courses.
Currently, his research and teaching activities are related to multimedia networking (image/video coding and network delivery) and wireless network technologies (MANETs, VANETs and WSNs).