A Highly Parallel SAD Architecture for Motion Estimation in HEVC Encoder

Ahmed Medhat, Ahmed Shalaby, Mohammed S. Sayed, Maha Elsabrouty

Egypt-Japan University of Science and Technology
P.O.Box 179, New Borg El-Arab City,
Alexandria 21934, Egypt
{ahmed.abdelsalam, ahmed.shalaby, mohammed.sayed, maha.elsabrouty}@ejust.edu.eg

Farhad Mehdipour

E-JUST Center, Kyushu University
3-8-33 Momochihama, Sawara-ku,
Fukuoka 814-0001, Japan
farhad@ejust.kyushu-u.ac.jp

Abstract—The high computational cost of the motion estimation module in the new HEVC standard raises the need for efficient hardware architectures that can meet the real-time processing constraint. In addition, targeting HD and UHD resolutions increases the motion estimation processing cost beyond the capabilities of the currently existing architectures. This paper presents a highly parallel sum of absolute difference (SAD) architecture for motion estimation in HEVC encoder. The proposed architecture has 64 PUs operating in parallel to calculate the SAD values of the prediction blocks. It processes block sizes from 4x4 up to 64x64. The proposed architecture has been prototyped, simulated and synthesized on Xilinx Virtex-7 XC7VX550T FPGA. At 458 MHz clock frequency, the proposed architecture processes 30 2K resolution fps with ± 20 pixels search range. The prototyped architecture utilizes 7% of the LUTs and 5% of the slice registers in Xilinx Virtex-7 XC7VX550T FPGA.

Keywords—HEVC, inter prediction, SAD architecture, variable block size motion estimation (VBSME)

I. INTRODUCTION

Recent estimates indicate that more than 50% of current network traffic is compressed real-time video, and this share is expected to rise to 90% within a few years [1]. In addition, the growing popularity of high definition (HD) videos and beyond HD videos as well as creating stronger needs for better video compression efficiency. These facts and needs raise the demand for new video coding standard with high compression efficiency compared to the currently used H.264/MPEG-4 AVC standard. 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 AVC for the same video quality [2]-[3].

Quad tree structure is the fundamental feature that differentiates HEVC from MPEG-4 AVC. As shown in Fig. 1, HEVC is based on code tree unit (CTU) instead of the macroblock in H.264/MPEG-4 AVC. The size of CTU is variable, unlike traditional macroblock. CTU size is selected by the encoder and can be larger or smaller than a traditional macroblock [2]. Each CTU is partitioned into one or more code units (CUs), and each CU has an associated partitioning

into prediction units (PUs) and a tree of transform units (TUs) [4]. Each PU is coded using either inter or intra prediction.

Motion estimation (ME) and motion compensation are the major loads at video encoder. They consume more than 90% of encoding time [5]. Although, HEVC provides a simple inter prediction process, the overhead involved is larger compared to H.264/MPEG-4, which consequently increases the complexity of HEVC encoder [6]. Typically as in H.264/MPEG-4 AVC, for every prediction block (PB) in HEVC, block matching algorithm (BMA) finds the best matching block within a certain search window. In the last decade, many hardware architectures for ME were proposed. Nevertheless, most of these architectures endeavor was H.264/MPEG-4 AVC but not HEVC.

In this paper, we propose a high performance hardware architecture, in terms of parallelism and computational complexity, for Sum of Absolute Difference (SAD) unit in HEVC. The proposed architecture has been implemented on FPGA and compared with other SAD units architectures for HEVC. Synthesis results show that the proposed architecture processes the video data with high processing rate than existing ones. Moreover, it can meet the requirements of 30 2K resolution frames per second (fps) real time video coding. The rest of the paper is organized as follows; HEVC ME is explained in section II. In section III, the related work is reviewed. The detailed description of the proposed SAD unit architecture is explained in section IV. Section V shows the simulation results and discussion. Finally, section VI concludes the paper.

Fig. 1. HEVC quad-tree structure and subdivision of CTU into CUs

II. HEVC MOTION ESTIMATION THEORY

Motion estimation is a significant stage in video coding as it utilizes and gets use of the temporal redundancies within a video sequence. The purpose of ME unit is to find the best matched block in a reference frame within a certain search window. This process is applied on every block at the current frame in order to find the lowest residual information at the reconstructed frame. Coding block (CB) in HEVC, unlike H.264/MPEG-4 AVC, has a variable size. For each CU, the size of its luminance CB can be defined as $2N \times 2N$, where $N \in \{4, 8, 16, 32\}$. CBs can be further divided into one, two or four prediction blocks (PB).

HEVC employs three different modes of partitioning that can be used to divide CBs into PBs. Square motion partition (Square) mode and symmetric motion partition (SMP) mode are adopted from H.264/MPEG-4 AVC to meet large sizes of CBs in HEVC. Asymmetric motion partition (AMP) mode is introduced to present irregular object boundaries with fewer bits than Square and SMP modes [7]. The supported modes include two Square modes ($PART_2Nx2N$ and $PART_NxN$), two SMP modes ($PART_Nx2N$ and $PART_2NxN$) and four AMP modes ($PART_2NxnU$, $PART_2NxnD$, $PART_nLx2N$, and $PART_nRx2N$) where, n represents smaller one of the partitions which equals to $N/2$, and U , D , L , and R denote up, down, left, and right, respectively [7].

HEVC defines certain constraints for partitioning CBs into PBs to reduce ME unit's complexity. One of these constraints is that HEVC disables AMP with $N = 4$. Table I lists the total number of different SADs combinations and partitions for 64x64 code tree block (CTB). Therefore, there are more than 600 blocks with different sizes at every candidate in a search window for every CTB. In full search algorithm, all possible candidates are checked to find the best matched block in terms of both the PSNR and the bit rate (BR) through the Lagrangian cost function. In HEVC, This Lagrangian optimization problem is proposed in [8] as

$$(r^*, m^*) = \arg \min_{r \in R, m \in M} D_K(r, m) + \lambda_M \cdot R_K(r, m) \quad (1)$$

Given a reference picture R and a candidate set M of motion vectors, the rate $R_k(r, m)$ represents an estimate of the number of bits to transmit the motion parameters. The distortion $D_k(r, m)$ is measured as the SAD value between 1) the predicted block which consists of motion vector $m = [m_x, m_y]$ and 2) the reference block in the reference picture indicated by reference index r , and λ_M represents the Lagrangian multiplier.

III. RELATED WORK

Motion estimation becomes more and more vital especially in HEVC that targets real time HD and beyond HD videos. Nevertheless, it takes the major computational and time complexity share in HEVC encoder. In addition, HEVC adopts variable block size motion estimation (VBSME) to obtain advanced coding efficiency. However, this comes at the expense of computational complexity and poses a challenge to hardware design and implementation. Profiling shows that ME unit reaches up to 96% of the encoding time [9], and SAD unit takes the majority of this complexity, especially in the case of

CTB size 64x64, which can be divided into more than 600 PBs with different sizes. Several hardware architectures were proposed to speed up the SAD unit for ME in HEVC [10-12].

For example in [10], Yuan et al. proposed a ME unit for HEVC that disregards calculating particular CTB partitions in order to speed up the ME unit. In addition, the largest CTB size that can be processed in [10] is 32x32 instead of 64x64. Although, this architecture reduces the complexity of the ME unit, it increases the required bit rate by 3% [11]. In [12], Nalluri et al. proposed a different SAD architecture for VBSME with parallel stages. It calculates SADs from 4x4 to 64x64 with AMP sizes. However, it cannot provide 30 2K resolutions fps real time video coding or higher. In addition, all proposed architectures [10-12] exploit parallelism at different levels with different strategies.

IV. PROPOSED PARALLEL SAD ARCHITECTURE

In this work, we present high-performance low cost SAD unit architecture for ME unit in HEVC that provides calculations for various sizes of SADs from 4x4 to 64x64. In addition, it exploits parallelism which is a main requirement for efficient HEVC implementation. The proposed design provides an optimized implementation for SAD unit in HEVC.

The abstraction levels of our implementation for highly parallel SAD unit are shown in Fig. 2. The architecture is composed of 1) current block memory, 2) reference block memory, 3) processing unit (PU), 4) a SAD of combination tree and 5) registers for storing the 316 Square and the 340 SMP SADs. Current and reference frames are stored in off-chip external memory. On the other hand, current CTB and its search window are stored in on-chip internal memory. A CTB with size of 64x64 needs 39KB to support its ± 64 search range [11]. Therefore, about 44KB on-chip memory and about 14MB off-chip memory are requisites to store reference and current frames with 2K resolution.

Concerning processing units, there are 64 PUs each contains four processor elements (PEs). A PU, shown in Fig. 3, calculates 4x4 block SAD every four cycles while PE, shown in Fig. 4, calculates the absolute difference between two pixels, one pixel from the current CTB and the other from the candidate CTB at the reference frame.

TABLE I. SUMMARY OF TOTAL NO. OF SADs FOR EACH PARTITION IN 64x64 CTB

Block Size	No. of SADs	Block Size	No. of SADs
64x64	1	8x32(left)	8
32x64	2	16x16	16
64x32	2	8x16	32
64x16(up)	2	16x8	32
64x16(down)	2	16x4(up)	32
16x64(right)	2	16x4(down)	32
16x64(left)	2	4x16(right)	32
32x32	4	4x16(left)	32
16x32	8	8x8	64
32x16	8	4x8	128
32x8(up)	8	8x4	128
32x8(down)	8	4x4	256
8x32(right)	8	-----	-----
Total No. of SADs	316 Square + 340 SMP + 168 AMP

Fig. 2. Architecture of highly parallel SAD unit

Fig. 3. The architecture of a PU

Fig. 3 shows the PU, group of four PEs, that calculates one row of 4x4 SADs every cycle. Hence, these values are stored in registers and accumulatively added to calculate the SAD of the whole row. Next, the SAD of the whole row is fed to the accumulator to be added incrementally with the SAD value of the next three rows to calculate the SAD value of a 4x4 block. Therefore, the output 4x4 SAD value is latched every four cycles using the 2-bit counter shown in Fig. 3. An extra logic gates shown in Fig. 3, are used to reset the accumulator every four cycles.

Fig. 4 shows the PE architecture that calculates the absolute difference between two pixels. We use 8-bit comparator based on the 2-bit comparator architecture that is proposed in [13]. The 8-bit comparator works to fetch two 8-bit pixels as inputs then it compares between these pixels as un-signed numbers. Hence, it returns the 1st complement of the smaller number, and the larger number as it is. In addition, when the two 8-bit pixels are not equal, the processor element returns a Not Equal (NE) signal. Following this, the outputs from the 8-bit comparator are added together plus one to compute the absolute difference. An extra logic gate is used to ensure that the absolute difference is set to zero when the values of the two pixels are equal.

The SAD combination tree combines the sixty-four 4x4 SADs from the PU to return all the combinations from 4x4 up to 32x32 every four cycles. PU is used four times sequentially

Fig. 4. Processor Element Architecture

Fig. 5. Large SADs calculations

Fig. 6. Logic circuit to alternate storing SADs at every four cycles at different registers

to calculate the remaining larger partitions (64x64, 64x32, 32x64). Consequently, it takes 16 cycles to calculate all possible partitions from 4x4 up to 64x64. Since the PU is used four sequential times, the SADs output from the PU at every four cycles have to be stored at different registers. Fig. 6 shows the proposed logic circuit which is used to store SADs values at every four cycles by using 4-bit counter. Fig. 5 shows the calculation process for the larger remaining partitions from 32x32 SAD values.

V. PROTOTYPING RESULTS AND DISCUSSION

The proposed architecture is implemented in Verilog HDL, and synthesized using Xilinx ISE. The target platform is Xilinx Virtex-7 XC7VX550T FPGA. In addition, simulation and function verification of our architecture were done using Xilinx integrated software environment (ISE) simulator. Our synthesis results are shown in Table II. It can be observed that our architecture can achieve a maximum frequency of 458 MHz where it utilizes 24,957 LUTs which represent 7% of LUT resources and 39,901 slice register bits which represent 5% of slice register resources.

TABLE II. SYNTHESIS RESULTS OF THE PROPOSED ARCHITECTURE

No. of slice registers	39901(5%)
No. of slice LUTs	24957(7%)
Maximum frequency	458.7 MHz
Available SAD partitions	4x4 up to 64x64
No. of processed 64x64 CTBs every second	28.7M CTBs

TABLE III. COMPARISON WITH YUAN ET AL. [10] ARCHITECTURE IMPLEMENTED ON VIRTEX-6 FPGA

	Yuan et al. [10]	Proposed Architecture
No. of slice registers	19744(2.9%)	32012(4%)
No. of slice LUTs	55346(16%)	15042(8%)
Maximum frequency	110 MHz	550 MHz
Available SAD partitions	4x8 up to 32x32	4x4 up to 32x32
No. of processed 32x32 CTBs every second	220M CTBs	275M CTBs

TABLE IV. COMPARISON WITH NALLURI ET AL. [12] ARCHITECTURE IMPLEMENTED ON VIRTEX-5 FPGA

	Nalluri et al. [12]	Proposed Architecture
No. of slice registers	20736(30%)	38032(55%)
No. of slice LUTs	15453(22%)	25173(36%)
Maximum frequency	172 MHz	348 MHz
Available SAD partitions	4x4 up to 64x64	4x4 up to 64x64
No. of processed 64x64 CTBs every second	2688K CTBs	21750K CTBs

From simulation results, we find that our architecture takes 16 cycles to obtain 316 Square and 340 SMP SADs partitions from 4x4 up to 64x64 using only one 32x32 PE four sequential times which takes four cycles at each to calculate its SADs. Consequently, our proposed architecture can process up to 30 2K resolution fps with ± 20 pixels search range that represents 104x104 pixels search window. The architecture can be faster if more parallel PEs are used, but, it comes at the expense of additional hardware resources. In order to process higher resolutions, more parallel units of the proposed architecture are needed.

Table III and Table IV compare our architecture with previous state-of-the-art architectures implemented on different FPGA platforms. In order to realize fair comparison, we modified our proposed architecture to be proportioned to other architectures that we are comparing with. In addition, it was synthesized on the same FPGA platforms used by the architectures compared to. Comparing with [10] in Table III, the modified proposed architecture with two parallel 32x32 PEs achieves better results in terms of computational and time complexity. The operating frequency of the proposed architecture witnessed a significant increase to be five times faster than in [10]. This is due to the fact that the proposed architecture uses less hardware resources in terms of LUTs, and achieves better processing rate for more SAD partitions by around 125%. In addition, the 30% slice registers increase compared to [10] comes from that our architecture supports more SADs partitions. Our proposed architecture operates on higher frequency when it is used to achieve only 32x32 SADs but not 64x64. Compared with [12] as shown in Table IV, our architecture achieves higher processing rate by around 386% but it comes at the expense of the hardware resources as the additional required resources are around 80% more.

VI. CONCLUSION

An efficient real-time parallel SAD architecture for ME in HEVC has been presented. The proposed architecture

calculates 316 Square and 340 SMP SAD values for block sizes from 4x4 up to 64x64 using SAD combination tree. A group of 64 PUs operates in parallel and calculates 32x32 block size's SAD values with all of its partitions every 4 clock cycles. Hence, these 64 PUs are used 4 times to calculate the 64x64 block size every 16 clock cycles. The proposed architecture has been prototyped, simulated and synthesized on the Xilinx Virtex-7 XC7VX550T FPGA. At 458 MHz clock frequency, the proposed architecture processes 30 2K resolution fps with ± 20 pixels search range that represents 104x104 pixels search window.

ACKNOWLEDGMENT

We would like to thank Egypt-Japan University of Science and Technology (E-JUST) for the continuous support and the Egyptian Ministry of Higher Education for funding this research.

REFERENCES

- [1] P. Fröjdh, A. Norkin and R. Sjöberg, "Next Generation Video Compression," Ericsson Review, The Communications Technology Journal, pp. 1-8, April 2013.
- [2] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 12, pp. 1649-1668, December 2012.
- [3] I. Richardson, "HEVC: An introduction to high efficiency video coding," 2001, <https://www.vcodex.com/h265.html>
- [4] S. Tai, C. Chang, B. Chen and J. Hu, "Speeding up the decisions of quad-tree structures and coding modes for HEVC coding units," Advances in Intelligent Systems & Applications (SIST): Springer Berlin Heidelberg, vol. 21, pp. 393-401, December 2012.
- [5] T. Wiegand, G. J. Sullivan, G. Bjontegaard, A. Luthra, "Overview of the H.264/AVC video coding standard," IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 7, pp. 560-576, July 2003.
- [6] F. Bossen, B. Bross, K. Sühring, and D. Flynn, "HEVC complexity and implementation analysis," IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 12, pp. 1684-1695, December 2012.
- [7] J. Vanne, M. Viitanen and T. D. Hämäläinen, "Efficient mode decision schemes for HEVC inter prediction," IEEE Trans. Circuits Syst. Video Technol., vol. PP, no. 99, February 2014.
- [8] J.-R. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, "Comparison of the coding efficiency of video coding standards including High Efficiency Video Coding (HEVC)," IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 12, pp. 1668-1683, December 2012.
- [9] F. Sampaio, S. Bampi, M. Grellert, L. Agostini and J. Mattos, "Motion vectors merging: low complexity prediction unit decision heuristic for the inter-prediction of HEVC encoders," IEEE International Conference on Multimedia and Expo (ICME), pp. 657-662, July 2012.
- [10] X. Yuan, L. Jinsong, G. Liwei, Z. Zhi and R. Teng, "A high performance VLSI architecture for integer motion estimation in HEVC," IEEE 10th International Conference on ASIC (ASICON), October 2013.
- [11] M. E. Sinangil, V. Sze, M. Zhou and A. P. Chandrakasan, "Memory cost vs. coding efficiency trade-offs for HEVC motion estimation engine," IEEE Int. Conf. ImageProcess. (ICIP), pp. 1533-1536, October 2012.
- [12] P. Nalluri, L. N. Alves, A. Navarro, "A novel SAD architecture for variable block size motion estimation in HEVC video coding," IEEE International Symposium on System on Chip (SoC), October 2013.
- [13] L. Yufei, F. Xiubo and W. Qin, "A high-performance low cost SAD architecture for video coding," Consumer Electronics IEEE Transactions, vol. 53, no. 2, pp. 535-541, May 2007.