An Optimized Partial-Distortion-Elimination Based Sum-of-Absolute-Differences Architecture for High-Efficiency-Video-Coding

Paolo Selvo[1], Maurizio Masera[1], Riccardo Peloso[1], Guido Masera[1], Muhammad Shafique[2], and Maurizio Martina[1]

[1] Department of Electronics and Telecommunications, Politecnico di Torino, Italy, name.surname@polito.it

[2] Institute of Computer Engineering, Vienna University of Technology (TU Wien), Austria, muhammad.shafique@tuwien.ac.at

Abstract. Sum of Absolute Differences (SAD) is one of the most time consuming tasks in video coding. This paper proposes an architecture to compute the SADs for all the different block sizes required by the High Efficiency Video Coding (HEVC) standard. Moreover, the Partial Distortion Elimination (PDE), clock gating and a low leakage technology enable high power/energy reductions/savings over the state of the art.

Keywords: VLSI architecture, motion estimation, HEVC

1 Introduction

High Efficiency Video Coding (HEVC) is the latest video compression standard developed by the Joint Collaborative Team on Video Coding (JCT-VC) [10]. The encoding process relies on inter prediction to exploit the temporal redundancy between successive frames, by implementing the Motion Estimation (ME). ME is performed on the Prediction Unit (PU) basis, where PUs are built by splitting the current Coding Unit (CU) into one, two or four PUs, as shown in Fig 1 (a). During the Integer Motion Estimation (IME) phase, the encoder explores a search window in the reference frames in order to find the best match with the current block (see Fig. 1 (b)). This operation is repeated for each PU by computing the Sum of Absolute Differences (SAD) between the samples of current block and of the candidate blocks from the reference frames. The SAD metric is calculated as $SAD = \sum_{i=1}^{W} \sum_{j=1}^{H} |C_{i,j} - R_{i,j}|$, where and $C_{i,j}$ and $R_{i,j}$ denote each pixel of the current and the reference blocks, respectively, which size is $W \times H$. Therefore, the reference block, which produces the minimum SAD, is chosen as predictor of the current block. As argued in [1], the calculation of SAD and other distortion metrics can take up to 40% of the encoding time and about 80% of the energy budget [4], thus being one of the most time consuming and energy demanding tasks at the encoder side. For this reason, several architectures have been proposed in the open literature to accelerate the SAD computation

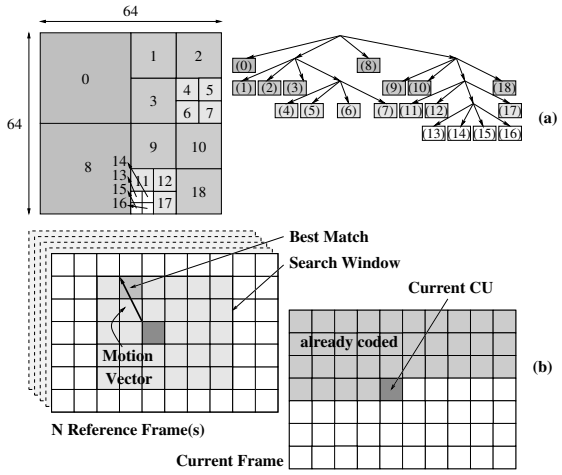

Fig. 1. The organization of the CU in HEVC (a). ME process (b).

in HEVC. In [7] an high speed SAD architecture for FPGA which supports all the PU sizes specified in HEVC has been proposed by relying on parallel computation of four 8×8 PU. Then, the works in [5, 6] implement two very fast accelerators on FPGA and ASIC respectively, which are able to compute the 64×64 SAD in only 16 clock cycles. However, neither bandwidth constraints nor real memories have been considered in these works, thus leading to implementations which are far from being easily integrated in a system. A slower, but more realistic hardware SAD architecture has been proposed in [3], which requires 32 clock cycles to compute the 64×64 SAD. Besides, Partial Distortion Elimination (PDE) is a very effective technique to improve the performance while not affecting the coding efficiency [2]. Since the IME aims to find the best match that minimizes the SAD cost function, one can stop the computation when a partial result is already larger than the current minimum SAD value. PDE has been recently exploited in [8] only for 4×4 PUs. Therefore, this paper proposes a reconfigurable and low-power SAD architecture, which extends the PDE approach to all the PU sizes specified in HEVC.

2 Proposed Architecture

The proposed architecture, which computes the SAD for all the PU sizes listed in the first column of Table 1, is made of three main blocks: three SRAMs, a control unit, named SAD interface, and a pipelined SAD datapath, as shown in Fig. 2. Moreover, a minimum SAD updater module handles the PDE, when it is activated, namely, it stores the current minimum SAD value and provides the comparison value to the comparators inside the datapath.

The maximum input sample rate has been fixed to about 50 GB/s, so that the external memory can be a common SDRAM running at 1280 MHz (*clock1*) with

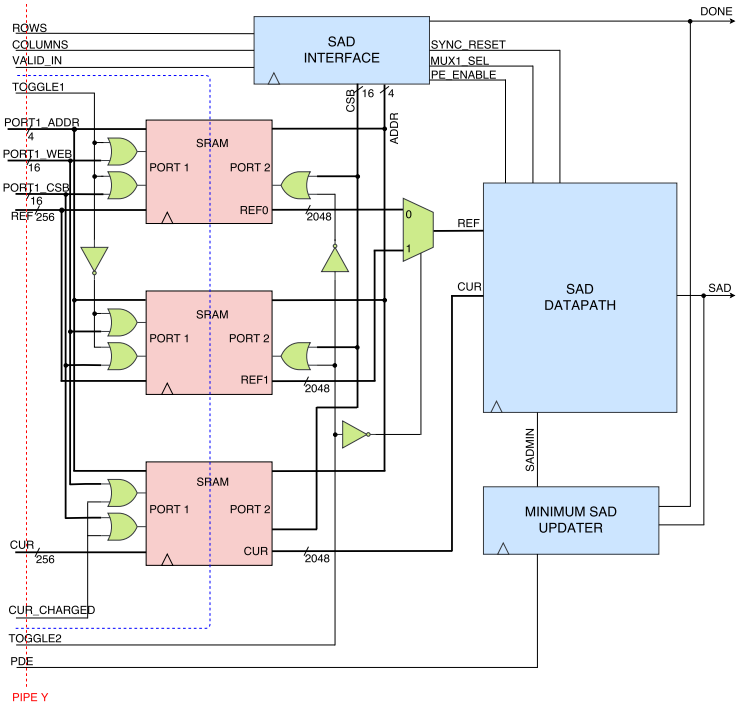

Fig. 2. Proposed architecture block scheme

a 256 bit data bus. As a consequence, with a clock frequency equal to 160 MHz (*clock2*), the architecture can read 2048 bits (256 samples) per clock cycle from the SRAMs, where the reference and the current PU are stored. Two SRAMs for the reference PU allow the SAD datapath to work every clock cycle avoiding pipeline stalls, namely, the two SRAMs are used in an interleaved fashion so that while the first one provides data to the datapath (reading), the second one loads a new frame from the external SDRAM (writing). The dashed blue line in Fig. 2 is used to separate the two clock domains. The data stored in the local SRAM are arranged as sets of 4×4 samples, so the datapath is made of 16 Processing Elements (PEs) working concurrently, as shown in Fig. 3. Each PE computes a 4×4 SAD, therefore, it contains an adder-tree made of 15 adders. At each clock cycle, up to 16 sets of 4×4 samples from the current and reference PUs are sent to the datapath. As a consequence, a 16×16 SAD requires one cycle to be computed, whereas a 64×64 SAD needs 16 clock cycles. This aspect is handled by the multiplexer in the upper-right part of Fig. 3 (before the SAD register), which selects the correct exit point.

It is worth noting that, each pipeline register (dashed red line in Fig. 3) has its own enable port (blue lines in Fig. 3), which allows for clock gating optimization to reduce the power consumption. Finally, five comparators have been placed in the datapath (see Fig. 3), so allowing to implement PDE, namely if one of the

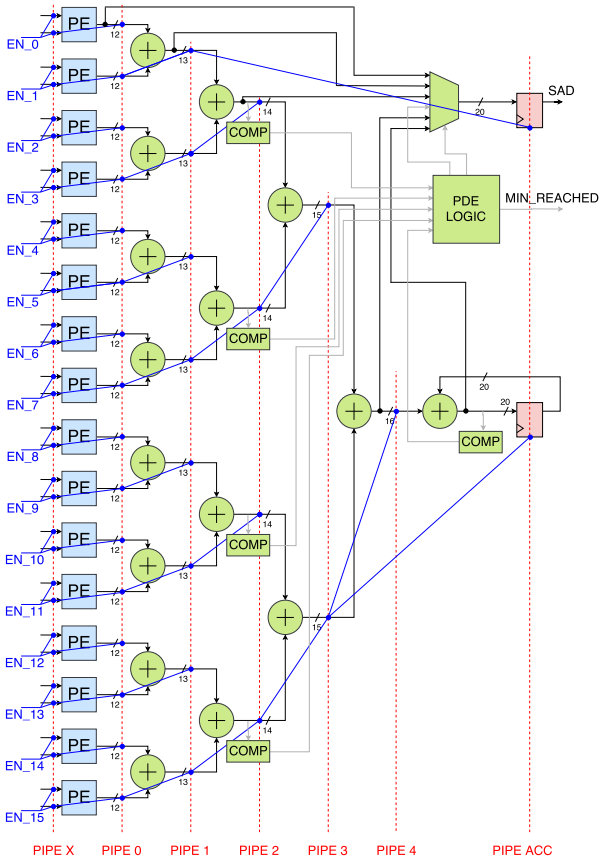

Fig. 3. Datapath block scheme

comparators detects a value greater than the current minimum SAD, then the SAD computation is stopped. Extensive simulations have shown that there is no significant latency reduction by adding other comparators inside the datapath.

3 Implementation Results

The proposed architecture has been described in VHDL [9] and synthesized on a CMOS 65 nm standard cell technology using SRAM IPs. Timing simulations have been performed by computing the SADs on successive frames of 1920×1080 standard sequences (e.g. *BQTerrace*). The SADs have been computed with the accelerator either activating or deactivating the PDE technique, thus allowing to measure the average number of clock cycles needed to calculate the SAD for each PU. Table 1 reports the total number of clock cycles required to compute the SAD on PUs of different sizes without and with PDE, respectively: as expected, the speedup offered by PDE increases with PU sizes. Power consumption results

Table 1. Timing, power and energy results for different SAD cases

SAD size	Clock cycles w/o PDE	Clock cycles w/ PDE	Time variation	Power w/o PDE [mW]	Power w/ PDE [mW]	Power variation	Energy w/o PDE [pJ]	Energy w/ PDE [pJ]	Energy variation
8×4	6.0	6.0	0.0%	12.10	12.12	0.2%	454	455	0.2%
8×8	7.0	7.0	0.0%	15.30	15.30	0.0%	669	669	0.0%
16×4	7.0	7.0	0.0%	15.42	15.43	0.1%	675	675	0.1%
16×8	8.0	7.2	-10.0%	21.77	23.20	6.6%	1089	1044	-4.1%
16×12	9.0	7.6	-15.6%	33.47	36.97	10.5%	1883	1756	-6.7%
16×16	9.0	7.8	-13.3%	33.82	36.16	6.9%	1902	1763	-7.3%
32×8	9.0	7.7	-14.4%	33.88	37.65	11.1%	1906	1812	-4.9%
32×16	11.0	9.4	-14.5%	40.56	44.74	10.3%	2789	2628	-5.7%
32×24	12.0	10.2	-15.0%	48.39	53.36	10.3%	3629	3402	-6.3%
32×32	13.0	10.8	-16.9%	54.02	60.58	12.1%	4389	4089	-6.8%
64×16	13.0	12.0	-7.7%	54.45	61.95	13.8%	4424	4646	5.0%
64×32	17.0	14.0	-17.6%	72.34	87.15	20.5%	7686	7626	-0.8%
64×48	21.0	15.8	-24.8%	86.02	98.67	14.7%	11290	9744	-13.7%
64×64	25.0	18.4	-26.4%	92.49	106.28	14.9%	14452	12222	-15.4%

have been obtained by extracting the switching activity of the circuit when running 1024 SAD test vectors for each PU. Since the architecture is parallel and pipelined, the clock gating technique has been employed to reduce the power consumption for the small PUs, which do not use all the datapath elements. This technique is effective for the PUs from 8×4 to 16×8, which save from the 24.5% to the 8.3%, respectively. Clock gating costs about the 0.4% of the total area, thus being negligible. Despite the use of PDE increases the power consumption as the PU size increases, the achieved speed up allows for an interesting energy reduction, ranging from about 4% to 15%.

A fair comparison of the proposed solution with those proposed in the literature is not straighforward. Despite several architectures in the literature, as well as the proposed one, employ 16 clock cycles in the datapath to compute a 64×64 SAD, no information about the number of clock cycles spent by the control unit and to load/store samples from/to the memory is given in the literature (the proposed architecture requires at most 25 clock cycles, 18.4 with PDE). Moreover, the architectures in [7], [6] and [5] achieve very high clock frequencies, thus their delay for a 64×64 SAD are respectively 96.63 ns, 34.88 ns and 22 ns. However, these works do not specify the required input bandwidth and they do not explain how to bring to their architecture all the required samples. As far as power consumption is concerned, the SAD architecture in [7] features a power consumption that is three times higher than the one of the proposed architecture. Moreover, it becomes 4 times higher if on-chip memory is not considered. Finally, the proposed architecture requires an area which is equal to 1.2 mm^2, corresponding to 90 k gates, which is more than four times lower than the area reported in [5].

4 Conclusion

This paper presented an energy efficient ASIC architecture, to perform all the SAD operations required by the HEVC standard. Optimized memory management and a parallel datapath allow to achieve a high throughput, while the PDE method permit to further speed-up the computation and to save energy, leading to an area and energy efficient solution.

References

1. Bossen, F., Bross, B., Suhring, K., Flynn, D.: HEVC Complexity and Implementation Analysis. IEEE Transactions on Circuits and Systems for Video Technology 22(12), 1685–1696 (Dec 2012)
2. Chiu, M.Y., Siu, W.C.: New results on exhaustive search algorithm for motion estimation using adaptive partial distortion search and successive elimination algorithm. In: 2006 IEEE International Symposium on Circuits and Systems. pp. 4 pp.–3981 (May 2006)
3. Dinh, V.N., Phuong, H.A., Duc, D.V., Ha, P.T.K., Tien, P.V., Thang, N.V.: High speed SAD architecture for variable block size motion estimation in HEVC encoder. In: 2016 IEEE International Conference on Communications and Electronics. pp. 195–198 (July 2016)
4. El-Harouni, W., Rehman, S., Prabakaran, B.S., Kumar, A., Hafiz, R., Shafique, M.: Embracing approximate computing for energy-efficient motion estimation in high efficiency video coding. In: Design, Automation and Test in Europe Conference. pp. 1384–1389 (Mar 2017)
5. Medhat, A., Shalaby, A., Sayed, M.S.: High-throughput hardware implementation for motion estimation in HEVC encoder. In: IEEE International Midwest Symposium on Circuits and Systems. pp. 1–4 (Aug 2015)
6. Medhat, A., Shalaby, A., Sayed, M.S., Elsabrouty, M., Mehdipour, F.: A highly parallel SAD architecture for motion estimation in HEVC encoder. In: IEEE Asia Pacific Conference on Circuits and Systems. pp. 280–283 (Nov 2014)
7. Nalluri, P., Alves, L.N., Navarro, A.: High speed SAD architectures for variable block size motion estimation in HEVC video coding. In: IEEE International Conference on Image Processing. pp. 1233–1237 (Oct 2014)
8. Seidel, I., Brascher, A.B., Guntzel, J.L.: Combining pel decimation with partial distortion elimination to increase SAD energy efficiency. In: International Workshop on Power and Timing Modeling, Optimization and Simulation. pp. 177–184 (Sept 2015)
9. Selvo, P.: VHDL code of an optimized SAD architecture for HEVC (Oct 2017), `http://personal.det.polito.it/maurizio.martina/hevc.html`
10. Sullivan, G.J., Ohm, J.R., Han, W.J., Wiegand, T.: Overview of the High Efficiency Video Coding (HEVC) Standard. IEEE Transactions on Circuits and Systems for Video Technology 22(12), 1649–1668 (Dec 2012)