

Grouped Approach for the Design of H.264/AVC Motion Estimation Architectures

Sebastián López, Gustavo M. Callicó, Félix Tobajas, Valentín de Armas, José F. López, and Roberto Sarmiento

ABSTRACT—This letter presents a novel approach for organizing computational resources into groups within H.264/AVC motion estimation architectures, leading to reductions of up to 75% in the equivalent gate count with respect to state-of-the-art designs.

Keywords—H.264/AVC, motion estimation architecture, video encoder, CMOS.

I. Introduction

Variable-block-size motion estimation improves the rate-distortion performance of H.264/AVC video encoders by partitioning each macroblock (MB) of the video sequence to be encoded into variable block sizes according to the seven motion estimation modes defined by the standard: 4×4 , 4×8 , 8×4 , 8×8 , 8×16 , 16×8 , and 16×16 . Nevertheless, this kind of partition implies that H.264/AVC video encoders should compute 41 motion vectors for each MB, making motion estimation the most computationally intensive process in the encoders. For this reason, various research groups have recently developed several architectures for real-time applications, mainly composed of an array of processing elements (PEs) which compute the sum of absolute differences (SAD) between the current MB and each candidate position within the search area. Recent developments have focused on the topology considered for the array and/or the strategy for distributing the data among the different PEs, obtaining different results in terms of hardware cost and throughput. However, such methods suffer from the adoption of a rigid

approach in order to allocate the computational resources within the architecture without exploring different grouping possibilities of the PEs involved, thus, giving little room for further architectural improvements.

II. H.264/AVC Motion Estimation Architectures

The majority of the previously published architectures for implementing the motion estimation process stated by the H.264/AVC standard are based on the full search algorithm due to its highly regular behavior, which allows the reuse of the results obtained for 4×4 blocks to compute the motion vectors for the rest of the motion estimation modes. Depending on the topology adopted to organize the PEs within the motion estimation array, these architectures have been traditionally classified as one-dimensional (1-D) or two-dimensional (2-D) architectures. This letter proposes a complimentary classification attending to the distribution of the search positions to be evaluated among the different PEs of the array, distinguishing between grouped and non-grouped architectures. In non-grouped architectures, each PE is responsible for computing the SAD of one or more search positions by itself, while in the grouped ones, each PE collaborates with at least one other PE in the array to compute the SAD associated with one or more search positions.

The 2-D H.264/AVC motion estimation architectures recently published in [1], [2], and [3] represent clear examples of grouped architectures. These architectures are basically composed of a 16×16 array of PEs and an adder tree which is responsible for computing the SADs associated with all the H.264/AVC block sizes by reusing the SADs of the individual 4×4 blocks computed by the array. In this case, the advantages of a grouped approach are evident as the 256 PEs collaborate in

Manuscript received July 30, 2008; revised Sept. 24, 2008; accepted Oct. 21, 2008.

This research was funded by the Spanish government under the project TEC2005-08138.

Sebastián López, Gustavo M. Callicó, Félix Tobajas, Valentín de Armas, José F. López, and Roberto Sarmiento are with the Institute for Applied Microelectronics, University of Las Palmas de Gran Canaria, Las Palmas de Gran Canaria, Spain.

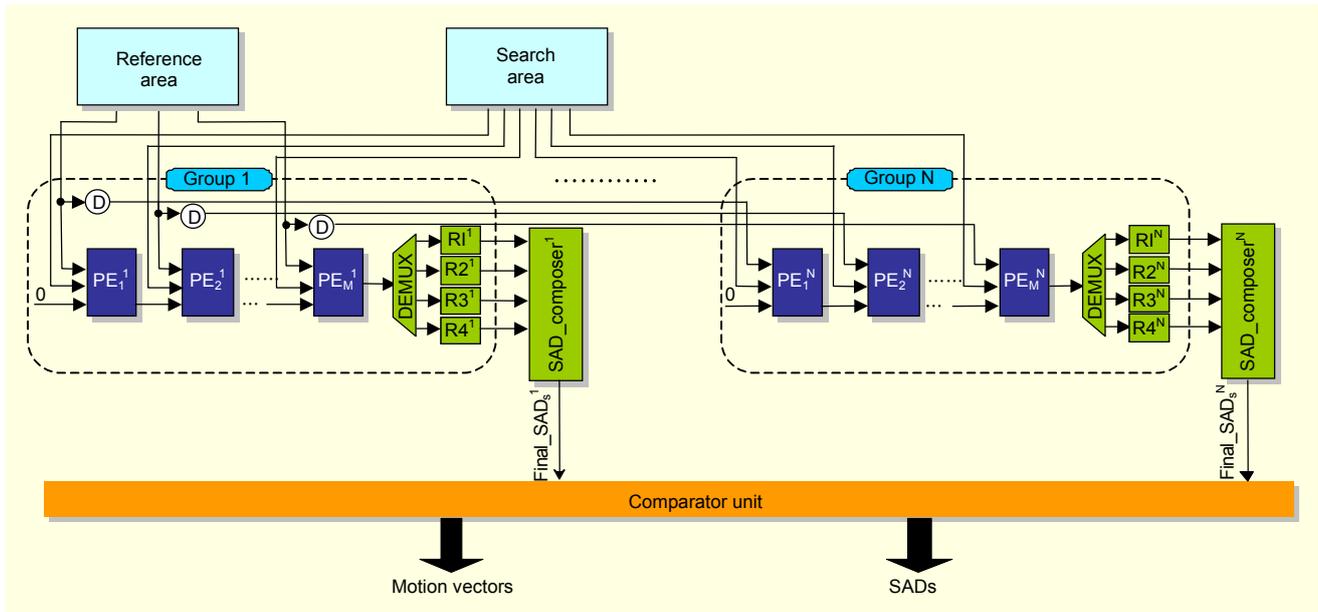


Fig. 1. Proposed architectural template.

computing the SADs associated with the sixteen 4×4 blocks that compose a whole MB; thus, no intermediate registers are needed to compose the SADs of the remaining motion estimation modes.

On the other hand, the recent H.264/AVC 1-D motion estimation architectures published in [4] and [5], composed of only 16 PEs in order to reduce the silicon area, are non-grouped architectures. However, none of these works establishes a clear justification of the approach followed, as they adopt *ad hoc* solutions for the allocation of the PEs within the array without exploring alternative options. For this reason, it is necessary to study the impact on the hardware cost of the different grouping options for 1-D arrays, as they are intended for applications in which the final circuit area represents the key performance factor.

III. Proposed Architectural Framework

Figure 1 shows the proposed architecture with a flexible 1-D architectural template that consists of N groups of M processing elements each. The proposed architecture is able to compute the 41 motion vectors established by the H.264/AVC standard for the MB stored in the reference area cache memory with respect to the search area stored in the search area cache memory. Each group of PEs is in charge of computing the SAD associated with a subset of MB positions within the search area. Figure 2 summarizes this process, where correlative bubbles denote the search positions evaluated by each group of PEs.

To perform this task, each PE computes the absolute difference between one pixel of the reference block and one

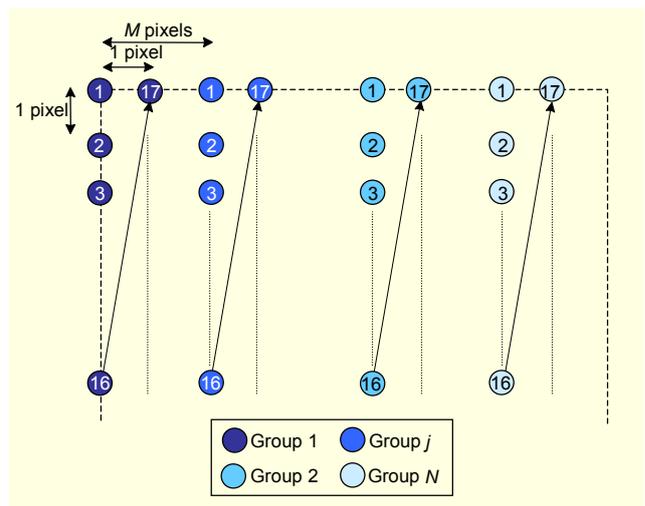


Fig. 2. Search positions evaluated by each group of PEs.

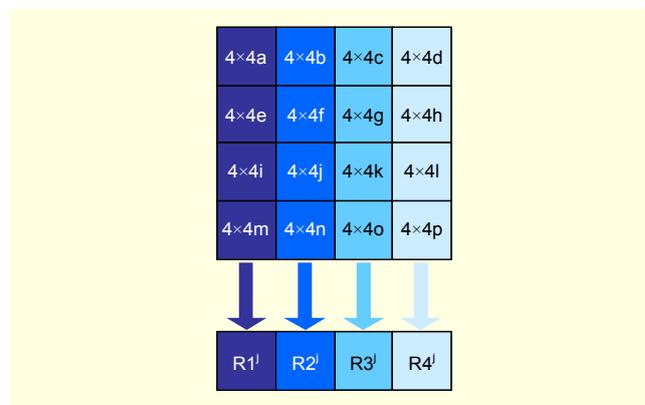


Fig. 3. Strategy for storing the SADs computed by each PE group.

pixel from the search position under evaluation at each clock cycle. Additionally, each PE sends the result of this operation to the next PE in the same group for it to be added to the result obtained by the latter. Four 12-bit registers (named R1, R2, R3, and R4 in Fig. 1) and one 1-to-4 demultiplexer (DEMUX) have been incorporated into each group of PEs so that they can easily store the SAD of each 4×4 block in a 4-block row separately as shown in Fig. 3.

To compose the SADs corresponding to all the possible motion vectors for the inspected position, the SAD composer module associated with each group uses the results stored in these registers. Finally, each SAD composer transfers these SADs to the comparator unit, giving 41 motion vectors with their associated minimal SADs as a result.

IV. Results

The previously mentioned 1-D template was implemented in HDL at the RTL level to evaluate three representative grouping options in terms of the (M, N) tuple: (1, 16), (16, 1), and (4, 4). These three architectures were synthesized using 0.25 μm CMOS standard cell technology with a clock frequency of 100 MHz, allowing real-time processing of CIF video sequences at 60 frames per second with a search area of 31×31 pixels. Table 1 outlines the results obtained in terms of NAND2 equivalent gates, together with the results given in [4] and [5], which are somehow equivalent to the (1, 16) architecture, as both represent non-grouped arrays. To evaluate the overall hardware cost of the proposed solutions, Table 1 highlights the requirements of the reference and search area memories in terms of pixels to be read by the clock cycle. This is important as all the architectures detailed in Table 1 have the same number of PEs and the same processing capability in terms of search positions evaluated per second. As seen in this table, the final gate count is reduced by 45% (proposed (4, 4) architecture against reference [4]) to 75% (proposed (16, 1) architecture against reference [5]) with the proposed grouped approach, and the rate of pixels to be read from both cache

memories is kept within affordable limits. In addition, as the total number of PEs is the same for all cases, no throughput penalties resulted for the (4, 4) and (16, 1) grouped arrays when compared with previously published non-grouped 1-D arrays.

V. Conclusion

This letter has introduced a novel approach for the design of 1-D H.264/AVC motion estimation architectures, based on different grouping alternatives for the PEs within the array. The results demonstrate that a significant reduction of the equivalent gate count is obtained by using any of the two new proposed grouped architectures when compared with previous non-grouped architectural approaches.

References

- [1] L. Deng et al., "An Efficient Hardware Implementation for Motion Estimation of AVC Standard," *IEEE Trans. on Consumer Electronics*, vol. 51, no. 4, Nov. 2005, pp. 1360-1366.
- [2] M. Kim, I. Hwang, and S.I. Chae, "A Fast VLSI Architecture for Full-Search Variable Block Size Motion Estimation in MPEG-4 AVC/H.264," *Proc. of Asia and South Pacific Design Automation Conf.*, 2005, pp. 631-634.
- [3] C. Wei and M.Z. Gang, "A Novel VLSI Architecture for VBSME in MPEG-4 AVC/H.264," *Proc. of IEEE Int'l symp. on Circuits and Systems*, 2005, pp. 1794-1797.
- [4] S.Y. Yap and J.V. McCanny, "A VLSI Architecture for Variable Block Size Video Motion Estimation," *IEEE Trans. on Circuits and Systems II*, vol. 51, no. 7, July 2004, pp. 384-389.
- [5] C.L. Hsu, M.H. Ho and M.K. Liu, "High-Efficient Mode Decision Design for Motion Estimation in H.264," *Proc. of IEEE Int'l Conf. on Consumer Electronics*, 2007, pp. 1-2.

Table 1. Comparison with previously published works.

	NAND2 gates	Pixels/clk cycle from ref. mem.	Pixels/clk cycle from search mem.	Number of PEs
(1, 16) arch.	69.82k	1	2	16
(4, 4) arch.	33.41k	4	8	16
(16, 1) arch.	21.30k	16	16	16
[4]	61k	1	2	16
[5]	88k	1	2	16