

# Efficient Hardware Implementation for H.264/AVC Motion Estimation

Mahdi Nazm Bojnordi, Mehdi Semsarzadeh, Mahmoud Reza Hashemi, Omid Fatemi  
Nano-Electronics Center of Excellence, School of Electrical and Computer Engineering  
University of Tehran, Tehran, Iran.  
{m.bojnordi, m.semsar}@ece.ut.ac.ir, hashemi@comnete.com, omid@fatemi.net

**Abstract**— Variable block size motion estimation (VBSME) is adopted in H.264/AVC to improve the coding efficiency. However, supporting various block sizes significantly increases the complexity of both video encoding and decoding. In this paper a multi-level parallel architecture for H.264/AVC motion estimation is proposed. A SIMD architecture is proposed for absolute differentiator and accumulator (ADA). Using the ADA as the main processing engine, a cost and performance SAD processor is proposed. Experimental results indicate that more than 100% performance improvement is achieved by the proposed architectures compared to the state-of-the-art architectures with similar resources.

**Keywords**— H.264/AVC, VBSME, parallel architecture

## I. INTRODUCTION

Motion estimation techniques are used in video compression algorithms to achieve high compression ratios. The block-based schemes are the most commonly used ones, since they are easy to implement, provide good compression, and provide reasonable video reconstruction quality [1]. Experimental results indicate that up to 80% of the H.264/AVC encoder computational load is caused by the motion estimation (ME) module [2]. Recently, many fast mode decision and ME algorithms have been proposed in the literature [3]-[9]. Real-time requirements of the new video applications, however, drive the designers towards implementing dedicated hardware accelerators for motion estimation. Therefore, many VLSI architectures have been proposed by the designers for motion estimation [10]-[13].

Full search block matching algorithm (FSBMA) is the most efficient algorithm for hardware implementation because of superior encoding efficiency and regularity. Variable block size motion estimation (VBSME) algorithms that are employing the FSBMA approach offer the highest efficiency and more precise estimation during the video encoding. Such algorithms, however, suffer from large amount of redundant computations. This paper proposes an efficient VLSI architecture for VBSME in H.264/AVC. Efficient parallelism and pipelining schemes are harnessed to achieve the highest data reuse and minimum operation redundancy. A single instruction parallel data (SIMD) architecture is proposed for the absolute differentiator and accumulator unit – named ADA- that is the basic unit of the

motion estimation architecture. An array of ADA modules, i.e. processing engines (PE), is used to compose a processing unit (PU). Using the PU, as the main building block, a high-performance architecture is presented. Typically, more than 100% improvements in the processing is achieved by the proposed architecture compared to the state-of-the-art designs.

This paper is organized as follows. The H.264/AVC VBSME algorithm is overviewed and discussed in Section II. The proposed architecture is described in Section III. In Section IV the experimental results and comparison with the existing architectures are presented. Finally, the paper ends with conclusion in Section V.

## II. H.264/AVC MOTION ESTIMATION

### A. The Algorithm Description

In H.264/AVC seven different block sizes are specified for inter frames prediction [1]. A block can be of size  $16 \times 16$ ,  $16 \times 8$ ,  $8 \times 16$ , or  $8 \times 8$ , and each  $8 \times 8$  block can be further broken down into sub-macro-blocks of size  $8 \times 8$ ,  $8 \times 4$ ,  $4 \times 8$ , or  $4 \times 4$ . Although VBSME can achieve higher compression ratio, it not only requires huge computations but also increases the difficulty of implementation ME in hardware.

To find the best mode for a macroblock a rate-distortion algorithm, determines the cost of each mode, and gives the lowest cost using the following equation:

$$RDCost = Distortion + \lambda_{mode} \times Rate. \quad (1)$$

Where, *Rate* represents the required bits for encoding the macro-block header, the motion information and the transformed coefficients.  $\lambda_{mode}$  is the Lagrange parameter which depends on the quantization parameter (QP) and *Distortion* is defined as the sum of squared differences between the original block and reconstructed one.

For each mode, the given cost by equation (1) is very dependent on the optimal motion vector estimated using the motion estimation operation. To find the best motion vector, in the motion estimation phase, cost is calculated as follows: