An Efficient Self-Transposing Memory Structure for 32-bit Video Processors

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

***Abstract*— Many 2-D data processing applications can be simplified and represented by use of 1-D operations. Such tools, however, require applying both vertical and horizontal operations to the data blocks. The data transposing units is preferred to be used by the designers rather than applying individual operations for horizontal and vertical directions. Hence, designing a cost efficient and extendible transposing memory is a key issue for these applications. This paper proposes an efficient management strategy for using the SRAM modules in order to make a self-transposing memory architecture (STMA). In addition to its lower cost compared to flip-flop based buffers, the proposed architecture is more than 29% faster than usual SRAM based memory units. Simulations indicate that using the STMA in the H.264/AVC deblocking filter results in 60% speed improvement.**

***Keywords*—Self-Transposing memory**

I. INTRODUCTION

Many signal processing algorithms need to apply very complex 2-D operations to blocks of data. Representing 2-D operations by simpler 1-D ones have been widely considered in the literature to decrease the algorithm complexities [1]-[3]. These algorithmic modifications significantly reduce the implementation complexity, as well. In many memory bound applications, like video and image processing, 1-D tools are still the processing bottlenecks for real-time implementations. Hence, these tools have been targeted by the designers for cost and speed optimization.

In these applications, the newly defined 1-D operations are sequentially applied to horizontal and vertical lines of the desired data block. In order to decrease cost of the hardware implementation, horizontal and vertical operations are considered to be similar. Therefore, a processing engine is required for processing the data block in both horizontal and vertical directions. Also, a memory management is necessary to select the proper data line for the processing engine. Therefore, these architectures are composed of two main parts:

- *Transposing Unit (TU)*: transposes the data block and prepares appropriate data for both horizontal/vertical processing modes.
- *1-D Processing Engine (1-DPE)*: applies the required vertical and horizontal operations to the data lines of the input matrix.

The TU is responsible for transposing the data block in order to switch between the vertical and horizontal processing modes of the 1-DPE. In the TU, data accessing strategy is very important. Using simple memories results in decreasing the hardware area and power consumption. In turn, the processing speed decreases. In order to increase the processing speed, multiple access memory modules, e.g. register files, can be employed. Using these architectures increases hardware area and power consumption and hence, they are not suitable for low power applications.

In this paper, an efficient 32-bit self-transposing memory architecture (STMA) for using in video and image applications is proposed. STMA uses the SRAM modules for maintaining the block data; hence, it is cost-efficient. It can be adopted in various tools of image and video processing architectures, like DCT, IDCT, and deblocking filter. Based on the work in [4], a 1-D 32-bit processing engine for H.264/AVC deblocking filter is implemented to evaluate the STMA. Simulation results indicate that 60% speed improvement is achieved by using the STMA in the H.264/AVC deblocking filter.

The paper is organized as follows. The proposed structure for self-transposing memory is presented in Section II. The architecture is analyzed in Section III. Section IV presents experimental results and comparison. Finally, the paper finishes by the conclusion in Section V.

II. THE PROPOSED MEMORY STRUCTURE FOR 8×8 SELF-TRANSPOSING MEMORY

In many block-based video and image coding standards, images are divided to 8×8 data blocks to perform an efficient compression [5]-[7]. 2-D processing tools are applied to 8×8 data blocks, as well. An 8×8 data block, called M, is composed of sixty four 8-bit data entries, known as pixels. The pixels are the smallest accessible data elements in video and image applications. High-performance processors and accelerators, however, access to the data of sizes greater than a pixel. For example, 32-bit data bus is very frequently used in existing system-on-chips [8].

Typically, in a 32-bit video/image processing system, pixels are maintained in the memory in the form of quadruples of neighboring pixels (Q_i). Therefore, the memory map of the block M is as illustrated in Figure 1. The STMA is proposed based on this memory mapping. Its detailed structure is shown in Figure 2. The STMA has two ports for simultaneous reading