



# 4th DPVSA

IEEE Seasonal School on Digital Processing of Visual Signals and Applications  
Virtual, October 19 - 21, 2020

Universidade Federal de Pelotas

# SIMD IMPLEMENTATION OF MOTION COMPENSATION FOR PROCESSING-IN-MEMORY EXPLOITATION IN VIDEO DECODERS

Garrenlus de Souza<sup>1</sup>, Marco Antonio Zanata Alves<sup>2</sup>,  
Bruno Zatt<sup>3</sup>, Sergio Bampi<sup>1</sup>, Felipe Sampaio<sup>4</sup>

gsouza@inf.ufrgs.br

<sup>1</sup>Federal University of Rio Grande do Sul (UFRGS), Brazil

<sup>2</sup>Federal University of Paraná (UFPR), Brazil

<sup>3</sup>Federal University of Pelotas (UFPeL), Brazil

<sup>4</sup>Federal Institute of Rio Grande do Sul (IFRS), Brazil

## INTRODUCTION

- **Motion compensation (MC) for HEVC video decoding**
  - Goal → reconstruct (at the encoder side) the blocks predicted using interframe modes (Fig. 1);
  - Support of bi-prediction and half- and quarter-pixel interpolations;
  - Memory aspects → poor temporal locality [1].
- **Processing-in memory (PIM)**
  - Key idea → move computations to near the data array blocks (aka. near-memory computing);
  - Overcomes cache hierarchy inefficiency in case of: poor data locality (mainly temporal), high data traffic and intensive computing.
- **This work**
  - Goal → to exploit a PIM-based hardware to improve motion compensation performance and energy efficiency in video decoding;
  - Main contribution → SIMD implementation of MC for PIM exploitation onto VIMA architecture.

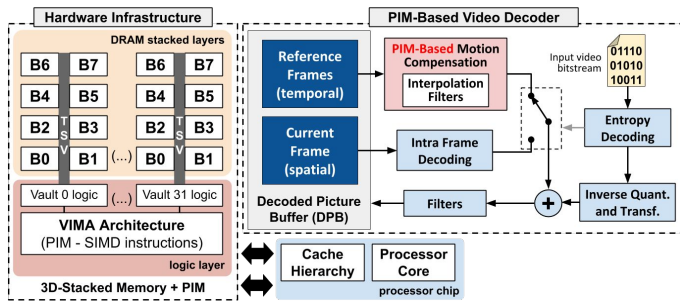


Fig. 1 PIM hardware infrastructure and video decoder diagram.

## SIMD IMPLEMENTATION

- **Interpolation filters**
  - Critical operation for fractional motion vectors;
  - Half- and quarter-pixel precisions.
- **Adopted Simulation methodology**
  - VIMA intrinsics library (C language);
  - OrCS cycle-accurate simulation environment.
- **PIM-Based implementation strategy**
  - Exploit entire data segments → 256B-8KB accessed in parallel thanks to through-silicon vias (TSV) at 3D-Stacked DRAM organization;
  - Interpolation filters decomposed into bulk operations, like multiplication and sum;
  - Vertical and horizontal interpolation achieved by breaking the filter calculations into several arrays, one for each of the weights.

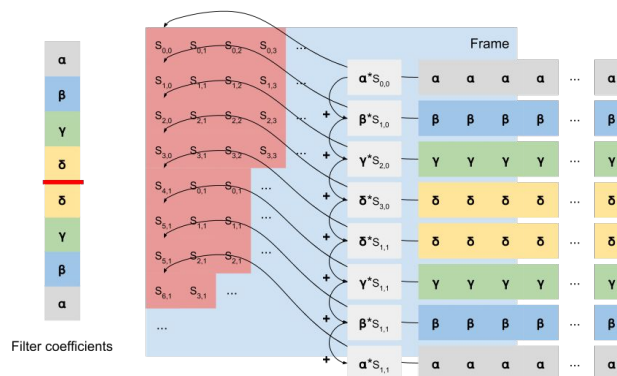


Fig. 2 Vertical Interpolation with an 8-tap filter.

## Filters implementation

- **Vertically** → The sum of products is achieved by the paired aggregation of the temporary array that holds the products by the filter weights (Fig. 2).
- **Horizontally** → The sum of products is achieved by the lateral offset and shifted sum (Fig. 3)

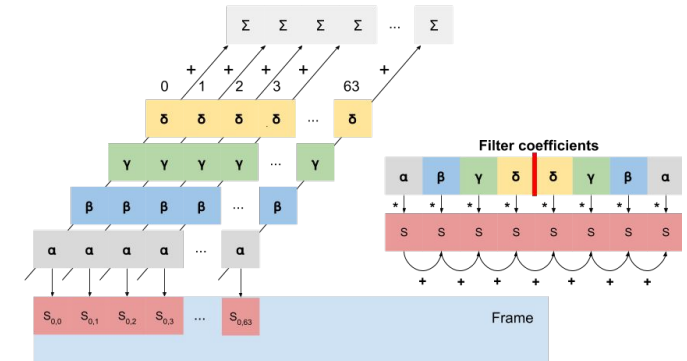


Fig. 3 Horizontal Interpolation with an 8-tap filter.

## FINAL CONSIDERATIONS

- Research status → execution for first set of experiments for preliminary evaluations (ongoing);
- Further analysis will be made in order to exploit the pitfalls and advantages of such an approach;
- Future works: evaluation of other kernels that can take advantage of PIM, such as low power, faster memory access and I/O latency reduction.

[1] G. Souza, A. Cerveira, B. Zatt, S.Bampi and F. Sampaio, "Evaluation of Cache-Based Memory Hierarchy for HEVC Video Decoding" in IEEE 33rd Symposium on Integrated Circuits and Systems Design (SBCCI), 2020.