IEEE International Symposium on Field-Programmable Custom Computing Machines

Implementation and Performance Comparison of the Motion Compensation Kernel of the AVS Video Decoder on FPGA, GPU and Multicore Processors

M. Owaida*, N.Bellas*, C. D. Antonopoulos*, K. Daloukas*, Ch. Antoniadis*, K. Krommydas† and G. Tsoumblekas‡
*Dept. of Comp. & Comm. Engineering, University of Thessaly, Volos, Greece
{mowaida, nbellas, cda, kodalouk, haadonia}@inf.uth.gr
†Dept. of Computer Science, Virginia Tech, VA, USA.
kokrommy@vt.edu
‡Dept. of Informatics, National and Kapodistrian University of Athens, Greece
grad1134@di.uoa.gr

***Abstract*—Next generation video standards have strict and increasing performance demands due to real-time requirements and the trend towards higher frame resolutions and bitrates. Leveraging the advantages of reconfigurable logic and emerging multi-core processor architectures to exploit all levels of parallelism of such workloads is necessary to achieve real time functionality at a reasonable cost.**

***Keywords*-Video Compression, AVS Motion Compensation, FPGA, Reconfigurable Computing, Multi-cores, CellBE, GPU**

I. Introduction

This paper is the first work to explore, map and optimize the Motion Compensation module of the AVS decoder on an FPGA. It is also the first work to combine a complete repertoire of software optimizations in order to achieve real time FullHD decoding on Intel's Core i7 processor, the Cell processor and an NVidia GTX480 GPU.

II. AVS Motion Compensation

A. Algorithm

Video decoding of each frame is performed at the granularity of a macroblock (MB), i.e. a 16x16 pixels area of the frame. Only a subset of the MBs in the frame, called inter-predicted MBs, are predicted using the Motion Compensation algorithm. AVS supports four sizes of sub-blocks (8x8, 8x16, 16x8 and 16x16) for inter prediction.

B. Performance and Parallelism Potential

The AVS Motion Compensation is characterized by high data-level parallelism. All Luma and Chroma sub-pixel values for one MB can be computed in parallel. Moreover, there is a high degree of data- and computation-reuse, which can be exploited to eliminate redundant memory accesses and computations.

III. FPGA implementation

The AVS Motion Compensation module is depicted in Figure 1.

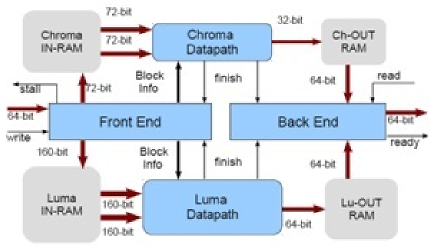

Figure 1. FPGA AVS Motion Compensation Module.

IV. Multicore and GPU Implementations

A. Multicore Implementation

Optimization and mapping of the code to the target multicore platforms is done in two steps: a) Exploitation of thread-level parallelism, and b) Vectorization to exploit the SIMD capabilities of the platforms.

B. GPU Implementation

Parallelism in the GPU is exploitable at a different granularity than in the other three platforms. GPUs perform better when each thread processes a single pixel (fine-grain parallelism), whereas the remaining platforms need coarser-grain parallelism (at the MB level) to minimize parallelism management overheads.

V. Conclusion

During the mapping and optimization steps, we observed that thread-level parallelism is the prevalent source of performance improvement for MC processing and video decoding in general, as technology moves towards platforms with hundreds of cores. This is due to the plentiful MB-level parallelism within and even across video frames. However the exploitation of fine-grained parallelism is also an integral part of the optimization of video applications, especially in hardware implementations.

Acknowledgment

This work is partially supported by the EC Marie Curie International Reintegration Grant (IRG) 223819.