A High-Performance Hardware Accelerator for HEVC Motion Compensation

Matthias Göbel
Embedded Systems Architecture Group
Dept. of Computer Engineering and Microelectronics
Technische Universität Berlin

m.goebel@tu-berlin.de

Abstract: The presented master's thesis has focused on the design and implementation of a motion compensation hardware accelerator for use in HEVC hybrid decoders, i.e. decoders that contain hardware as well as software parts. As the motion compensation is the most time consuming step in the decoding process it is crucial to implement it in a fast and efficient way. This paper elaborates the theoretical background and motivation and highlights the main design choices. In the following evaluation a comparison between the hybrid decoder and a pure software decoder is performed. The results show that the design is capable of increasing the decoding frame rate in the range of 60% for 1080p video streams when running at 100 MHz.

1 Introduction

High Efficiency Video Coding (HEVC) [1] is the latest video coding standard by the *Joint Collaborative Team on Video Coding* (JCT-VC) and has been ratified as H.265 in April 2013. It is the direct successor to the famous H.264/*Advanced Video Coding* (AVC) standard and reduces the bit rate by 50% for the same video quality when compared to H.264. For cost and power reasons it is common practice in video decoding to use dedicated hardware accelerators. Dedicated hardware blocks can perform the most expensive parts of the decoding process in a fast and efficient way thereby offering more performance while consuming less power than a pure CPU-based solution. This paper in particular focuses on designing and implementing a hardware accelerator for motion compensation, i.e. an interpolation filter that should substitute the according part in an existing software decoder as it is the most time-consuming part of software decoders.
Similar to its predecessors HEVC allows to exploit temporal and spatial redundancy in video streams by referring to similar regions in previous frames instead of storing all the data explicitly. This technique that is known as *inter-frame prediction* is implemented in HEVC by using so called *motion vectors* that point to such regions in previously decoded frames. These motion vectors can also have a horizontal or vertical shift relative to the target region with an accuracy of 1/4th of a sample for the luma plane and 1/8th of a sample for the chroma planes. In order to successfully decode an inter-frame predicted HEVC video stream these *fractional samples* must be derived from the adjacent full samples by

using an interpolation filter. This process is called *motion compensation* and has been the main task of the discussed master's thesis.
This paper is organized as follows. Section 2 lists related work that focuses on hardware solutions for motion compensation in general as well as for HEVC in particular. In Section 3 the design process is highlighted followed by a discussion of the evaluation in Section 4. Finally, in Section 5 a conclusion is given regarding the results of the thesis.

2 Related Work

As HEVC has only been standardized in April 2013 the amount of related work in general and regarding motion compensation in particular has been very limited. Guo et al. [2] deal with the motion compensation interpolation and propose a resource-efficient ASIC implementation for the FIR interpolation filter as well as an efficient filter engine that is based on splitting a frame into blocks of 4x4 luma samples. An HEVC video-decoder chip for 4K applications has been presented by Tikekar et al. [3]. This chip is capable of processing 249 MPixel/s which is sufficient for real-time decoding of 4K video streams with 30 FPS. However, a huge amount of related work for AVC motion compensation has been available. An efficient memory access solution is discussed by Tsai et al. [4]. By reusing previous pixels via a cache they can decode a 2048x1024 video stream running at 30 FPS in real-time with less than 200 MB/s of memory bandwidth.
While these approaches focused mostly on pure hardware implementations, this work follows a hardware/software codesign approach. By partitioning the task accordingly the advantages of software and hardware can be combined thus getting a maximum of performance.

3 Design

For the design decision several approaches have been analyzed. While the parallel processing of multiple samples has theoretical advantages regarding the throughput, such solutions tend to occupy many logic resources. Furthermore, the memory will probably be a bottleneck for them. Therefore a solution that is capable of filtering one sample per cycle has been chosen with parallel processing of luma and chroma planes.
The final design consists of two similar independent datapaths: one for luma as well as one for chroma. An overview that is valid for both datapaths can be seen in Figure 1. As the interpolation process involves a two-dimensional FIR filter a two-step procedure has been selected that performs first a one-dimensional horizontal interpolation filtering and afterwards another one-dimensional vertical one. Between these steps a buffer is implemented that stores the results of the first filter before they can be processed by the second filter. This is required as almost the complete horizontal filter process must have finished before the vertical filter process can start. As a result the theoretical throughput is reduced to 0.5 samples per cycle. For each luma and chroma two reference blocks can be processed

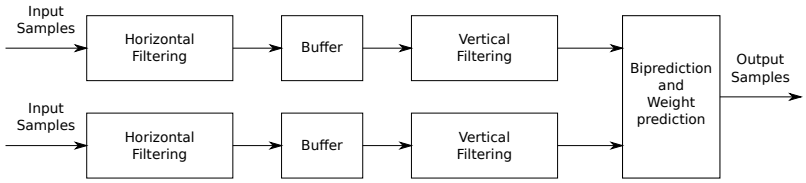

Figure 1: One of the two independent datapaths. Each of them again consists of two sub-datapaths that are required for biprediction. Note that one chroma datapath is sufficient for a subsampling ratio of 4:2:0.

in parallel to support *biprediction*, i.e. interpolating a region in a frame by using two different regions as a reference or input. If biprediction has been selected the results of the two vertical filters will be averaged; otherwise only the result of the first vertical filter will be used. Finally, the result can also be weighted, i.e. be multiplied with a certain factor. This feature is called *weighted prediction* and is implemented by an additional multiplier at the end of each sub-datapath.
For the level of granularity, i.e. the partitioning of the overall work into software and hardware parts of the decoder, the *prediction unit* (PU) has been selected. This is a rectangular block of between 8x4/4x8 and 64x64 luma samples and the according numbers of chroma samples that has a fixed set of parameters. This choice allows to perform most of the complex tasks like parameter evaluation in software while offering the advantage of massive parallelism that hardware solutions provide for the actual interpolation process.

4 Evaluation

The discussed design has been implemented for the Zynq-7020 SoC from Xilinx. A theoretical analysis of the accelerator itself (i.e. only of the motion compensation) yielded an upper bound for the throughput of 50.5 FPS for 1080p video streams when running at 100 MHz. For the software part of the hybrid decoder a scalar software decoder developed at TU Berlin has been modified to use the hardware accelerator for the interpolation process. The interface between hardware and software parts is implemented using a register-based solution in which the CPU handles all the memory access. As the memory overhead is expected to be high, an additional DMA-based interface has been implemented as well to be able to derive the speed-up of such a solution. To be able to compare all three implementations (pure software, register-based implementation, DMA-based implementation) the Kimono video stream of the JCT-VC test sequences [5] has been used in different 1080p encodings. The results when using a frequency of 100 MHz can be seen in Figure 2.
While the frame rate for the register-based interface is reduced significantly by the huge memory overhead, the DMA-based interface is capable of delivering a significant speed-up of about 60% compared to the pure software decoder. However, the memory access still poses the main bottleneck. Figure 2 also shows the luma throughput of the accelerator for PUs of different sizes. For large PUs it converges to the theoretical maximum of 0.5 samples per cycle as the interpolation overhead is decreasing.

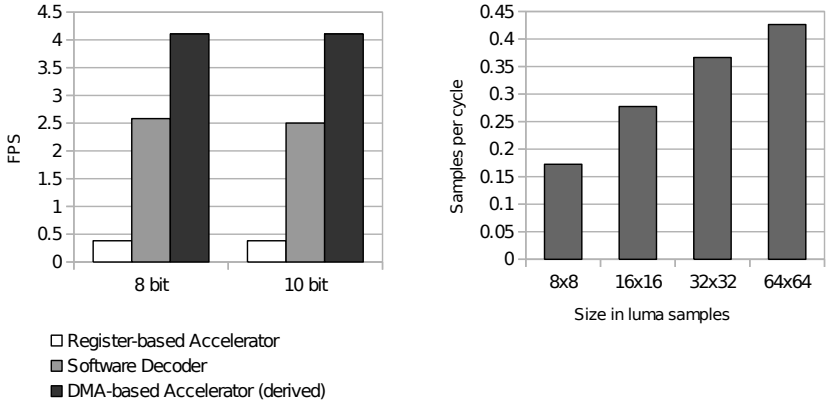

Figure 2: An evaluation of the accelerator. The left diagram shows the achieved frame rates using the evaluation setup. On the right side the luma throughput for PUs of different sizes can be seen.

5 Conclusion

This paper described the design of a hardware-accelerator for HEVC motion compensation. Based on the idea of a hybrid decoder such an accelerator has been implemented. The evaluation proved the feasibility and reasonability of the design as it offers a speed-up of about 60% compared to a pure software solution. Based on the results of this thesis additional work is currently in progress. In particular, further optimizations regarding the memory access will be performed as this turned out to be the major limiting factor in the implementation.

References

[1] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand. Overview of the High Efficiency Video Coding (HEVC) Standard. IEEE Transactions on Circuits and System for Video Technology, Volume 22, No. 12:1649-1668, 2012.

[2] Z. Guo, D. Zhou, and S. Goto. An Optimized MC Interpolation Architecture for HEVC. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2012.

[3] M. Tikekar, C.-T. Huang, C. Juvekar, V. Sze, and A.P. Chandrakasan. A 249-Mpixel/s HEVC Video-Decoder Chip for 4K Ultra-HD Applications. IEEE Journal of Solid-State Circuits, Volume 49, Issue: 1, 2014.

[4] C.-Y. Tsai, T.-C. Chen, T.-W. Chen, and L.-G. Chen. Bandwidth Optimized Motion Compensation Hardware Design for H.264/AVC HDTV Decoder. 48th Midwest Symposium on Circuits and Systems, 2005.

[5] F. Bossen. Common test conditions and software reference configurations. ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC) Document JCTVC-K1100, 2012.