Performance Requirements for Reconfigurable Hardware for a Scalable Video Decoder

Harald Devos and Hendrik Eeckhaut

Abstract— **Nowadays multimedia applications emerge on portable devices everywhere. These applications typically require a high amount of computational power together with real-time performance. Another requirement is that they must be reconfigurable i.e. that the application or the characteristics of the application can be modified at will. The performance requirement often drives the design towards a hardware implementation while the reconfigurability requirements are better served by a software implementation. We try to reconcile these two requirements by exploring a third implementation option: FPGAs. By using an FPGA we can achieve both the performance and the flexibility that is needed for multimedia applications. In the RESUME project we are designing a scalable wavelet video decoder as a proof of concept. In this article we explore through analytical means the question of whether an FPGA implementation can be used to reach the performance goals. We find that modern FPGAs offer enough computational power to obtain real-time performance, but that reaching the necessary memory bandwidth will be a challenge.**

Keywords—**Multimedia, reconfigurable hardware, scalable wavelet video, HW resource estimation**

I. Introduction

When designing a multimedia system one would prefer to implement most of the functionality in software running on a general purpose processor since this improves the flexibility and avoids the high cost of designing an ASIC (Application Specific Integrated Circuit). Still, often the performance requirements of the system are such that specific hardware is a necessity.

An important disadvantage of ASICs is that their functionality is cast in stone. For example an ASIC designed for doing JPEG compression utilizing Huffman encoding becomes useless when application changes require arithmetic encoding. Accommodation for all the required operations of the system, that may only be needed infrequently, results in an overdimensioning of the ASIC which increases the complexity and the production cost.

In the last decade a new class of hardware devices has emerged which holds the middle between ASICs and general purpose processors: FPGAs (Field Programmable Gate Arrays) [1]. These devices contain a large amount of gates whose functionality and interconnection can be reprogrammed. This results in a performance which leans towards an ASIC with some of the flexibility of a software implementation.

In the RESUME project (Reconfigurable Embedded Systems for Use in scalable Multimedia Environments [2]) we explore the usefulness of FPGAs for multimedia systems by building a *scalable* wavelet based video decoder [3], [4]. "Scalable" means that it allows to change the quality of service (QoS) i.e. the framerate, resolution, color depth, ... of the video without changing the encoded video stream used by the decoder, except for skipping unnecessary blocks of data without decoding.

This paper gives an overview of the decoder (II), points out its scalability features (III), presents the profilation of the software version of the decoder (IV) and some preliminary performance estimates (V). It further describes the future evolution of our project (VI) and ends with some conclusions (VII).

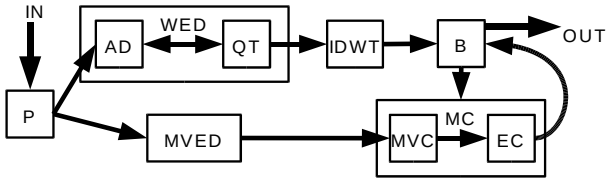

Fig. 1. High-level overview of the video decoder.

II. System Overview

Figure 1 shows a high-level overview of the wavelet video-decoder. It consists of the following parts:

P: The "parser" receives a bit stream representing a compressed video. It analyzes the structure of this stream and is capable of extracting relevant parts and dropping other parts that are not required for the further decoding process.

WED: The "wavelet entropy decoder" decodes entropy encoded parts of the bit stream into wavelet transformed frames. It consists of two strongly interconnected parts: the "arithmetic decoder" (AD) and the "quadtree decoder" (QT) [4].

IDWT: The "inverse wavelet transform" takes a wavelet transformed frame and produces either a reference image (the first image of a group of pictures) or an error frame.

MVED: "Motion vector entropy decoding" performs the entropy decoding of the motion vectors.

MC: "Motion compensation" reconstructs the final video frames. The "motion vector compensation" (MVC) constructs a first estimate of the resulting video frame by translating blocks of 2 reference frames along motion vectors. Fractional translation of blocks is obtained through interpolation of the original pixels. The "error correction" (EC) adds the corresponding error frame to this estimate in order to produce the final image.

B: The "reorder buffer" puts the decoded frames in the correct order.

III. Exploiting the Scalability

The scalability of the QoS of the decoder includes the framerate, the resolution, the accuracy (cfr. PSNR - Peak Signal to Noise Ratio), the availability of colors, ... The scalability of the hardware cost contains parameters like used chip area, power consumption, used network bandwidth, ... It is clear that these are not independent of each other. We will focus here on some parameters that describe the scaling of the QoS of the video and describe their influence on the calculation cost.

Frame rate: Different frame rates are possible through a dyadic temporal decomposition [5] of the video stream (Fig. 2). The decoder can choose to which (temporal) level the stream is decoded. Each extra level doubles the frame rate.

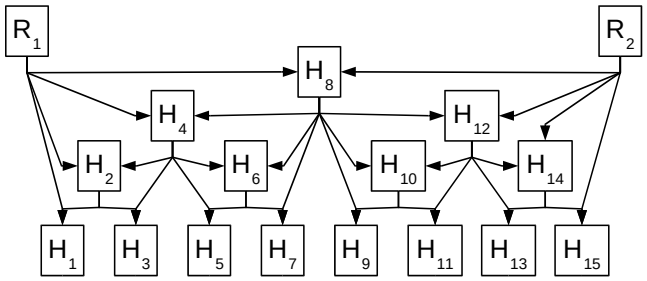

Fig. 2. The temporal decomposition of one GOP (Group Of Pictures). The arrows illustrate which frames are used to reconstruct (via motion compensation) other frames at lower decomposition levels. R_1 and R_2 are reference frames, the H_i are reconstructed (intermediate) frames.

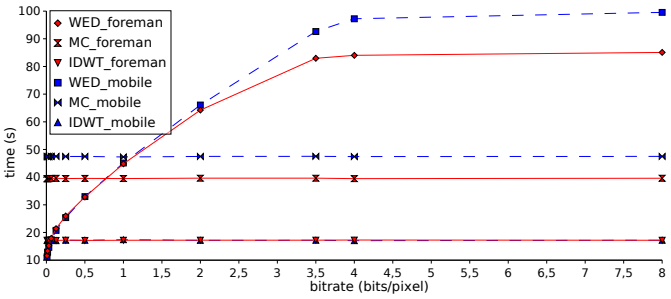

Fig. 3. The execution time for Wavelet Entropy Decoding (WED), Inverse Discrete Wavelet Transform (IDWT) and Motion Compensation (MC) for both the 'foreman' and the 'mobile' sequence.

Resolution: In the encoder a 2-D discrete wavelet transform (DWT) in K levels is performed on the error and reference frames. The inverse discrete wavelet transform (IDWT) in the decoder can stop at an arbitrary level. The resolution and computation cost are proportional to the number of performed levels.

Image quality: The QT decodes a bit stream containing the wavelet entropy encoded frames. Resources can be saved by skipping some less significant bit planes, with a lower quality (measured by e.g. the PSNR) of the decoded frames as a result.

IV. A rough profile of the decoder

In this section we try to obtain a rough estimate of where the computationally expensive parts of the decoding system are located. Therefore we profiled the decoding algorithm based on a preliminary C/C++ implementation (on a standard PC[1]).

In Figure 3 we plotted the execution time of the three major time consuming blocks of the algorithm versus the bitrate. Only the WED depends on the bitrate. For higher bitrates this is the most time consuming block of the decoding algorithm. The MC is independent of the framerate but varies for different sequences. It varies from frame to frame dependent on the actual values of the motion vectors. The share of the IDWT is rather limited and independent of both bitrate and sequence; each frame needs exactly the same number of computations. The 'mobile' sequence is apparently harder to decode.

It takes now about 140 seconds (without IO) to decode a CIF sequence of 288 frames. To achieve real time decoding, i.e. 30 frames per second, we have to perform the decoding algorithm some 15 to 20 times faster. This will not be possible on a portable device with solely a more optimal software implementation.

[1] Pentium IV 2.0 GHz, 512MiB RAM

V. Estimations

To be able to choose a hardware platform, the requirements (e.g. needed memory, bandwidth, computing power) of the application have to be known. The exact quantities are only known after implementation. Therefore estimations have to be made. Due to the limited size of this paper we will only describe the major conclusions: the necessary internal bandwidth will be the most restrictive bottleneck; the WED has a high compression performance but its memory access pattern and sequential nature hampers an elegant hardware implementation; the memory requirements are quite modest; and modern FPGAs offer enough calculation power to implement the video codec.

VI. Future Work

It will be possible to configure the FPGA with hundreds of implementations of the video decoder. To find the optimum configuration, techniques are needed to evaluate the performance and cost of a configuration without an actual instantiation.

Infrastructure is required to reconfigure FPGAs on-the-fly. It will need to stop the running task, transfer its state and restore this state once the FPGA is reconfigured.

VII. Conclusions

There is an ever increasing demand for scalable multimedia applications and software may not be able to cope with the high performance requirements. Reconfigurable hardware could be a solution. It combines flexibility with high performance.

VIII. Acknowledgments

Harald Devos is supported by the fund for scientific research Flanders (F.W.O.). Hendrik Eeckhaut by I.W.T. grant 020174.

References

[1] A. DeHon, "The density advantage of configurable computing," *IEEE Computer*, vol. 33, no. 4, pp. 41–49, April 2000.

[2] "The RESUME project: Reconfigurable Embedded Systems for Use in scalable Multimedia Environments," http://www.elis.UGent.be/resume.

[3] A. Munteanu, J. Cornelis, G. Van der Auwera, and P. Cristea, "Wavelet-based lossless compression scheme with progressive transmission capability," *International Journal of Imaging Systems and Technology, Special Issue on "Image and Video Coding"*, vol. 10, no. 1, pp. 76–85, January 1999.

[4] P. Schelkens, A. Munteanu, J. Barbarien, M. Galca, X. Giro-Nieto, and J. Cornelis, "Wavelet coding of volumetric medical datasets," *IEEE Transactions on Medical Imaging, Special Issue on "Wavelets in Medical Imaging"*, vol. 22, no. 3, pp. 441–458, March 2003.

[5] D. Turaga and M. van der Schaar, "Unconstrained motion compensated temporal filtering," in *ISO/IEC JTC1/SC29/WG11, m8388, MPEG 60th meeting*, May 2002.

[6] D. Stroobandt, *A Priori Wire Length Estimates for Digital Design*, Kluwer Academic Publishers, Boston / Dordrecht / London, April 2001.