Memory Management Proposals for a Video Encoder

Monica Magalhães, Milano Gadelha, André Brito, Ivan Saraiva Silva
DIMAp, UFRN
(monica, milano, brito)@natalnet.br, ivan@dimap.ufrn.br

Abstract

One of the main difficulties concerning the development of video encoders is the implementation of the memory management. This paper presents proposals for a video encoder's memory management development, their restrictions and some techniques for data access.

1. Introduction

An encoder/decoder module (CODEC) performs the coding of an image or video sequence and decodes it producing an exact copy or an approximate copy of the original sequence.

One of the main bottlenecks commonly found in the development of CODECs is related to memory management. The availability of memory is limited by: data flow, bandwidth, and memory latency. These problems are increased by the speed and the number of accesses to the memory, as is the case for real-time video compression.

Finding out the proper memory management techniques for video compression is not a trivial task. It is necessary to consider many requirements related both to the adopted video coding standards and the amount of memory resources.

This paper presents proposals for the memory management used in a video encoder. Restrictions on memory utilization are discussed, and some data access techniques are presented.

This paper is organized in the following manner: the next section shows the encoding process. Then it will be proposed some memory management alternatives. The fourth section applies some of the proposals in the implementation of a motion estimation algorithm and in the last section the final considerations are made.

2. Video Coding

The two main advantages of video compression are: the possibility of digital video transmission and a more efficient usage of storage and transmission resources.

The video encoding process is done in three steps. The first step is responsible for the reduction of the temporal redundancy. The second step consists in reducing spatial redundancy. The third step is the entropic codification of the results obtained in the preceding steps [1].

The temporal stage performs a process called motion estimation. The motion estimation consists on the calculation of the motion vectors that indicates the block or the shifted region.

The motion estimation algorithms [2], besides computationally expensive, define how many accesses to the memory are made. Because of these facts, the study about the memory management was based on the motion estimation which is a critical part of video encoding.

3. Memory management proposals for a video encoder

The search for solutions to optimize the memory usage comprises many aspects: the adopted algorithms, communication subsystems (bus), memory hierarchy, data format, etc.

In a CODEC the access rate is influenced mainly by the need of multiple accesses to the image data and by the generation of intermediate results, as well as the necessity of accessing them. To reduce the number of accesses made, a solution might be the use of a memory hierarchy where each component of the micro-architecture contains a local memory. For example, an internal module of the motion estimator will have local memory capable of storing a macroblock of the current frame and a search area of the reference frame. The access to the macroblock's pixels is done through a local memory that should be accessed in a single clock cycle.

While moving data from the external memory to the local memory, it may be advantageous to change the sequence of this data in order to achieve a more appropriate way for the datapath part access. Taking again the motion estimation as an example, frame's pixels stored in the external memory are composed by three components, from which only the luminance component is necessary for motion estimation. Therefore the access to a portion of the image needs to return only the data that is comprised by the luminance component.

It is also possible to separate the memory in banks and use different kinds of buses to connect them. When the addresses are stored in different memory banks it is said that the memory is interleaved [3]. Its usage allows the simultaneous access to the data and reduces the utilization of the bus by the different encoder's modules. As the motion estimation algorithm requires access to data of at least two frames, it is emphasized the benefits brought by storing these frames in different memory banks.

In a complex system such as a CODEC for digital video is important to decide how the data will be managed before choosing the definitive memory modules (concerning the technology, quantity, capacity and disposal, as well as forms of data access and indexation).

There are two approaches that stand with more evidence. The first one joins the software flexibility with the hardware power. The memory management is done through specific routines implemented in software to be executed by a microprocessor. The microprocessor in this case is a data server that feeds the local memory on demand.

In the second approach, the memory management is done by specific components of hardware. On this method, the system gains from the hardware performance. However, it loses flexibility and domain of executable code generation in relation to the generation of synthesizable code in HDL (Hardware Description Language).

The first solution enables the use of hardware-software co-design techniques. The usage of co-design accelerates the development process; however longer validation steps are needed in order to validate the partitioning and synthesis of both hardware and software.

4. Case study

The study about the memory management is being applied in the development of a component that performs the motion estimation and in the implementation of the encoder's memory management.

From this study an architecture for a component that performs the motion estimation of a video encoder based on the ITU-T H.264 recommendation [4] was proposed. This component is part of a H.264 video encoder of the Sistema Brasileiro de Televisão Digital (SBTVD) project [5]. It uses local memories that store global data available for all components of the encoder.

The component that is responsible for the calculation of the motion vectors contains two memories. One memory stores the macroblock of the current frame while the other stores the search area of the reference frame. Figure 1 shows this component.

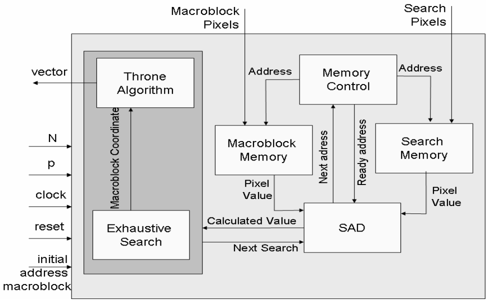

Figure 1. Motion vectors calculation

According to the figure, the search memory stores the search area from the reference frame, while the macroblock memory contains the macroblock from the current frame. The component responsible for vector calculations abstracts how the search area is found, who is responsible for locating this area and which are the reference frames, as well as who has the current frame and the macroblock's origin.

5. Final Considerations

One of the major issues in developing architectures for an encoder/decoder is related to the memory requirements.

This paper presented a study about the difficulties concerning the usage of memory and some solutions to allow a better efficiency in data access.

The requirements of the other encoder's modules are being studied, in order to elaborate an external memory that satisfies all the components of the architecture and its memory management.

6. References

[1] Richardson, I. E. G., *H.264 and MPEG-4 Video Compression*, Wiley Publishing, England, 2003.

[2] Roma, N. F. V., *Processadores Dedicados para Estimação de Movimento em sequências de vídeo*, PhD Thesis, Universidade Técnica de Lisboa, Lisboa, 2001.

[3] Hwang, K., *Advanced Computer Architectures: parallelism, scalability, programmability,* Mac Graw-Hill Publishing, 1993.

[4] International Telecomunication Union, *AudioVisual and Multimedia Systems Infraestructure of audiovisual services – Coding of moving video*, Geneva, 2005.

[5] Sistema Brasileiro de Televisão Digital,
http://sbtvd.cpqd.com.br/, June.