Specialized Motion Estimation Processor for Heterogeneous Multicore Video Coding Systems

Nuno Sebastião[†], Tiago Dias[†,§], Nuno Roma[†,‡], Paulo Flores[†,‡], Leonel Sousa[†,‡]

[†] *INESC-ID, Rua Alves Redol, 9, 1000-029 Lisboa, Portugal*
[‡] *Instituto Superior Técnico, TU Lisbon, Av. Rovisco Pais, 1049-001 Lisboa, Portugal*
[§] *Instituto Superior de Engenharia de Lisboa, PI Lisbon, Rua Conselheiro Emídio Navarro, 1, 1959-007 Lisboa, Portugal*

ABSTRACT

In the last few years, several highly efficient processor architectures (e.g.: ARM, Power-PC, etc.) have been proposed to implement computational demanding multimedia applications. This paper proposes to use these capabilities to extend the main processor computational resources, in order to efficiently execute one of the most critical parts of the video coding algorithm: motion estimation. This extension is accomplished by embedding a specialized Application Specific Instruction Set Processor (ASIP) core for motion estimation that works together with a General Purpose Processor in a heterogeneous multicore environment.

KEYWORDS: Motion Estimation; Multicore; Video Encoding

1 Introduction

In the last few years, with the increasing use of digital video acquisition devices, there has been a growing demand for real-time video encoding systems. Some of these encoding systems are often used in mobile devices (e.g.: digital cameras, cellular phones, etc.), which usually present strict power consumption restrictions. Several tasks have to be executed to perform video encoding, such as quantization, entropic coding, motion estimation and others. All of the these video encoding tasks are usually implemented in a General Purpose Processor (GPP). However, this type of implementation is usually inefficient in terms

of power consumption and is often inappropriate for embedded systems, especially those with strict power constraints, like portable battery-supplied devices.

Among the several tasks involved in video encoding, Motion Estimation (ME) is the most computationally expensive one, representing between 50% and 70% of the total number of required operations for video encoding. Therefore, according to Amdahl's Law, an eventual improvement in the implementation of this task should potentially provide a significant global speed-up of the entire system. Hence, a multicore video encoding system, with specialized processing units, presents an adequate solution that effectively reduces the power consumption and maintains the system performance and flexibility. In this approach, the ME task is performed by an efficient specialized processing core, while the remaining tasks are performed by a GPP core, such as the Power-PC.

This paper is organized as follows. In Section 2, the architecture of the specialized processor core for ME is described. Section 3 presents the implementation environment of the video encoding system in a heterogeneous multicore system, as well as the obtained experimental results. The conclusions and future work are presented in Section 4.

2 Motion Estimation Core Architecture

The programmable and specialized architecture for ME proposed in [DMRS07] was tailored to efficiently program and implement a broad class of powerful, fast and/or adaptive ME search algorithms. This architecture supports the most used Macroblock (MB) structures, such as the traditional fixed 16×16 pixels block size, adopted in the H.261/H.263 and in the MPEG-1/MPEG-2 coding standards, or even any other variable block-size structures, adopted in the H.264/AVC video standard.

The Instruction Set Architecture (ISA) of this ASIP was designed to meet the requirements of most ME algorithms, including some recent approaches that adopt irregular search patterns, such as the data-adaptive ones. Such ISA is based on a register-register architecture and provides a quite reduced number of different instructions (eight), that focus on the set of operations that are most widely used in ME algorithms:

- `J` - *control operation*, to change the execution-flow of a program, by updating the program counter with an immediate value;
- `MOVR` - *data transfer operation*, to store the content of a given register in another target register;
- `MOVC` - *data transfer operation*, to store an 8-bit constant immediate value in the upper or lower byte of a target register;
- `SAD16` - *graphic operation*, to compute the Sum of Absolute Differences (SAD) value corresponding to two sets of 16 pixels and to accumulate the result in a target register;
- `ADD` - *arithmetic operation*, to add the contents of two registers;
- `SUB` - *arithmetic operation*, to subtract the contents of two registers;
- `DIV2` - *arithmetic operation*, to evaluate the integer division by 2 of the contents of a given register;
- `LD` - *memory data transfer operation*, to load the pixels data into two fast and small scratch-pad local memories.

These instructions directly operate the values stored in a register file composed by 32 registers, capable of storing one 16-bit word each.

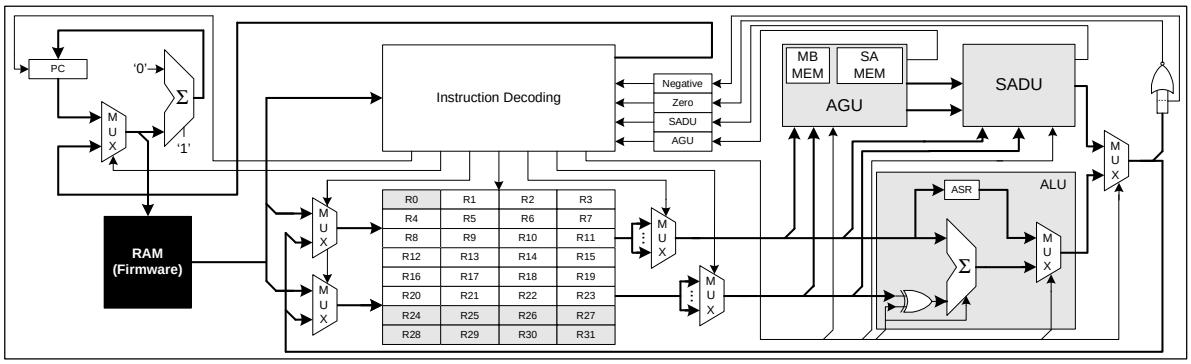

Figure 1: Architecture of the ME core.

The processor datapath, depicted in Fig. 1, includes two specialized units to increase the efficiency of the most complex and specific operations: an Address Generation Unit (AGU) and a SAD Unit (SADU). Hence, the `LD` operation is efficiently executed by the dedicated AGU, which is capable of fetching all the pixels of a reference MB or of a search area, and store them in the corresponding scratch-pad local memories (located in the AGU). On the other hand, the `SAD16` graphic instruction is implemented by the SADU. In the architecture that is now being considered, the implemented SADU adopts a serial structure that restricts the required hardware, at the cost of requiring more clock cycles to compute this operation. Nevertheless, a complete parallel implementation of the SADU could equally be considered, leading to a faster execution time, with an added cost of requiring more hardware resources. Further and more detailed information about this processor architecture can be found in [DMRS07]

3 Prototyping and Experimental Results

A complete video encoding system was developed and implemented, in order to validate the functionality of the presented core. The base configuration of the encoding system consists of a GPP that executes all the video encoder operations, except for the ones concerning ME, which are executed by the specialized ASIP core. This processor core computes, in parallel with the other operations, the several motion vectors that are required by the encoder to implement the temporal prediction mechanism.

The performance analysis of the prototyped motion estimation ASIP considered a simplified AGU that does not allow data re-usage and a power efficient serial processing structure for the SADU module. The implementation of the whole video encoding system was realized using a Xilinx ML310 development platform, which includes a 100MHz 256MB DDR memory bank and a Virtex-II Pro XC2VP30 FPGA device from Xilinx. This FPGA offers two Power-PC processors, several Block RAM modules and high speed on-chip bus-communication links. The specialized processor core for ME was implemented using the configurable logic blocks provided by this FPGA, while the main processing unit of the video encoder consists of a Power-PC 405 D5 processor, operating at 300MHz, running an optimized software implementation of the H.264 video encoder (JM H.264/AVC Reference Software - version 13.2). This encoding system was implemented using the EDK 9.1i and ISE 9.1i tools from Xilinx.

Table 1 presents the implementation results obtained for the ME ASIP core. These results evidence that FPGA based implementations of the considered ME architecture allow a maximum operating frequency of about 100 MHz. They also show that very few hardware resources (about 6k equivalent logic gates) are required to implement the ME processor core in an FPGA device.

Table 1: Implementation results of the ME core using the Virtex-II Pro XC2VP30 FPGA device.

Occupied Slices	811 (5%)
Occupied LUTs	1235 (4%)
Estimated Equivalent Logic Gates	6 kGates
Occupied BRAMs	4 (2%)
Maximum operating frequency	100.3 MHz

The functionality of the implemented video encoding system was successfully verified by encoding a set of benchmark QCIF video sequences with quite different characteristics in terms of movement and spacial detail, and by using several different ME algorithms. The adopted encoding used the typical set of video coding parameters: 8-bits to represent the pixel values, MBs with 16×16 pixels and search areas with 32×32 pixels. This performance assessment considered the Full-Search Block-Matching, the Three-Step-Search and the Diamond Search ME algorithms, which were programmed using the instruction set presented in section 2.

To increase the system performance so as to achieve real-time video encoding of higher resolution video sequences, the implemented encoding structure may be easily expanded in order to include multiple ME cores of the proposed ASIP. Such multicore structure will allow a truly parallel computation of the motion vectors corresponding to several different reference macroblocks, thus significantly decreasing the overall processing time.

4 Conclusions and Future Work

This paper presented the implementation of a complete video encoding system using an heterogeneous multicore structure. The multicore system includes an efficient and dedicated core for motion estimation and a general purpose processor, to implement the remaining tasks of the video standard. The presented results demonstrate that this architecture is capable of performing real-time video encoding of QCIF video sequences. Future work addresses the inclusion of more ME cores to increase the system performance, in order to achieve real-time encoding of higher resolution video sequences.

References

[DMRS07] Tiago Dias, Svetislav Momcilovic, Nuno Roma, and Leonel Sousa. Adaptive motion estimation processor for autonomous video devices. *EURASIP Journal on Embedded Systems - Special Issue on Embedded Systems for Portable and Mobile Video Platforms*, (57234):1–10, May 2007.