

DYNAMIC VOLTAGE SCALING FOR POWER EFFICIENT MPEG4-SP IMPLEMENTATION

Antoni Portero[†], Guillermo Talavera[†], Marius Montón[†], Borja Martínez[†], Francky Cathoor^{*}, Jordi Carabina[†]

[†]Dept. of Microelectronics and Digital Systems
University Autonomous of Barcelona
08193 Bellaterra Spain
{name.surname}@uab.es

^{*}ESAT - INSYS, Kapeldreef 75,
K.U.Leuven
BE-3001 Heverlee, Belgium
Francky.Cathoor@esat.kuleuven.be

ABSTRACT

Traditionally, engineers design for the worst case scenario but in most cases the maximum performance is not required so that there is an important waste of energy consumption. Developers should design systems for different power consumption versus execution time trade-offs. By exploiting Dynamic Voltage and Frequency Scaling (DVFS) techniques we can reach different computational/power trades offs points and thus design power efficient platforms. In this paper, we present a high level methodology to get an optimal set of working points for an MPEG-4 Single Profile (SP) Video encoder implementation. The flow starts from a C++ description of a MPEG-4 encoder which is translated to a SystemC implementation which will be analyzed and further mapped into different platforms. Refined code is migrated to four different processor architectures: a processor research framework (trimaran), a soft core processor with specific functional units implemented on an Altera FPGA, an ASIC and a typical DSP.

1. INTRODUCTION

Multimedia applications are usually fully specified in software oriented languages (like, Java, UML, SDL, C++), starting from a reference or golden model, that need in many cases, to be executed in real-time cost/energy-sensitive heterogeneous SoC platforms.

Current and future SoC platforms will have to satisfy many critical requirements such as energy efficiency for non deterministic applications and should still provide enough computation and communication capacities, etc. at the same time. To satisfy these requirements, most SoCs will contain several types of processor cores and memory units. In most state of the art designs such as: OMAP and daVinci Digital Media SoC from TI or cell processor from IBM [1]. These units will be connected through a hierarchical shared bus or a NoC . These platforms can handle real-time MPEG-4 SP compression and are based on heterogeneous solutions containing at least one DSP for multimedia data flow acceleration and one processor core for control flow and interactivity.

In this paper, we present an energy aware design flow for a MPEG-4 Video Single Profile (SP) specification. The flow goes from the C++ specification into C and SystemC. At the SystemC level, optimal energy work points are evaluated, and then, the code is migrated to different real platform architectures. The goal is to give criteria (at design time) in order to allow the decision (at run time) on what is the most power-performance efficient assignment between code tasks and processors. Up to our knowledge, no such code implementation has been done yet, starting from the same source (C++ Code).

The remainder of the paper is organized as follows. In Section 2 we describe the design flow followed. Section 3 presents the algorithms selected for study. Section 4 details the SystemC energy aware models. In Section 5 shows the target platforms where the code is mapped. Section 6 and 7 present our results, conclusions and future work

2. FLOW DESCRIPTION

The starting point of the flow was a C++ specification of the MPEG4 standard. From this specification, we develop a MPEG-4 Single Profile (SP) code without dynamic memory allocation and without pointers. This type of code is needed for latter SystemC synthesis.

Afterward, on this refined C++ code, we carry out a series of data structure transformations (DTSE [2][3]) to obtain code that minimizes memory transfers between Level 2 memory (L2) and the internal cache (L1). The resulting code after DTSE transformation is a platform independent optimal code. Further platform dependent transformations could be done afterwards. One of the platform dependent transformations is to optimize the size of the data structures for an efficient fit on the specific target memory architecture.

Later on, we modeled a SystemC description of the resulting C++ code and we analyzed the performance and gain obtained by applying Dynamic Voltage and Frequency Scaling (DVFS) [4] for multiple clock domains (frequencies). As the result of the analysis we can derive different performance-energy efficient working points. A scheme about the code transformations is shown in Fig 1.

3. ALGORITHM SELECTION

Our SystemC model is partitioned in two parts: data flow and control flow dominant part. The data flow dominant part is mostly responsible of the transformation of each pixel in every macro-block. In our case, the algorithms list needed from the most intensive computation to the least for the dominant part is: Motion Estimation (FSME) [5] Discrete Cosinus Transform (DCT), Motion Compensation (MC), Quantization (Q) and Zig-Zag (ZZ). These pieces of the model will represent the kernel of our application; they are computational intensive and good targets to exploit any available Data-Level (DLP) or Instruction-Level parallelisms (ILP).

The control flow dominated part is related to the management of the transport stream and Arithmetic Coding (AC) that could be easily implemented in a micro-processor and does not consume many resources. In the rest of the paper, we will call that part of the MPEG model as COD. The implementation of the control flow part in software also provides another advantage: the reuse of the algorithms between MPEG-2 and MPEG-4 SP due to the similarity between both.

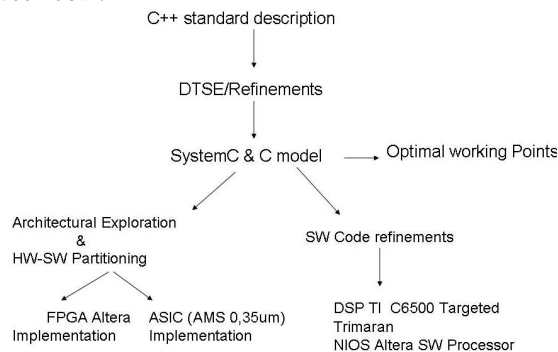


Fig. 1. Design flow related to code transformation

Differences in algorithm complexity, processing computation and configurations to handle macro block computation level is managed by the QoS module to obtain the user requirement for a real time implementation

For our experiments, we studied the algorithms required to compress a macro block. Macro block can be: I (intra), P (predicted) or B (bidirectional). I blocks are just spatially compressed with the two dimensional DCT algorithm. P macro-block temporally compresses with a forward ME and also a spatial compression. B macro-block needs two ME (forward and backward vectors) and a spatial compression. Hence, I macro-block is the less computational intensive and B ones have the higher requirements.

4. SYSTEMC ENERGY MODELLING

The energy model used in our flow takes profit of the fact that multimedia algorithms are based on some computationally intensive loops that are the kernel of the application. In the model, we consider the energy of the data computation and the transmission between the different memory layers as main sources of power consumption.

In our case, the most computational intensive part of the algorithm is the encoder that requires in its inner loop a Multiply-Accumulate (MAC) operation. The SystemC description of the encoder allows the evaluation of different implementations with one, two, four and eight Functional Units (FU) hence being capable to exploit ILP and DLP.

The energy and power figures were estimated from different technologies and scaled to 90 nm technology with different values of V_{DD} : 1V, 1.2V, 1.5V and 1.95V. The technological parameters used as the basis for our case study come from reference [4].

5. PROCESSOR AND HARDWARE IMPLEMENTATIONS

Thanks to the SystemC model, we can obtain different work-points with different energy-computation trade-offs. Afterwards, the application has been targeted in three different real platforms to evaluate the performance achievable with different platforms and hardware architectures.

5.1. FPGA and ASIC mapping

The QuartusII-FPGA Nios II Development Kit, Stratix II Edition is a HW-SW platform for reconfigurable systems and ASIC prototyping. The Altera architecture consists on fine-grained reconfigured logic and diverse memories resources.

5.1.1. Embedded processor on a FPGA

This experimental system is based on an Altera NiosII [6] soft-core processor and uses the Development Kit Cyclone Edition running a NiosII system with CPU set to economical configuration. The system clock is the board default clock of 50MHz. Also, we set-up all RAM memory present in the board (1 MByte) available to the processor.

5.1.2. kernel mapped on an ASIC

Starting from the SystemC description is also possible to reach ASIC technologies, In our case, we dispose of the design flow for Austrian Microsystems AMS 0,35 μ m 4LM process. We have mapped the kernel part of our code in this technology.

5.2. Target Processors

We have decided to employ diverse processors to get different performance and decide what code part is more suitable to be implemented in each one.

5.2.1. VLIW Processor

With the cost of silicon area decreasing, Very Long Instruction Word (VLIW) solutions are becoming an interesting and powerful trade-off between design complexity, flexibility and power consumption. In order to deliver high performance, a VLIW must exploit instruction level parallelism (ILP) available in the application. In this work; we used a real processor, a Texas Instruments DSP and a research framework for architectural exploration: CRISP-Trimaran.

5.2.2. TIDSP

The TMS320C64x [7] core is a VLIW processor core specifically designed to maximize channel density in communications infrastructure equipment. It contains two identical clusters with four functional units each and represents a typical clustered DSP.

5.2.3. CRISP-Trimaran

The CRISP framework [8] is a retargetable compiler and simulator framework based on Trimaran [9] (Trimedia Technologies Inc 1999). The architecture described by CRISP consists on a number of Functional Units (FUs) with coarse-grained reconfigurable logic.

6. RESULTS

In this section, we show the results of the energy exploration and the performance results in the target processors and platforms.

6.1. Energy-optimal work points

Derived from our SystemC model Fig. 2. shows the work points corresponding to different configurations. These points are obtained with different power and performance constraints. These points are obtained for different hardware architectures (number of FUs), supply voltages and running at different frequencies (150, 200, 450 and 600 Mhz).

For a given frequency, the most power hungry points in Fig. 2. correspond to the large number of FUs with higher supply voltage but they need less time to compute any macro block. For example, to compress a macro block, we can use eight FUs for the MPEG kernel at high frequency and voltage, then processor is quite power hungry. This solution will nevertheless be the fastest one to compute the macro block. On the other hand, if we do not have strict timing constraints, we can reduce voltage and frequency and then decrease power consumption using just one FU. Intermediate points were also obtained with diverse power-timing trade-offs.

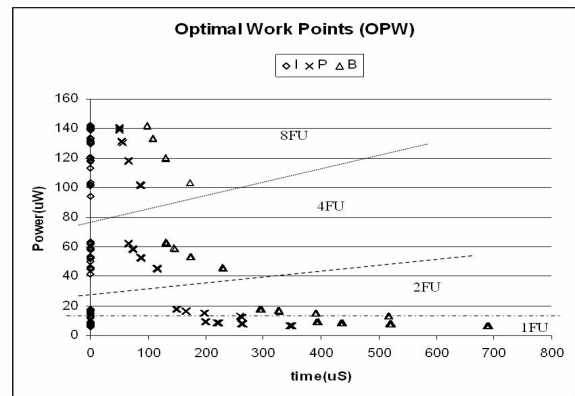


Fig. 2. Optimal Work Points (OWP).

6.2. NIOS SW-Embedded Processor on a FPGA

The results of the implementation of the MPEG4-SP encoder on a NIOSII-FPGA are shown in Fig 3. The processes considered, the MPEG kernel and the COD functions, always consume more than 96% of the total cycles.

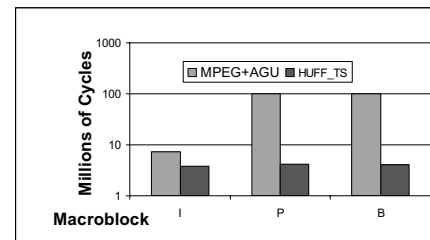


Fig. 3. Execution Cycles on a FPGA with an embedded RISC without FUs.

6.3 Kernel in dedicated HW and COD on NIOSII

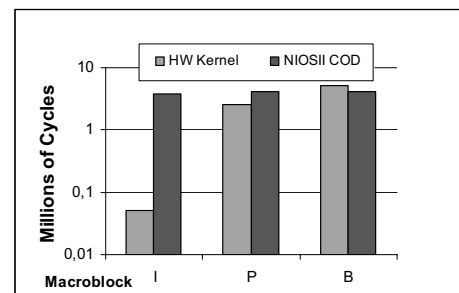


Fig. 4. Execution Cycles for the Kernel Code in dedicated HW and COD on NIOSII

The results of the implementation of the MPEG4-SP encoder on a dedicated hardware and COD on a NIOSII are shown in Fig 4. We can observe that dedicated HW

accelerates a lot the kernel code processing but that COD can not be implemented on a NIOSII and performance requirement should be quite more tight in comparison with an ARM to get real time.

6.4 TI DSP

On a TMS320C64x core the total cycles of execution of the different macro blocks are shown in the Fig. 5.

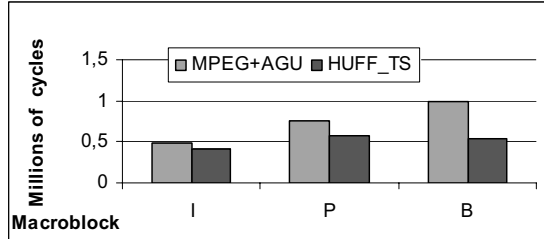


Fig. 5. Execution Cycles of execution of I,P and B macroblocks on a TI-DSP.

6.5 CRISP-Trimaran

On the Crisp-Trimaran framework, the results are shown in the following figure:

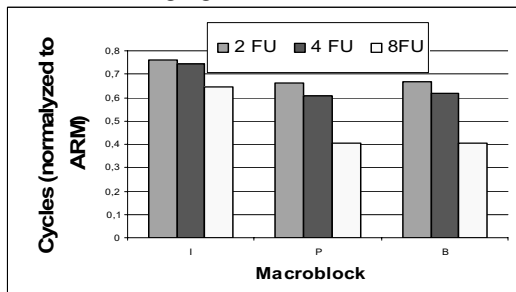


Fig. 6. MPEG Kernel Execution Cycles with r different hardware Functional Units on the CRISP framework.

As a base of comparison for the experiments, we took an ARM7TDMI [10] as a reference and the results obtained are referred to this processor.

6.6 Synthesis MPEG with ForteDS

To evaluate the HW size for FPGA and ASIC implementation, we used ForteDS Synthesizer [11], a behavioral SystemC synthesis tool. We synthesized the MPEG kernel encoder to an Altera FPGA BASIC synthesis (only 1 FU) and to an ASIC from Austrian Microsystems technology (0,35 μm four metal layers). We verified (using ModelSim) that the SystemC MPEG behavioral model is equivalent to the RTL model

synthesized with ForteDS. Synthesis results are shown in the following figure:

MPEG kernel (1FU)	LUTs	Registers	DSP Block 9-bit elements	Pins
Stratix II EP2S60F672C3	25183 / 60440 (42%)	1220 bits	2/36(<1%)	78
AMS 0,35 μm (C35B3L4)	Area (square mm ²)			
	Cell Area	Memory Area	Dynamic Power	Leakage Power
	1.8 mm ²	5,7 mm ²	17,9 W	44,4 μW

Fig. 7. Area for the two MPEG kernel HW.

7. CONCLUSIONS AND FUTURE WORK

In this paper we presented a flow that goes from a C++ description of an application to a SystemC implementation. From the SystemC implementation we can derive some optimal work points. Also, the same application has been mapped into 5 different platforms which show diverse hardware configurations and the time to accomplish the encoding of I, P and B macro blocks.

Acknowledgement:

Special thanks to David Pursley and ForteDS engineering team for helping us to synthesize the MPEG SystemC code to the FPGA platform. Thanks also to Dr. J.L.Merino from IMB-CNM (Centro Nacional de Microelectrónica) for mapping the code to an ASIC.

8. REFERENCES

- [1] Dac C. Pham, et al, "Overview of the Architecture, Circuit Design, and Physical Implementation of First-Generation Cell Processor", IEEE Journal of Solid State Circuits, Jan2006.
- [2] F. Catthoor et al., "Custom Memory Management Methodology", Kluwer AP. 1998.
- [3] F. Catthoor et al, "Data access and storage management for embedded programmable processors", Kluwer AP. 2002.
- [4] A. Azevedo et al, "Profile-based dynamic voltage scheduling using program checkpoints," in *Proc. IEEE Design, Automation and Test in Europe Conference.*, March 2002, Paris, France.
- [5] Peter Kuhn, "Algorithms, Complexity Analysis and VLSI Architectures for MPEG-4 Motion Estimation", Kluwer A.P.
- [6] <http://www.altera.com/products/ip/processors/nios2/ni2-index.html>.
- [7] TMS 320C000 CPU and Instruction Set Reference Guide, SPRU189F, Texas Instruments. October 2000.
- [8] P. Op de Beeck, et al, "Crisp: A template for reconfigurable instruction set processors". In *FPL*, pages 296–305, 2001, Belfast, Ireland.
- [9] The Trimaran Compiler Infrastructure, <http://www.trimaran.org>
- [10] <http://www.arm.com/products/CPUs/ARM7TDMI.html>
- [11] <http://www.fortedds.com/products/cynthesizer.asp>