Power optimization for H.263/MPEG-4 VLSI video coding

A. CHIMIENTI*, L. FANUCCI†, R. LOCATELLI◊, S. SAPONARA◊

*National Research Council, CSMDR, Via Diotisalvi 2, I-56122 Pisa, Italy
Tel: +39 050 568 668 Fax: +39 050 568 522

◊University of Pisa, Department of Information Engineering, Via Diotisalvi 2, I-56122 Pisa, Italy

Abstract: In this paper the design of a VLSI architecture for H.263/MPEG-4 low-power video coding is addressed. Coder high-level modeling and relevant software profiling determines a hardware-software system partitioning based on a RISC engine enhanced by dedicated hardware processing. To reduce the system power consumption two main strategies have been pursued. The first consists in the design of a low-power high efficiency hardware co-processor for motion estimation. It is based on a fast predictive algorithm which, exploiting the correlation of video motion field, attains the same high coding efficiency of the full-search approach for a computational burden lower than about two orders of magnitude. Combining the decrease algorithm complexity with proper low-power VLSI design techniques the motion estimation power consumption can be reduced down to $2 \text{mW}$. Secondly, after a trade-off analysis in terms of memory area complexity and memory and system bus power consumption, a proper buffer hierarchy configuration have been implemented for the DMA-based hardware-software interface. With reference to the trivial case with no buffering a memory and bus power reduction of 78% and 90% respectively can be achieved. Finally, performance measurements on a FPGA-based emulation platform validate our analysis.

Key-Words: Image coding, Motion Estimation, Low-Power, VLSI Architecture, Hardware-Software co-design.

1 Introduction

Real time video processing has become a key issue for the success of consumer applications, such as portable multimedia devices and the forthcoming 3G mobile telephony system (UMTS), where the low-power and low-cost constraints are mandatory. In this scenario video compression plays a fundamental role to reduce the enormous bit-rate for transmission and storage. Communication applications like video telephony, remote surveillance, emergency systems and wireless multimedia are efficiently covered by the H.263 and MPEG-4 simple profile recommendations [1][2]. These standards combine computation intensive low-level tasks, featuring a regular computation on simple data structures (like motion estimation and discrete cosine transform), with data dependent medium tasks characterized by an irregular data flow with a lower computational demand. A complete software (SW) implementation based on a microprocessor platform gives the highest flexibility but usually, for real time applications, it does not provide sufficient performance and/or low energy consumption [1][3][4]. On the other hand, fully dedicated hardware (HW) solutions are limited by a poor flexibility and reusability. Dedicated and programmable approaches can be efficiently combined in the design of hybrid architectures reflecting the inner structure of multimedia processing that consists of multiple tasks with different computational burden. At present the best suited solution for the design of flexible, low-complexity and low-power video processing scheme is embedding a reduced instruction set computer (RISC) engine with HW dedicated units for the intensive computation tasks. According to this approach several architectures have been recently proposed in literature to integrate in a single-chip a whole H.263/MPEG-4 codec for a power consumption lower than $300 \text{mW}$ [3][5]. Unfortunately they present two main drawbacks still to overcome: the design of a low-power but high efficiency motion estimation (ME) algorithm and the optimization of the HW-SW communication memory hierarchy to reduce system power consumption by exploiting data reuse. To this objective in this paper we propose a hybrid architecture for H.263/MPEG-4 video coding which addresses both the above issues. First the system HW-SW partitioning has been derived as a result of coder high level modeling and relevant SW profiling.
on RISC micro-architecture. Special emphasis has been devoted to the design of a low-complexity and low-power VLSI macrocell for ME and to its integration in a System-on-Chip (SoC) architecture. It is based on a fast predictive algorithm which, exploiting the correlation of video motion field, attains the same high coding quality of the full-search (FS) full-pixel approach for a computational burden lower than about two orders of magnitude. Then to reduce the impact on system performance of large data transfer the exploration analysis of a DMA-based communication between HW and SW tasks and relevant memory hierarchy organization have been performed. Finally, to validate our analysis the whole system has been prototyped on a FPGA-based emulation platform and its measured performance has been compared with the ones of the well-known FS based TMN coding scheme [6]. After this Introduction in Section 2 we briefly describe the co-design methodology and the relevant system profiling and partitioning. Section 3 addresses the issue of a low-power but high efficiency ME technique. In Section 4 we detail the proposed memory hierarchy organization. Section 5 describes the architecture prototyping and relevant performance measurements. Some conclusions are drawn in Section 6.

2 Video coding VLSI architecture

The target system is a low-power and low-complexity architecture for H.263/MPEG-4 simple profile video coding suitable for real time processing of full motion QCIF video for a maximum transmission bandwidth of roughly 500 Kbit/s. In such a coder the ME task aims to reduce the temporal data correlation between successive frames (inter coding) of a video sequence while the discrete cosine transform allows for the reduction of the spatial correlation (intra coding) of each frame. The whole scheme also includes data dependent tasks for quantization, variable length coding, bit-stream generation, system control and I/O. Starting from a high-level C description of the whole coder, a profiling analysis indicates a possible system HW-SW partitioning assuming area, power consumption and flexibility as main cost functions. Profiling data have been collected running the C code on RISC micro-architecture according to a computational cost approach. The resulting percentage breakdown of the different functions indicates that the most demanding task in terms of computational power (and so in terms of energy consumption) is the ME one (nearly 70 %) which has been selected for a dedicated HW solution. SW implementation on a programmable engine is the best-suited solution for the remaining video coding tasks. Particularly, after proper algorithm optimizations (such as fast implementation of the DCT/IDCT tasks based on the known Chen [7] algorithm with fixed-point arithmetic) and handcrafted code refinement the computational power requested by the SW tasks can be supported by a low-power, low-cost ARM9 microprocessor. Fig. 1 shows the architecture block diagram.

![Software ARM 9 Processor I/O video buffers](image)

![Frame Memories DMA System Bus](image)

**Fig. 1** Proposed SoC hybrid architecture

The resulting system partitioning is aligned with the ones presented in literature based on a RISC engine enhanced by dedicated HW processing. As it will be detailed in Section 4 the communication between the two agents, i.e. the HW-SW interfacing problem, is addressed by proper buffer levels. This way, an efficient and low-power memory hierarchy exploiting temporal locality and data reuse has been implemented. A DMA engine ensures best performance to the processor data management by handling all the data transfers between system frame memories, communication memory and I/O buffers.

3 Low-power motion estimation

A straightforward technique for the design of a ME engine is that of FS full-pixel [1] [2] [8-9] where the current frame of a video sequence is divided into reference $N \times N$ blocks and, for each of them, a block in the previous frame (candidate block), addressed by a Motion Vector (MV), is exhaustively searched for the best matching within a proper search area according to a SAD (Sum of Absolute Differences) cost function. If $a(i, j)$ and $b(i, j)$ are the pixels of the reference and candidate blocks and $m$ and $n$ are the coordinates of the MV (with $-p \leq m, n \leq p-I$), the SAD is defined as follows:

$$SAD(m, n) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |a(i, j) - b(i + n, j + m)| \quad (1).$$

This distortion is computed for all the $(2p)^2$ possible positions of the candidate blocks within the search window. The block corresponding to the minimum distortion ($SAD_{min}$) is used for prediction and the
relevant MV is given by: $MV = (m,n)[SAD_{min}]$. The FS full-pixel approach achieves optimal performance in terms of PSNR (Peak Signal-to-Noise Ratio) for a given compression factor but at the expenses of high computational burden and data bandwidth. For instance the real time processing of a 30 Hz QCIF video, considering $N=16$ and $p=16$, requires about $780 \times 10^6$ absolute difference operations per second. To reduce the ME complexity several different block matching algorithms have been proposed and implemented in state-of-art single-chip video coding scheme. They are all based on the reduction of the number of candidate blocks and/or the number of pixels investigated for each candidate block, such as the three-step-search (TSS), hierarchical search, 2D log-search, pixel sub-sampling [2], [8], [10]. To further reduce the power consumption these algorithms can be joined with proper circuit clock gating strategy. Processing is stopped when a partial SAD exceeds the current SAD$_{min}$ because it will never be selected as minimum distortion value [5]. Unfortunately, adopting these ME algorithms the lower computational complexity achieved is paid with an increase of the bit-rate or, in case of constant bit-rate transmission, with a lower quality of the coded image. To overcome the complexity of the exhaustive search while maintaining the same coding efficiency for the considered low bit-rate applications, we propose a fast predictive spatio-temporal algorithm. In a typical video sequence, particularly in low bit-rate coding, the motion field is usually slowly varying with a high correlation along both horizontal and vertical directions. By exploiting this correlation, the MV of a given block can be predicted from a set of 4 initial MV candidates, 2 selected from its spatial neighbors and 2 from its temporal ones, according to the minimization of the SAD cost function (predictive phase). To further reduce the residual estimation error the initial predictive phase is followed by a refinement phase on a grid centered around the position pointed by the predictive phase winner, hereafter called V0, and made up of 4 points on cross directions and 4 points on diagonal ones. To reach half pixel resolution the points on cross directions have $1/2$ pixel distance from the center while the points on the diagonal ones have 1 or 3 pixel distance. The amplitude of the grid corner points is selected according to this rule: if SAD(V0) is greater then a proper threshold it means that V0 is likely to be a poor predictor and so the search area must be enlarged. Since this happens specially in case of scene change or sudden motion change, grid amplification allows a quicker recover of the true motion field. We also consider the calculation of the cost function for the null vector for a total complexity of just 13 SAD evaluation for each macro block (MB) processing with respect to the 1032 SAD evaluation of the FS approach with the commonly used search window of $16 \times 15.5$ pixels. It is worth mentioning that the SAD for the null vector may be reduced by a proper threshold to improve the coding efficiency (Static Priority Option, see [9]). As it will be detailed in Section 5 and Table 1 an exhaustive test campaign on a FPGA-based codec emulator validate the strength of the proposed predictive ME algorithm. It attains the same high video compression quality of the FS full-pixel technique outperforming other fast algorithms such as the TSS adopted in [4], [5]. The proposed technique has been implemented as intellectual property (IP) VLSI macrocell (patent filed) whose block diagram is sketched in Fig. 2.

![Block diagram of the fast ME coprocessor.](image)

The main unit of the IP is the ME_Engine which is able to process the SAD cost function and the detection of the MV that minimizes it. The backbone of this engine is a parallel array of 8 processing elements each of them implementing, at pixel level, an absolute difference operation. The engine also incorporates a parallel array of 8 interpolation modules needed for the implementation of the half-pixel accuracy search. To be noted that the chosen size of the processing array is the result of a trade-off between circuit complexity and degree of parallelism to attain clock rate reduction. Other units of the IP are in charge of the management of both data flow and memory resources to permit the implementation of the algorithm functionality according to a pipeline processing. Particularly the internal MV memory stores the set of MV predictors for all the blocks of the reference frame. Since the considered predictive algorithm features a fixed computational workload and regular data flow, the control unit is realized by a simple finite state machine. The block diagram of Fig. 2 also includes a reference buffer hierarchy and a candidate one to exploit data reuse reducing system bandwidth and power consumption (see Section 4). This cell has been characterized by means of logic
synthesis on a 0.18 μm standard cells CMOS technology. The total complexity amounts to 27 Kgates for the logic plus 150 bytes of single port MV RAM. Combining the decreased algorithm complexity with the proper use of parallelism and pipelining at architectural level, a large reduction of the required clock rate is obtained. The proposed ME coprocessor requires 687 cycles for each 16 x 16 reference MB supporting real time processing of 30 Hz QCIF video with a clock frequency of nearly 2 MHz. The clock rate reduction, and the consequent lowered circuit speed requirement, can be exploited for power saving. This can be achieved by scaling down the supply voltage and/or by using low-speed, low-leakage version of the considered standard cells library when available. For instance, the adopted 0.18 μm CMOS technology provides two different versions: a Device High Speed (DHS) optimized for low circuit propagation delay and a Device Low Leakage (DLL) optimized for low leakage-power consumption. The optimization is mainly obtained by using two different threshold voltages which are scaled down of about 20% going from DLL library to DHS one. Gate level simulations demonstrate an average dynamic power consumption of 2 mW at 1.6 V and 40 °C for typical MPEG-4 video sequences. By using the DLL library the leakage power contribution is negligible since it amounts to 2.5 μW instead of the 750 μW for the DHS one. The achieved results in terms of circuit complexity, power consumption and coding efficiency are very interesting when compared with state-of-art ME technique. For instance, the low-complexity FS systolic array presented in [9] required 29 kgartes plus 9Kbits of dual port RAM and about 42 mW for a 30 Hz QCIF. On the contrary the solutions adopted in recent single-chip architectures [4],[5] achieve the same low-power consumption of few mW (i.e. the ME processing dissipates 5% of the total 60 mW in [4]) for a 10 Hz QCIF) but at the expenses of a reduced coding quality since they are based on the TSS algorithm. Moreover the low-power performance is also due to full-custom circuit optimizations such as clustered voltage scaling and variable threshold voltage scheme which reduce the portability of these solutions to different silicon technologies.

4 Memory hierarchy
A large percentage of the system power consumption in data dominated applications, such as real time video coding, is due to data communication and storage in large background memories [11]-[12]. In order to reduce this power component the design of a proper memory hierarchy between the processor and the ME HW coprocessor is outlined in this section. The basic idea is that exploiting temporal locality and data reuse by introducing an optimized multi-level buffer hierarchy a great reduction on the overall energy consumption can be achieved. Power savings are expected when high read read/write access rates are related to smaller buffer instead of large system memories [11].

According to the system architecture sketched in Fig. 1, we focus our data flow analysis on the communication between the frames memories and the data path registers (level 0) of the ME HW co-processor. This communication can be split into three separate data flows independently optimized: reference and candidate inputs to the ME engine and output ME results (MV and SAD represented on 4 bytes). These ME results are represented on few bytes with respect to the other input data, so the relevant memory organization has a limited impact on the whole analysis. Then, we concentrate our exploration only on the reference and candidate memory hierarchy.

Starting from the trivial solution, which does not include any buffer, we have selected all possible configurations taking into account two main rules. According to the first one, when more levels are introduced the buffer size have to increase from lower (data path registers side) to higher levels (system bus side). For the second one, introducing levels, power saving is expected only if access throughput to/from a memory is reduced.

Our exploration and evaluation of the selected configurations is based on the following power and area memory models (P_memory and A_memory) [11]-[14]:

\[
P_{\text{memory}} = \frac{1}{2} V_{\text{dd}}^2 C_{\text{read}} F_{\text{read}} + \frac{1}{2} V_{\text{dd}}^2 C_{\text{write}} F_{\text{write}}
\]

\[
A_{\text{memory}} = \beta \cdot \sqrt{\text{word} \cdot \text{bit}}
\]

where \( F_{\text{read}} \) and \( F_{\text{write}} \) represent the throughput of read and write accesses, \( C_{\text{read}} \) and \( C_{\text{write}} \) are the read and write equivalent capacitances (which depend on the size – word x bit - of the buffer), \( V_{\text{dd}} \) is the power supply voltage and \( \beta \) is a technology dependent parameter. In the above area and power memory models we only consider single port memory configurations because dual port ones determines nearly double the power consumption [14]. Our analysis takes into consideration also the system bus dissipation. In that respect, a simplified power model of the bus can be expressed as [13]:

\[
P_{\text{bus}} = \gamma \cdot C_{\text{bus}} \cdot B_{\text{bus}}
\]

where \( B_{\text{bus}} \) and \( C_{\text{bus}} \) represent the data bandwidth and the equivalent capacitance of the main bus respectively, and \( \gamma \) is a technology dependent.
parameter. We apply that model only to the system bus, considering the relevant bandwidth. Assuming that the equivalent capacitance of the main bus \(C_{bus}\) is much higher than other intra buffers connections, thus the power consumption due to the communication between buffer levels is negligible.

Assuming that the equivalent capacitance of the main bus \(C_{bus}\) is much higher than other intra buffers connections, thus the power consumption due to the communication between buffer levels is negligible.

### Fig. 3 Reference memory hierarchy configurations.

### Fig. 4 Candidate memory hierarchy configurations.

Fig. 3 and 4 shows a selection of the most interesting configurations for the reference and candidate flows respectively (\(Row\) is the number of MB into a picture line and it amounts to 11 for QCIF case). \(Old\) and \(New\) are the two frame memories which store the Luma (Y) components of the current frame to be processed (\(New\)) and the reconstructed previous one (\(Old\)). For both reference and candidate cases the trivial configuration without any buffer (denoted as A in Figs. 3 and 4) is considered to evaluate the final improvement of this approach. It has to be noted that buffers at level 1 of the configurations C and D refer to a prefetch scheme. In particular, we adopted a 1 MB prefetching in the reference data-flow (Fig. 3) and 3 MBs in the candidate one (Fig. 4). Data prefetching is a simple and efficient solution for the data flow synchronization between HW and SW. It provides a hidden communication scheme which avoids any stall in the ME data-path. Otherwise, a proper pre-load phase in the HW accelerator has to be considered (configurations B and E). Another possible solution for the HW-SW interfacing is based on dual port memories but here it is not considered because of its power inefficiency as mentioned above.

Figs. 5 and 6 report normalized values of area and power for reference and candidate memory hierarchies respectively. Normalized system bus power contribution is also depicted. For the candidate case we choose D as the best configuration. With respect to B and C solutions the D one provides more than 66% of bus energy saving for roughly the same memory power dissipation. Even if the area penalty seems to be considerable (40% with respect to B and C cases due to the introduction of level 2 buffer) considering the amount of RAM size of the whole encoding architecture, the \(3 \times Row\) MB level 2 buffer represents only the 6.7%. The chosen candidate hierarchy differs from the one presented in [11] since our analysis includes system bus power evaluation and directly refer to the predictive ME algorithm described in Section 3 instead of the FS.

For the reference case we choose C as the best configuration. First of all the 1xRow MB buffer foreseen in the D and E cases does not provide any improvement with respect to B and C ones because the latter already achieve the minimum read/write access throughput and the minimum system bus bandwidth. Then, among C and B, the former is preferable because it allows for a prefetch scheme with a minimum memory area overhead.

As above mentioned, the chosen prefetch based hierarchy provides a simple HW-SW synchronization mechanism which avoid any stall in the ME processing. This hidden communication saves power in the data path by reducing the required ME clock frequency nearly of 20% minimizing memory area and power costs.

Relating to the trivial solution A without any hierarchy scheme, the memory power reduction amounts to about 86% for reference data flow and 68% for the candidate one; considering the bus power on both data flows the reduction increases up to 90% (in terms of bus bandwidth this corresponds to a reduction from 18.8 Mbytes/s to 1.4 Mbytes/s).

The whole system bus bandwidth for a QCIF 30Hz, considering also video I/O and the communication
between processor and frame memories, is 5.8 Mbytes/s. Without the proposed buffer hierarchy this value becomes 23.2 Mbytes/s. With reference to Fig.1 and including the I/O video buffer, the system frame memories and the proposed ME buffer hierarchy, the total RAM size amounts to 123 Kbytes (the ME buffer hierarchy is less than 10% of this total value).

5 System prototyping
In order to validate the proposed analysis a fast prototyping approach has been chosen rather than a SW co-simulation of mixed HDL-C descriptions. Indeed the enormous amount of data and operations which have to be performed for video processing application determines too long simulations while a video sequence can be coded in few seconds by a real time HW emulator. Moreover system emulation allows for testing the quality behavior of the video algorithms by direct checking of an image on an output screen. The prototype consists of a general purpose PC platform and a PCI board with a reconfigurable HW based on FPGA technology. A PENTIUM III at 800 MHz was selected as host processor to implement the SW part of the video coder, providing enough computational power to support real time processing. The Celoxica RC1000-PP [15] the selected FPGA-based PCI development board, is equipped with a Xilinx Virtex FPGA XCV1000, which provides 1 million of system gates. The IP ME macro cell, including the buffer levels, was implemented in FPGA technology by means of logic synthesis occupying less than 30% of the XCV1000 resources. A DMA engine complete the description of resources used for mapping the HW-SW architecture into the breadboard prototype. The host PC grabs real time video data by a commercial web camera, converts them into the appropriate format and eventually sends the coded bit stream into Internet Protocol network to any connected workstation equipped with the decoding SW. This way it is possible to test the system with real sequences, improving the completeness and efficiency of the validation. A quality analysis, based on several test conditions (standard sequences and real ones grabbed by the acquisition system), compares the efficiency of the new fast algorithm implemented by the prototype with respect to the SW implementation of the well-known TMN coder considering both FS and TSS algorithm for ME [6].
Both the cases of constant bit-rate coding and variable bit-rate coding have been considered. For example Table 1 compares the obtained bit-rate versus the quantization parameter Q for the 30Hz QCIF test sequences Akiyo and Coastguard (the quantization steps were fixed to have near constant quality, producing therefore variable bit-rate). Similar figures have been obtained comparing the prototyped system and the SW implementation of the TMN for the other test cases.
The results demonstrate that the proposed coding scheme with the fast predictive ME attains the same high video compression performance of the TMN with the FS full-pixel and outperforms the TSS algorithm. It is worth mentioning that for some cases
(e.g. $Q=20$ in Table 1) the prototype performs even better than the TMN-FS since the predictive ME algorithm tends to find more regular motion fields which require fewer bits to be coded (TMN adopts a differential coding scheme for the MV field).

Table 1. Prototyped system comparison vs. TMN

<table>
<thead>
<tr>
<th>Bit-Rate (Kbit/s)</th>
<th>Akiyo</th>
<th>Coastguard</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$Q=5$</td>
<td>$Q=10$</td>
</tr>
<tr>
<td>Our</td>
<td>61.69</td>
<td>23.07</td>
</tr>
<tr>
<td>TMN-FS</td>
<td>59.32</td>
<td>22.56</td>
</tr>
<tr>
<td>TMN-TSS</td>
<td>75.09</td>
<td>24.91</td>
</tr>
</tbody>
</table>

6 Conclusions
The design of a VLSI architecture for H.263/MPEG-4 low-power video coding has been addressed. A co-design approach has been followed to define a hybrid RISC microprocessor based architecture. The ME function has been selected to be implemented as a dedicated HW coprocessor based on a low-power and high efficiency fast predictive technique exploiting the high correlation of video motion field. The proposed algorithm achieves the same high coding efficiency of FS while outperforms other low-complexity ME techniques usually adopted in state-of-art single-chip video codecs. The complexity of the ME cell on a 0.18 µm CMOS technology amounts to 27 K gates for the logic plus 150 bytes of single port RAM for an overall power consumption of 2 mW for typical MPEG-4 video sequences. An exploration analysis of the communication between HW and SW tasks and the relevant memory organisation completes the architecture definition as a key issue for the power optimisation of the whole system. Considering a 30 Hz QCIF video format with respect to the trivial case with no buffering, the proposed approach provides a memory and bus power reduction of 78% and 90% respectively. Finally, performance measurements on FPGA-based emulation platform validate our analysis. With reference to the well-known FS based TMN coding scheme the proposed architecture achieves the same coding efficiency while outperforms other known low-complexity coding technique.

References: