CONFIGURABLE MOTION-ESTIMATION HARDWARE ACCELERATOR MODULE FOR THE MPEG-4 REFERENCE HARDWARE DESCRIPTION PLATFORM

J. Dubois, M. Mattavelli[(*)], L. Pierrefeu, J. Miteran
jdubois@u-bourgogne.fr
Université de Bourgogne, Laboratoire LE2I
CNRS, Dijon, France,
[(*)]Ecole Polytechnique Fédérale de Lausanne (EPFL)
Signal Processing Institute, Signal Processing Laboratory 3
CH-1015 Lausanne, Switzerland.

ABSTRACT

This paper describes a motion estimation co-processor architecture that explicitly separates the implementation stages consisting of data access to the search window and the evaluation of the matching criterion from the implementation of the search strategy. The architecture is modular and can be re-configured according to the different MPEG video coding profiles and level parameters. Although the architectural solutions described here can be in principle applied to any SoC implementation technologies, the solution presented here is expressly conceived and validated on FPGA co-processing architectures supporting mixed SW/HW implementations of video encoders such as generic PC platforms with a standard PCMCIA FPGA cards. The module has been developed in the framework of the MPEG reference hardware description activity.

INTRODUCTION

The motion estimation is well known to be the most computation-intensive stage of video coding process and has been the subject of many research works (on both algorithmic and architecture sides) which aim to reduce the implementation complexity. For brevity we cannot here provide an accurate review of the wide literature on the subject, we suggest referring for instance to [1] [2] and their references, for an updated review of the state of the art, but we just remind the main families of approaches developed so far. Although some algorithmic and architectural solutions can include different elements from different families of approaches, and thus a clear distinction between algorithms belonging to different categories is practically impossible, nevertheless we can roughly partition them into four main families. The first family is the so called "Reduced Search Algorithms" which aim to reduce the complexity by limiting the measure of the matching criterion to only a (small) subset of candidate vectors in the search window. The key element here is the "intelligence" of the search algorithm that may completely change depending on the requirements of the video coding application (portable video telephony, HDTV for sport events, etc…). A second family is constituted by the approaches in which the complexity reduction is achieved by using multi-resolution searches on sub-sampled image search windows. A third approach is to use simplified matching criteria at the place of the classical Maximun Absolute Difference (MAD) criterion. A fourth family of approaches is based on various pre-processing stages that reduce the images to binary images for which simple XOR operations are used for the evaluation of the MAD. For most of the algorithms, using one or more elements from the different families sketched above, dedicated optimized implementations using systolic arrays and/or specific data flows handlings are needed to achieve effective performances as well as relevant complexity reductions. While a lot of effort has been devoted to developing such reduced complexity "solutions" (search algorithm + data flow implementation), much less effort has been devoted to study how to be able to scale the "solutions" versus the different parameters such as the size of the search window, the available memory bandwidth (that usually is a off-chip memory) and the processing power. Most of the "solutions" requires a close coupling between data flow handling and the dedicated hardware. Nowadays, with the appearance of powerful processors with specialized instruction sets and new families of FPGAs with embedded arithmetic for which the MAD evaluation is not anymore a difficult burden, some of the reduced complexity "solutions" presented in the past have lost their interest. Modularity, flexibility to cover the different coding applications and the possibility of upgrading using the more recent results and improvements are getting more and more importance. Moreover, for two other motivations the development of programmable motion estimation block becomes extremely interesting. One is the recent appearance of

0-7803-9134-9/05/$20.00

PCMCIA FPGA cards that can be plugged on standard PC supporting high performance HW co-processing that can be linked to generically optimized SW codecs running on PC or on other DSP platforms [3] [4]. The second is the development of mixed SW/HW standard descriptions of ISO MPEG video codecs [5] [6] [7] that directly provides as starting point for the designer a choice between different SW and HW assisted partitioning including HDL or System C source code of the HW modules.

For such reasons in this work we focus on architectures that present a wide degree of flexibility and modularity of the different elements versus the various performance and implementation parameters of motion estimation so as to be able to cover different applications ranging from high quality HDTV up to mobile video. The family of approaches supported is the reduced search algorithms on a specific search window. The recent results including the comparison with other approaches has shown that using appropriate (for the video application) reduced search algorithms is possible to achieve optimal coding results [1]. The main idea of the architecture so as to achieve the desired level of flexibility and re-programmability for the search is thus to separate the data access and the matching criterion implementation stages from the intelligence of the algorithmic search. Therefore, the co-processor architecture, depending on the support platform and matching criterion used, can be programmed freely by the user programming at high level his preferred "intelligent" search exploiting the resource budget in terms of available "candidate" vectors for each search. This architecture has been designed for recent FPGAs families which provide the usual embedded gates and local memory storage as well as one or more embedded arithmetic processors that can be freely programmed by the final user.

The implementation objective of the architecture is not only an efficient hardware implementation to speed up matching operations, but also to provide the possibility of randomly matching any area in a search window without any other limitation on the access sequence. The random-block access in the search window is obtained with a specific data flow architecture and address generation strategy. Address-generation is processed by a flexible unit (FPGA-embedded DMA or processor) to enable the translation of the user search strategy into the hardware set-up.

The flexibility of the hardware accelerator supports all the different motion estimation modes which appear in the recent extension of the MPEG video standard family such as AVC. The search-window size is parametric so that it can be configured according to the search window size of the profile under consideration or reduced according to the desired application.

2. A MODULAR PROGRAMMABLE CO-PROCESSOR ARCHITECTURE FOR REDUCED SEARCH MOTION ESTIMATION

The major challenge of the architecture design of the co-processor is to allow the setting of the size of the image pattern (i.e. macroblock, block or quarter of a block) the size of the search-window, the random access to guarantee any search strategy to be user defined, providing sufficient processing resources for the different standard versions and profiles. So as to guarantee the random access in any position of a search window the search window pixels have to be accessible to the matching engine and need to be stored in the FPGA to reduce the number of access to the external memory, so as not to exceed the available bandwidth. The theoretical minimization of the bandwidth would cost an internal storage size equivalent to the width of the image in pixels multiplied by the height of the search window. Such solution it is too demanding in terms of internal storage and a better trade-off is to accept an increase of the external accesses bandwidth (which is indeed available on current state of the art SDRAM and ZBT chips) of a factor given by the ratio between the search window height and the macroblock size.

Thus the co-processor is constituted of the internal buffer, the external memory controller and the MAD estimation core. The external memory controller is in charge of the downloading of a pattern and the associated search window into the internal buffer memory. Into the motion estimation core, three elements are in charge of the internal memory interface in the MAD estimation core:

- ✓ one element controls the input throughput provided by the external memory,
- ✓ a second element provide two rows (one from the pattern one from the search window) to the MAD estimation element,
- ✓ a third element is in charge to store temporarily the results before the transfer into the external memory.

So as to provide the required search random access, the address-generation block needs to be flexible. The design has been implemented using DMA and a processor mapped on the FPGA logic. Such block could also be implemented using an embedded processor available in some of the FPGAs families. The control block synchronizes of the system and control commands and relative hand-shaking. The figure 1 reports the main elements constituting the architecture of the co-processor. The data-flow bandwidth is relative to a 16x16 pattern (one macroblock vector estimation).

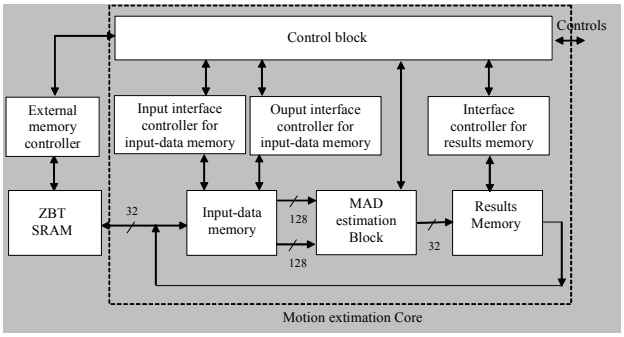

Figure 1. Motion estimation core for 16x16 pattern size.

The MAD matching evaluation criterion has been used in the current version of the implementation, other criterion could be implemented in future version of the architecture. The implemented pipelined architecture enables to match a block and a portion of search window, row by row. At each cycle, the two internal memories permits to access at two rows, one extract form the block and the corresponding one in the search window. An input data memory (Figure 2) and the associated control have been designed in order to provide the row of 16 pixels (128 bits) at each cycle. The internal search window memory has been implemented on the FPGA internal memory (Virtex-II BRAMs).

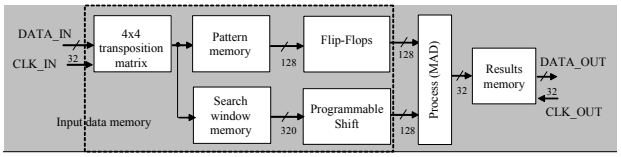

Figure 2. Data flow in the motion estimation co-processor.

An address allows selecting one column into the search window and one into the block, the programmable shift to select 128 bits into the search window column according to the selected matching. The address and shift control are both generated by the control blocks.

At the initialisation it is possible to reconfigure pattern and search window sizes. Currently, the architecture has been validated for any pattern size between 8x8 and 16x16. The extension to smaller pattern sizes such as 4x4 as specified by the recent AVC standard is on-going. Obviously the size of patterns can only be configured at the system initialization. Any search-window width between 20 and 256, can be set according to the available processing. Rectangular search windows, wider in the horizontal dimension have been shown to be very efficient for most of the video applications. Considering the internal architecture of Virtex-II memory blocks (BRAMs) depth, an optional element in charge of the image transposition can be inserted into the data flow whenever the search window height is lower than its width. Such component yields a reduction of the number of Virtex-II memory block (BRAMs) usage. A data transposition example on a 56x40 search window example is reported on Figure 3.

The pixels are coded on a byte on most of the video profiles, so as to improve the system bandwidth efficiency, the external memory has been mapped with 32-bits bus width.

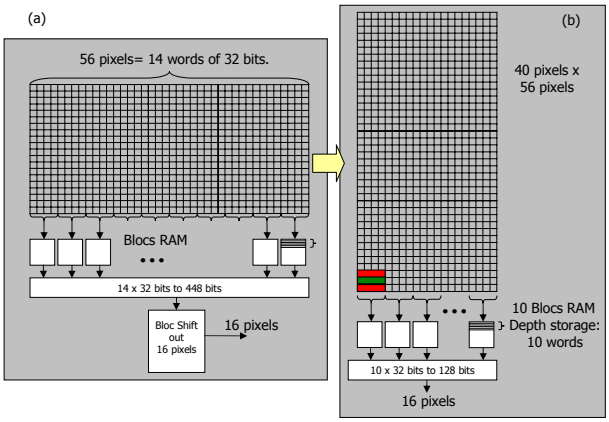

Figure 3. Search window transposition for the efficient usage of Virtex II BRAMs.

The resulting architecture allows to access in one clock cycle any search window column and consecutive pattern matching evaluations do not need to be adjacent in terms of memory locations. The implementation of the component, in charge of the address generation, is quite complex. Two implementations of this component have been investigated: the first is based on a custom DMA described in VHDL language, the second is based on microcode running on a processor (MicroBlaze) mapped on the FPGA. A powerPC based implementation (for The Virtex-Pro and equivalent other families of FPGAs) has also been considered and is currently in development.

3. EXAMPLE OF ACHIEVEABLE PERFORMANCES

All the results reported in this section are obtained for the implementation setting on a search window with a size of 56x40 and a 16x16 blocks (Figure 3.). The ZBT SRAM has 32 bits bus width, and is used with a 100 MHz frequency clock. In the diagram of picture 4, each couple, formed by a 16x16 block pattern and the associated pattern from the desired search window position, is marked with the same number. Input data can be pre-fetched, constituting the next pattern and search window position, simultaneously with the current matching evaluation process.

For a search with a 16x16 pattern and a 56x40 search window, 1025 matching can be processed in 202,5 µs on a VirtetexII FPGAs. Since the transfer time for a 16x16 block and a 56x40 image is 5 µs and the result transfer time of only 2 µs, for a full-search strategy, idle times on data bus represent 96 % of time, which enables to use the ZBT memory by others processing. Therefore, a random access to the search window size can be achieved without throughput limitation, thus enabling "*Reduced Search Algorithms"* on large search windows.

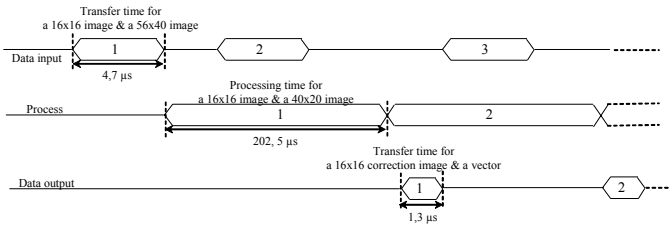

Figure 4. Processing diagram.

Table 1 reports the resource usage of the example above. As can be easily seen the FPGAs presents a large portion of unused resources that can be allocated to other co-processing tasks of the mixed SW/HW codec.

Device utilization summary:				

Selected Device : 2v3000fg676-4				
Number of Slices:	*2559*	*out of*	*14336*	*18%*
Slices FlipFlop…	*2225*	*out of*	*28672*	*8%*
Number of 4 input LUTs:	*2084*	*out of*	*28672*	*7%*
Number of BRAMs:	*14*	*out of*	*96*	*14%*
Multiplier	*3*	*out of*	*20*	*2,5%*

Table 1. Summary of resource utilization of the VirtexII FPGA

Using the MicroBlaze based controller for address generation, the proposed implementation uses only 18% of all available slices in the Virtex™-II XC2V3000-4 FPGA. The current investigations on PowerPC on Virtex-II or Virtex-IV aim to enable us to use complex "*Reduced Search Algorithms"* and then reduce the number of matching. For instance allocating 125 matching per block to the user programmable "*Reduced Search Algorithm*" (process requiring 24.69 us), it can enable the process of 40500 blocks/s, that means a CCIR 720x576 resolution image can be processed in real-time (25 frames/s).

4. CONCLUSION

A fully configurable motion estimation block that explicitly separate data flow handling and search strategy has been designed. It can support any "*Reduced Search Algorithm"* programmable by the user using simple programming language. The user can fully configure the search window and the search strategy in accordance to the specific profile and level. The architecture has been demonstrated on a FPGA family and is compatible with the concept and APIs [5] [6] developed in the MPEG reference HW description activity. Further improvements of the performances (i.e. number of MAD evaluation for a given configuration) are possible duplicating the HW dedicated to the MAD evaluations without exceeding the data flow capabilities but at the expenses of a higher usage of FPGAs resources. Another evolution under study is to extend the architecture concept so that it can process in three parallel stages 16x16, 8x8 and 4x4 patterns as required by the recent AVC standard, as well as introducing a half and quarter pixel refinements stage.

[1] M. Mattavelli, G. Zoia: "Vector tracing algorithms for motion estimation in large search windows", *IEEE Trans. on Circuits and Systems for Video Technology*, December 2000, Vol 10, Nr. 8, pp 1426-1437.

[2] J-H Luo, C-N Wang, T. Chiang: "A novel all binary motion estimation (ABME) with optimized hardware architecture", *IEEE Trans. on Circuits and Systems for Video Technology*, August 2002, Vol 12, Nr. 8, pp 700-712.

[3]Wildcard II data sheet available at: http://www.annapmicro.com/wildcard2.html.

[4] T. Mohamed, et al, "A Rapid Prototyping Framework for MPEG/H.264-enabled Consumer Products", *ICCE 2005*, Las Vegas, Nevada, USA, January 2005.

[5] P. Schumacher, R. Turney, and M. Mattavelli, "Information Technology – Coding of Audio Visual Objects – Part 9: Reference Hardware Description," ISO/IEC JTC1/SC29/WG11/N6091, Hawaii, USA, 8-12 December 2003.

[6] P. Schumacher and R. Turney, "Integrated Framework for MPEG-4 Part 7, Part 9, Part 10," ISO/IEC JTC1/SC29/WG11 N6092, Hawaii, USA, 8-12 December 2003.

[7] ISO/IEC SC29WG11: "Updated Call for the Submission of Hardware Reference Code for MPEG-4 Part 9: Reference Hardware Description", document N6506, Redmond, Washington, July 2004.