A NETWORK-ON-CHIP BASED ARCHITECTURE FOR H.264 MOTION ESTIMATION

Alba Sandyra Bezerra Lopes, Ivan Saraiva Silva

Departamento de Informática e Matemática Aplicada
Universidade Federal do Rio Grande do Norte, Natal, Brasil
alba@lasic.ufrn.br, ivan@dimap.ufrn.br

ABSTRACT

This paper presents an architecture that uses a hardware/software methodology, with a NoC-based hardware, for the H.264 motion estimation. The H.264 is the newest video compression standard, which achieved its goal reducing in 50% the number of bits needed to represent the information. The proposed architecture was implemented in hardware description language VHDL and validated by simulations and FPGA (Field Programmable Gate Array) prototyping. Silicon costs, frequency and the communication with the external memory were evaluated in order to verify the needs of improvement to achieve the requirements adopted by Brazilian Digital Television System.

1. INTRODUCTION

Nowadays a huge number of electronic devices are capable of controlling digital videos. The video compression becomes essential to make possible the storage and mainly the transmission of these data.

A video is formed by a sequence of images, called frames, which use to present great similarity. This similarity is known as temporal redundancy [1].

The Motion Estimation (ME) is the most compute intensive module of a video encoder [2]. This module takes advantage of the temporal redundancy in a video sequence to reduce the number of bits needed to represent the total information.

The H.264 [3] is the newest video compression standard, which achieved its goal reducing in 50% the number of bits needed to represent the information. This gain was a consequence of a high level increase in computational complexity of its modules, mainly the ME.

The architecture proposed here uses a hardware/software methodology, with a NoC-based hardware. The most time consuming part of the motion estimation is done in hardware and the others stages are made by software, providing a great flexibility.

This paper is organized as follow: the section 2 presents a short description of ME and the improvements proposed for this module in H.264 standard. In the section 3 is presented the proposed architecture. The section 4 presents the execution flow. The next one shows the results of synthesis and prototyping of the proposed architecture. Finally, in section 5 there is a conclusion and future work discussion.

2. MOTION ESTIMATION

Each frame in a video sequence is segmented into fixed non-overlapping square blocks. A video sequence can be constructed by using differences between blocks of adjacent frames. These differences are represented by motion vectors that indicate the relative move of a block from a frame to another.

The ME can be divided in two stages. The block matching stage consists in use an algorithm that moves a block of a current frame in a search area from a reference frame; and the function of similarity, which measures how similar blocks of adjacent frames are.

Numerous block matching algorithms were proposed to balance performance and accuracy. The Full Search (FS) [4] is an example of algorithm that prefers accuracy than performance. It is called optimal algorithm. Algorithms, like Diamond Search (DS) [5], that proves a great performance and an acceptable, but not exactly response, is called sub-optimal algorithm.

There are many functions used to measure the similarity between two blocks. The Sum of Absolute Differences (SAD) is a simple function that use just sums and subtractions to calculate the difference between two blocks. Because of its simplicity, that is the most used function of similarity on ME modules [6].

The H.264 standard proposes several improves to the ME module. One of them is related to the block size. Rather than exploiting a fixed block size like 16x16 or 8x8, H.264 supports the use of variable block size: 16x16, 16x8, 8x16, 8x8, 8x4, 4x8 and 4x4 [3]. A decision mode must be used to choice which block size is appropriate to each piece of image.

The architecture proposed in this paper uses the FS for block matching, SAD for similarity calculation and support variable block sizes.

3. PROPOSED ARCHITECTURE

The architecture, showed in the **Figure 1**, is formed by 8 SAD Calculation Units interconnected through a Network on Chip (NoC), a Nios II/f processor [7] and a memory, all interconnected by an Avalon bus [8] one of the NoC routers are connected to a component that implements a wrapper mechanism between the Avalon bus and the NoC.

The sub-sections bellow will present a short description of each architecture component.

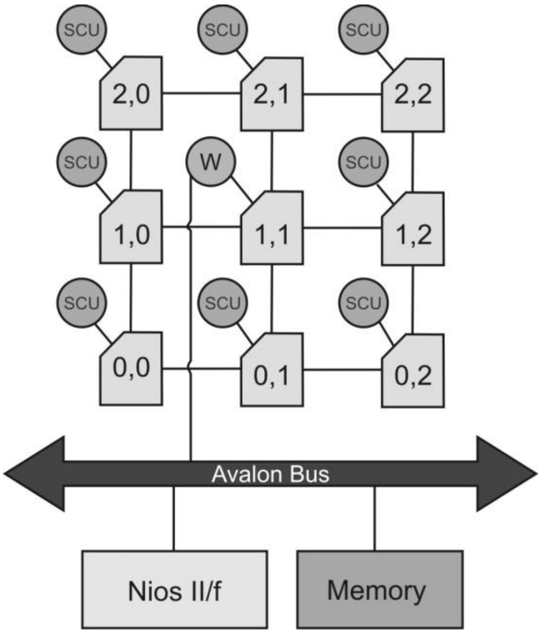

Figure 1 - Proposed Architecture

3.1. SAD Calculation Unit (SCU)

Each SAD Calculation Unit is responsible to provide the results of similarity calculation between a 16x16 macroblock and its correspondent search area. These units are coupled to the NoC routers and are composed by four modules, like showed on **Figure 2**.

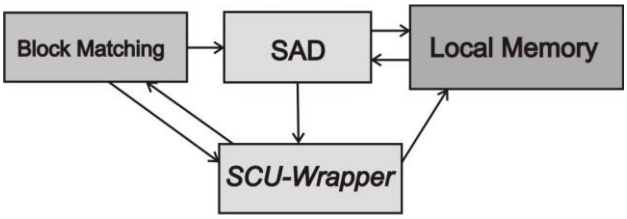

Figure 2 - SAD Calculation Unit

The local memory stores information about a macroblock (16x16 pixels) and a 32x32 search area. In this work only the luminance component of the image pixels were considered. So, the total memory capacity is 1280 bytes: the first 1024 are dedicated to the search area and the last 256, to the macroblock.

The SAD module implements the Sum of Absolute Differences to 4x4 block size. This is the smallest size allowed by the H.264 standard. Given the result of a 4x4 block size it is possible to obtain the others block sizes values through the reuse of these values.

The Block Matching module carries the logic of the block matching algorithm. This architecture version implements the Full Search Algorithm. Once the search area is a 32x32, 289 matches are done for each macroblock. Future works include the use of Diamond Search Algorithm.

SCU-Wrapper is the module that deals with the NoC packages, sending and receiving data from processor.

3.2. Nios II Processor

The processor is the one that makes the frames reading and distributes the macroblocks to the SCUs; it reads the SAD result calculated by the hardware module and reuses them to generate the others block sizes values allowed by the H.264 standard. So, it generates the motion vectors for each macroblock.

In this work, Altera's Nios II/f [7] processor was used. Nios II is a RISC, soft-core and general purpose processor designer to attend a vast number of applications. It was chosen to this work because of its easier integration with Altera devices.

3.3. Processor Wrapper

To one of the NoC routers are connected a component that implements a wrapper mechanism between the Avalon bus and the NoC. This component packs and unpacks data in the NoC protocol which allows the data to circulate throw the network.

3.4. NoC SoCIN

In this work, the SCUs were integrated using the NoC SoCIN. This is a very well documented and recognized network that implements the ParIS router [9]. Once this NoC was used and tested by some others academic works [10, 11], SoCIN has been chosen to interconnect the SCUs, saving implementation time.

SoCIN has 2-D grid (or mesh) topology and wormhole packet-based switching. Others network characteristics can be modified changing a set of parameters. In this work, a NoC of 3x3 dimension was configured to use deterministic XY routing, handshake flow control and round robin arbitration.

This network has a defined package format. To this specific work, three types of packages were used. These packages are showed on **Figure 3**.

SoCIN reserves the most significant bit of each word to indicate the end of the package. The second most significant bit indicates the start of the package. The last 4 bits of the header are used to address of the destination node of the package.

Additional useful information was incorporated to the package header. Three bits are used to identify the type of the package and 4 bits were used to identify the source node of the package.

The package showed on **Figure 3** (a) contains the pixels of a 16x16 macroblock and its respective search area. This information comes from the processor, and is delivered to a SCU.

Figure 3 (b) presents the SAD result package. The sixteen 4x4 results of one block matching are sent to the processor that is responsible to reuse these values and to provide the other block size values: 4x8, 8x4, 8x8, 8x16, 16x8 and 16x16.

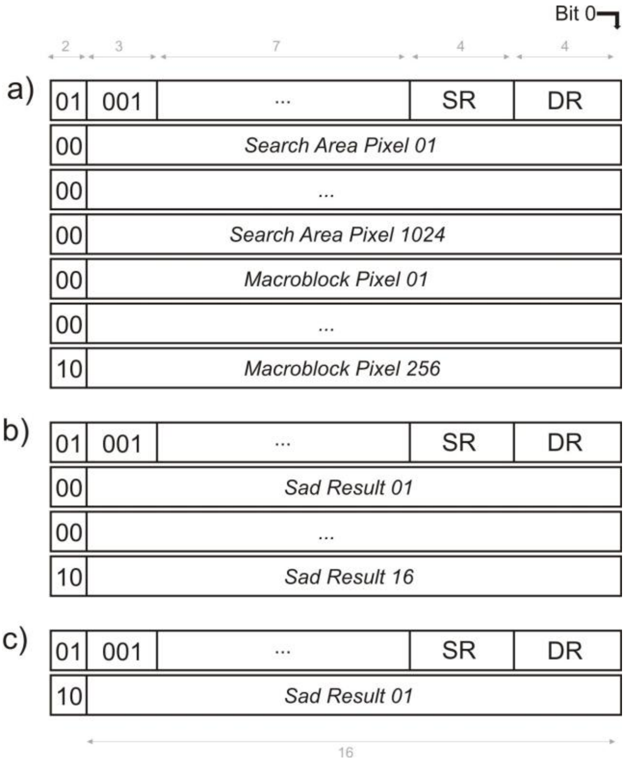

Figure 3 - The type of packages used by the architecture on the NoC SoCIN: **a)** The package used to send a macroblock and search area data; **b)** SAD result package; **c)** The execution end package.

The last package type, showed in **Figure 3** (b), is used to communicate the processor that the SCU has finished the block matching execution.

4. EXECUTION FLOW

The execution starts at the processor that reads the frames and segment the current frame in macroblocks. Then the processor sends a macroblock and its correspondent search area to each SCU through Avalon Bus. When the pixels arrive at the Processor Wrapper, it packs the information in the NoC pakage format and sends it to the destination node.

Once the package arrives at the SCU, the SCU-Wrapper unpacks the information and writes the pixels at the local memory. Then the block matching process is started. For each match, a package containing the sixteen 4x4 SAD results are sent to the processor that can reuse this values to generate the others block size modes. When the all matches are made, a package indicating the end of the block matching process is sent to the processor. So, while there are remaining unprocessed macroblocks, the processor sends a new macroblock to the SCU that has finished its execution and the process is restarted.

5. RESULTS

The architecture components presented in the section 3 was implemented in VHDL and integrated to the NoC SoCIN also described in VHDL. The complete architecture was synthesized and prototyped on a FPGA from Altera Cyclone II family. This FPGA is the EP2C35F672C6 presents at the Altera DE2 prototyping kit.

To VHDL description and synthesis was used the Altera Quartus II, version 8.1. The SOPC Builder, version 8.1, was used to create the system composed by architecture components, memory, Nios II processor and Avalon Bus.

A language C application was implemented to execute the software motion estimation parts at the processor. The Altera Nios II IDE compiled and injected the application into the system through the JTAG UART I/O component.

The maximum operating frequency of the system is 97.01 MHz. The **Table 1** bellow presents the results for silicon area obtained from the prototyping.

Component	Logic Elements
8 SCUs	3618
9 NoC Router	3402
Processor Wrapper	118
Nios II/f	2521
Performance Counter	1333
Total Area	**10992**

Table 1 - Silicon cost for each architecture component

The area unit used by Cyclone II family is the Logic Element (LE). This device has 33.216 so the architecture uses approximately 33% from the total of logic elements available.

The Performance Counter is an Altera's component and was used just to measure performance from parts of the application, so it can be removed without causing problems for the architecture.

Furthermore, 148864 memory bits from the device were used. It represents 33% of the total available and are used as the local memory of the SCUs and as some buffers used to store temporary data.

The **Figure 4** presents the percentage of time spent in some tasks of the architecture. It's possible to notice that 5% of the time is dedicated to the SAD calculation, the most costly task, realized in hardware. 36% of the time is used to transmit data between the hardware module and the processor through Avalon Bus. The remainder time is used by the processor to execute the software tasks described on section 3 and access the external memory (also by the Avalon Bus).

This results show that the Avalon Bus communication is a bottleneck to the total run time of the architecture.

This architecture is not yet capable to achieve real time required for Brazilian Digital Television System, but some improvements are been studied to make it possible. These improvements will be described on the next section as future works.

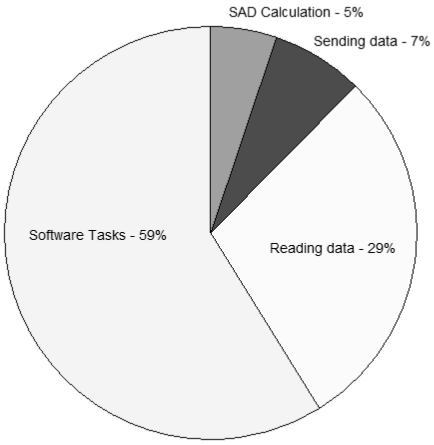

Figure 4 - Percentage of execution time tasks from the architecture

6. CONCLUSIONS AND FUTURE WORKS

In this paper was proposed an architecture that uses a hardware/software approach, with a NoC based hardware for the motion estimation. The architecture was implemented in VHDL, integrated with the NoC SoCIN, synthesized and prototyped on an Altera's FPGA chip.

The performance results showed that this architecture is still not able to achieve real time processing required for Brazilian Digital Television System. The results obtained showed that the Avalon Bus is a bottleneck to the data transfer. The tools used in this work require the use of Nios II/f processor with the Avalon Bus. The code of Nios II/f core is proprietary and no modification can be done to integrate another bus. So, another processor is been studied to be integrated to the architecture: Plasma is a simple processor that implements the MIPS architecture and its VHDL core code is available to distribution and modification [12].

Future works include the implementation of Diamond Search block matching algorithm. This algorithm has been studied from the past years and has shown great results on balancing performance and accuracy [13].

The internal parallelization of the SAD module from the SCU component must improve the total gain and will be done too.

Another future work is relative to the distribution of macroblocks between the SCUs. Currently, the next macroblock in the sequence is sent to the SCU that finishes its job and requests a new one. As adjacent macrobloks share search area data, each SCU must be responsible for a set of adjacent macroblocks avoiding resend data that already is in the local memory of another SCU.

10. REFERENCES

[1] I. E. Richardson, *H.264 and MPEG-4 Video Compression: Video Coding for Next-Generation Multimedia*. John Wiley and Sons: 2002.

[2] L. Deng, W. Gao, M. Z. Hu, and Z.Z. Ji, "An efficient hardware implementation for motion estimation of AVC standard", Consumer Electronics, IEEE Transactions on , vol.51, no.4, pp. 1360-1366, Nov. 2005

[3] ITU-T – International Telecommunication Union. ITU-T Recommendation H.264/AVC (05/03): advanced video coding for generic audiovisual services. In 2007.

[4] V. Bhaskaran, K. and Konstantinides. "Image and Video Compression Standars: Algorighms and Architectures". 2 ed. Kluwer Academic Publishers: Boston, 1997.

[5] X. Yi and N. Ling, "Rapid block-matching motion estimation using modified diamond search algorithm", Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on, vol., no., pp. 5489-5492 Vol. 6, 23-26 May 2005.

[6] J. Vanne, E. Aho, T.D. Hamalainen and K. Kuusilinna, "A High-Performance Sum of Absolute Difference Implementation for Motion Estimation", Circuits and Systems for Video Technology, IEEE Transactions on, vol.16, no.7, pp.876-883, July 2006.

[7] Altera Corporation, "Nios II Processor Reference Handbook", http://www.altera.com, 2009.

[8] Altera Corporation, "Avalon Interface Specifications", http://www.altera.com, 2009.

[9] C. A. Zeferino, F. G. M. E. Santo and A .A. Susin, "ParIS: A Parameterizable Interconnect Switch for Networks-on-Chip". In: 17th Symposium on Integrated Circuits and Systems Design (SBCCI), 2004, Porto de Galinhas. Proceedings. New York: ACM Press, 2004. p. 204-209.

[10] M. B. Costa and I. S. Silva, "A Core for Network-on-Chip Latency-Based Performance Analysis". In: 8th Microeletronic Students Forum (SForum), 2008, Gramado.

[11] H. C. Freitas, P. O. A. Navaux, and T.G.S. Santos, "NOC architecture design for multi-cluster chips", Field Programmable Logic and Applications, 2008. FPL 2008. International Conference on, vol., no., pp.53-58, 8-10 Sept. 2008

[12] S. Rhoads. "Plasma CPU Core". Disponível em: http://www.open-cores.org?do=project&who=plasma, 2009

[13] M. Porto, L. Agostini, S. Bampi, A. Susin, "A high throughput and low cost diamond search architecture for HDTV motion estimation", Multimedia and Expo, 2008 IEEE International Conference on , vol., no., pp.1033-1036, June 23 2008-April 26 2008.