International Journal of
Science and Engineering Investigations vol. 7, issue 80, September 2018
ISSN: 2251-8843

Embedded Architecture of the ADTF Algorithm Based on Low Cost Systems: Towards an Implementation OpenCL Architecture

Rachid Latif[1], Wissam Jenkal[2], Amine Saddik[3]
[1,2]Laboratory of Systems Engineering and Information Technology, ENSA, Ibn Zohr University, Agadir, Morocco
[3]SEIB Master, ENSA, Ibn Zohr University, Agadir, Morocco
([1]r.latif@uiz.ac.ma, [2]Wissamjenkal@gmail.com, [3]A.saddik.ma@gmail.com)

Abstract- The purpose of this work is to implement the Adaptive Dual Threshold Filter (ADTF), based on Low Cost systems, on OpenCL architecture. The ADTF implementation is an inexpensive filtering algorithm of the ECG signal, whether in terms of complexity or processing time, which makes the selected algorithm one of the best signal filtering techniques. The ADTF implementation, developed by our team, is described by the structural method of VHSIC (Very High Speed Integrated Circuit) Hardware Description Language (VHDL). This ADTF architecture is implemented with functional block by functional block, which may have given us an opportunity to implement this architecture in a heterogeneous system based on the OpenCL parallel programming.

The ADTF architecture proposed by our team is generally separated into 3 functional blocks which give a remarkable visibility to the architecture. In this work, we propose the embarkation of the same architecture proposed by our team, in an Intel-Altera heterogeneous architecture based on the OpenCL parallel programming. Thanks to the high power of its system either in arithmetic calculation or data processing and the most important thing is optimization such us time execution and logic blocks resource.

Keywords- *Heterogeneous System, Architecture OpenCL, ADTF Algorithm, FPGA, Low-cost systems, VHDL*

I. INTRODUCTION

The interest of the embedded system in our day is an irreplaceable need in all the field of our life. For example in the biomedical field, we have several application of the embedded systems. In our case, the approach filtering of electrocardiogram signal (ECG). This signal is the basis of the diagnosing heart disease. The ECG signals graphically presents the electrical activity of the heart. These activities are dedicated to the electric potential variations of the specialized cells in the contraction [1-3]. It is collected by electrodes on the surface of the skin. The signal collected from these electrodes contains a lot of information, usually P and T waves and the QRS complex. The P wave has the prior depolarization of the sinus node, the T wave is the electrical signal that corresponds to the rapid repolarization phase of the two ventricles and the QRS complex presents the ECG electrical signal that reflects bi-ventricular depolarization.

The ADTF technique developed by Wissam JENKAL in [4] is a filtering algorithm inspired by the recently published Median Double Threshold method [5], which offers the best image filtering results. In the case of double threshold method, the noisy pixels are identified in a relatively narrow range and can therefore reduce the probability of detection errors.

FPGAs are reconfigurable VLSI components, the interest of this re-configurability is to integrate a variety of data monitoring and analysis processes. In this work, the use data are recovered from the Electrocardiography instrument. FPGA based architecture is designed using High Level Synthesis tools (HLS) [6] or Hardware Description Language (HDL) such as VHDL [7-9] and Verilog [10].

The interest of this work is to propose an implementation of the ADTF algorithm in a DE1-FPGA Soc board. This card is cyclone V type and has both architectures a FPGA card and an ARM Cortex A9 processor. The DE1 Soc board contains a heterogeneous system that programmed by the HLS tools such as OpenCL. For the FPGA part alone, we can use the VHDL description language. This board is offered by Intel-Altera Company.

OpenCL is a new standard industry for tasks and parallel data, for heterogeneous computing on a variety of processors, CPUs, GPUs, DSPs and the FPGA card [11].

In this article, we propose OpenCL tool for the implementation of the ADTF algorithm proposed by Wissam JENKAL [4], thanks to the total optimization of the resources used.

II. THE ADTF ALGORITHM

In computer science, we speak of a real-time system when this system is able of controlling (or piloting) a physical process at a speed adapted to the evolution of the controlled process. The real-time systems are different from the other systems at the level of taking into consideration of time requirement. In this case the respect of delays is more important than the accuracy of the result, for that our team proposed the ADTF architecture in real time [12]. The ADTF algorithm is based on the calculation of data in the selection window. It is necessary to calculate the higher threshold, the lower threshold and the average of this window parameters [13-14], for each selection window of input signal ECG.

The equation of the average of a window is:

$$Ö = \frac{1}{\mu} \sum_{i=n}^{n+\mu} \rho(i) \quad (1)$$

ρ(i) is the noisy input, μ is the window size and Ö is the average of the selected window.

The equation of the lower threshold is given by the relation [4]:

$$£_L = Ö - [(Ö - \omega_{min}) \times \beta] \quad (2)$$

ω_{min} is the minimum value of the selected window, $£_L$ is the lower threshold of the selected window and β is the thresholding coefficient.

The equation of the higher threshold is as follows [4]:

$$£_H = Ö + [(\omega_{max} - Ö) \times \beta] \quad (3)$$

ω_{max} is the maximum value of the selected window, $£_H$ is the higher threshold of the selected window andβ is the narrow range coefficient with [4]:

$$0<\beta<1 \quad (4)$$

The role of coefficient β is adapted to the threshold of our proposed filter to provide adaptive filtering of ECG signal, β generally varies between 0% and 100% is to say that the value of β is between 0 and 1. The article [15] show the importance of this variation and the different results obtained. The low coefficient of β values are needed for low set SNR (Signal-to-Noise Ratio) input level. For high-set level of SNR input, a larger toleration is required, i.e. higher values of coefficient are recommended.

The key point in the ADTF algorithm proposed by our team is the size of the window, because with this size, it is possible to compare the median value with the size of the region window (*in the left and the right*). After the calculation of the parameters, which characterizes the ADTF algorithm, the higher threshold, the lower threshold and the average of this window, take the window size [i; i +μ] (*with i is the variation of signal window*). We interpret the value of $\rho(i+\frac{\mu}{2})$ and apply the following conditions [4]:

- If $\rho(i+\mu/2) > £_H \Rightarrow \Omega(i+\mu/2) = £_H$ (5)
- If $\rho(i+\mu/2) > £_L \Rightarrow \Omega(i+\mu/2) = £_L$ (6)
- If $£_L < \rho(i+\mu/2) < £_H \Rightarrow \Omega(i+\mu/2) = \rho(i+\mu/2)$ (7)

So the global algorithm of the ADTF filter is:

- The first step is to give i the initial value which is 1 thereafter loaded the signal ρ(i) in the input.
- The second step is the calculation of the thresholds and the average of a window.
- And lastly the application of conditions (5), (6) and (7).

After the processing and execution of the adaptive filter algorithm it remains the part of the test application. To test the performance of the ADTF algorithm, we have used the international data base MIT-BIH Physionet. The figure 2 shows the ECG signal filtering results with a WGN level of 20 dB. Figure 1 shows the noisy signal filtering results by White Gaussian Noise (WGN) by applying the ADTF filtering while noticing the high quality and performance after the filtering. In this case a 20 dB WGN was chosen with the MIT-BIH signal 100 which is a database that contains a lot of signals for the test.

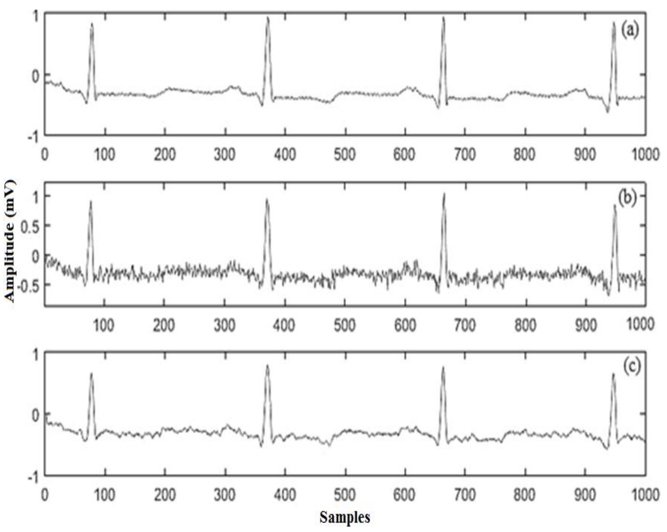


Figure 1. a) Original signal, b) signal convolved with the noise, c) signal result corrected with ADTF [15]

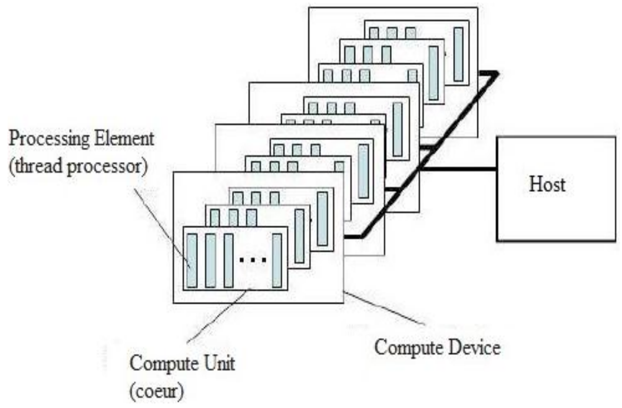


Figure 2. The OpenCL architecture

III. THE PARALLEL PROGRAMMING

The parallelism consists in implementing digital electronics architectures, which make it possible to process information simultaneously, as well as specialized algorithms for them. These techniques are intended to achieve the largest number of operations in the smallest possible time. The notion of parallelism is based on the competition of instruction and data flows:

- SIMD: Single Instruction Multiple Data. A programming model where a kernel is executed concurrently on multiple processing elements each with its own data and a shared program counter. All processing elements execute a strictly identical set of instructions.
- SPMD: Single Program Multiple Data. A programming model where a kernel is executed concurrently on multiple processing elements each with its own data and its own program counter. Hence, while all computational resources run the same kernel they maintain their own instruction counter and due to branches in a kernel, the actual sequence of instructions can be quite different across the set of processing elements.
- TPPM: Task Parallel Programming Model. A programming model in which computations are expressed in terms of multiple concurrent tasks where a task is a kernel executing in a single work-group of size one. The concurrent tasks can be running different kernels.
- MISD: Multiple Instruction, Single Data stream. Several instruction streams operate on the same data stream.
- MIMD: Multiple Instruction Multiple Data streams. Multiple Calculation Units Operate Multiple Data Flows Using Multiple Instruction Flows.

A. The heterogeneous system:

The strong need for increased computing performance in science and engineering has led to the use of heterogeneous computing, with GPUs and other accelerators acting as co-processors for intensive parallel data arithmetic workloads [16] .

The trend towards heterogeneous computing and highly parallel architectures has created a significant need for software development infrastructure in the form of parallel programming languages and subprogram libraries supporting heterogeneous computing on platforms. Material produced by several suppliers. Many existing scientific and technical applications have been adapted to efficiently utilize multi-core processors and massively parallel GPUs using tools such as the Threading (TBB), OpenMP, CUDA [17], and more. Existing programming tools were limited to one family of microprocessors or did not support heterogeneous computing, in general for the heterogeneous architecture we find, CPU-GPU, CPU-FPGA, CPU-DSP.... In our case we will discuss only the CPUs with GPUs or FPGAs.

Usually, the language used in parallel computing is the OpenCL, but there are other types like:

- **CUDA** (Compute Unified Device Architecture) is a GPGPU (General-Purpose Computing on Graphics Processing Units) technology, that is to say using a graphics processor (GPU) to perform general calculations in place of the processor (CPU). Indeed, these processors commonly comprise of the order of a thousand calculation circuits typically operating at 1 GHz, which represents a much higher potential than a central processor at 4 GHz, even if multicore and multi-threaded, if and only if the calculation to be performed is parallelizable [18].
- **OpenCL** (Open Computing Language) is an open royalty-free standard for general purpose parallel programming across CPUs, GPUs, FPGA and other processors, giving software developer's portable and efficient access to the power of these heterogeneous processing platforms. OpenCL supports a wide range of applications, ranging from embedded and consumer software to HPC solutions, through a low-level, high-performance, portable abstraction. By creating an efficient, close-to-the-metal programming interface, OpenCL will form the foundation layer of a parallel computing ecosystem of platform-independent tools, middleware and applications. OpenCL is particularly suited to play an increasingly significant role in emerging interactive graphics applications that combine general parallel compute algorithms with graphics rendering pipelines [18].
- **OpenMP** (Open Multi-Processing) is API (Application Programming Interface) for programming parallel applications on shared memory architectures. OpenMP is portable and scalable. It makes it possible to rapidly develop parallel applications with small granularity while remaining close to the sequential code.
- **OpenACC** is API (Application Programming Interface) and is a collection of compiler directives and runtime routines that allow you, the programmer, to specify loops and regions of code in standard C, C++ and Fortran that you want offloaded from a host CPU to an attached accelerator, such as a GPU. The OpenACC is a parallel programming standard describing a set of compiler directives which can be applied to standard C, C++, and Fortran to specify regions of code for offloading from a host CPU to an attached accelerator.

In our case study, we will focus on the heterogeneous programming language specifically OpenCL for several reasons cited and justified in the following sections.

IV. THE PROPOSED APPROACH TO IMPLEMENTING THE ADTF ALGORITHM VIA OPENCL

A. The heterogeneous architecture

The Heterogeneous System Architecture (HSA) is a set of inter-vendor specifications that allow the integration of central processors and graphics processors on the same bus, with shared memory and tasks. HSA is developed by the HSA Foundation, which includes (among others) AMD and ARM. The declared purpose of the platform is to reduce the communication latency between processors, graphics processors and other computing devices, and to make these different peripherals more compatible from the programmer's point of view, relieving the scheduler of the task of planning the movement of data between disjoint memories of devices (as it should be done now with OpenCL or CUDA).

The following figure shows the global architecture of OpenCL.

As you see in Figure 2 the heterogeneous architecture, it is separated in 2 part, a Host which is the CPU (Central Processing Unit), equipped with many arithmetic execution units. The device can be a GPU, FPGA or other.

Figure 3 shows the heterogeneous architecture CPU-GPU. It is contains the different elements mentioned below:

- **Private memory** of the Device: Each work-item has a private memory that is the fastest access, without synchronization primitive, and is allocated/partitioned at compilation for the kernel and the card: its size is unknown.
- **Local memory** of the Device: Memory shared by a whole workgroup. Each workitem of the same workgroup can access it. Generally on ship.
- **Global memory** of the Device: Accessible by all workgroups and is unsynchronized, possibly constant (i.e. read-only): largest memory.
- **Global memory** of the Host: The slowest but most consistent access. Memory to avoid (important loading time: it is better to keep the maximum possible, as long as possible on the GPU).
- **Work-item:** Equivalent of a thread. The execution of a kernel is done on a set of work-items whose number is parameterized by the developer. Each work-item executes the same code. Each work-item is identified by ID.
- **Work-group**: Set of work-items cooperating together within a Work-group. Reflects the organization of work-items in grid of 1, 2, 3 dimensions and are identified by an ID.

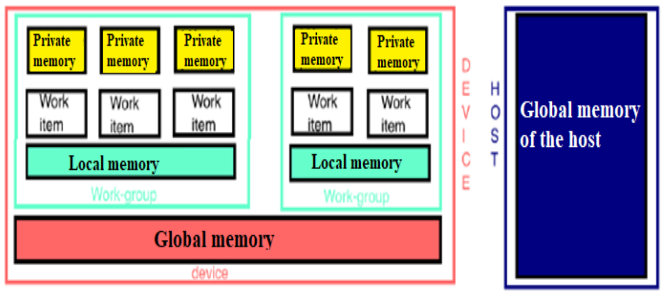


Figure 3. The CPU-GPU architecture

There is also the CPU-FPGA architecture. Figure 4 shows the heterogeneous architecture CPU-FPGA.

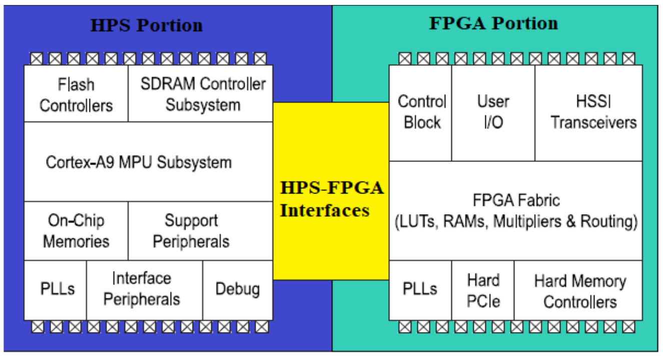
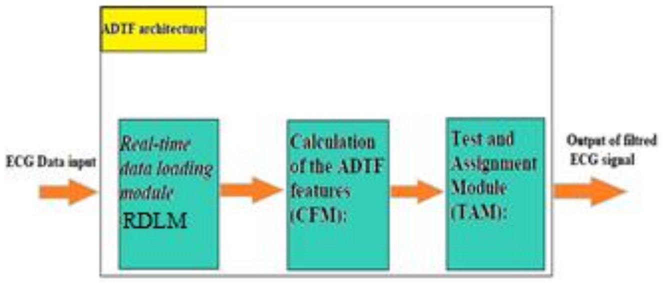


Figure 4. The CPU-FPGA architecture

B. *Towards a hetegenous implementation of ADTF filter*

In this part, we will discuss the possibility of architecture implementation proposed by W. Jenkal et all in [13]. This architecture is separated into 3 functional blocks. The first block is Real-time Data Loading Module (RDLM). This block is dedicated to the preparation of ECG signal emitted by the electrocardiogram. The second block is Calculation of the ADTF Features Module (CFM). This module permits the determination of the window averaging, higher threshold and lower threshold. The third block is Test and Assignment Module (TAM). This block is reserved for the test and output assignment. Figure 5 shows the architecture of the ADTF algorithm:



ADTF architecture
ECG Data input
Real-time data loading module RDLM
Calculation of the ADTF features (CFM):
Test and Assignment Module (TAM):
Output of filtred ECG signal

Figure 5. The ADTF architecture

In this figure, the architecture is separated to functional blocks, which makes the implementation in OpenCL easy. The proposed card for the OpenCL implementation of the ADTF architecture is DE1 FPGA Soc board.

Figure 6 shows the DE1 board of the Intel manufacturer Intel-Altera.

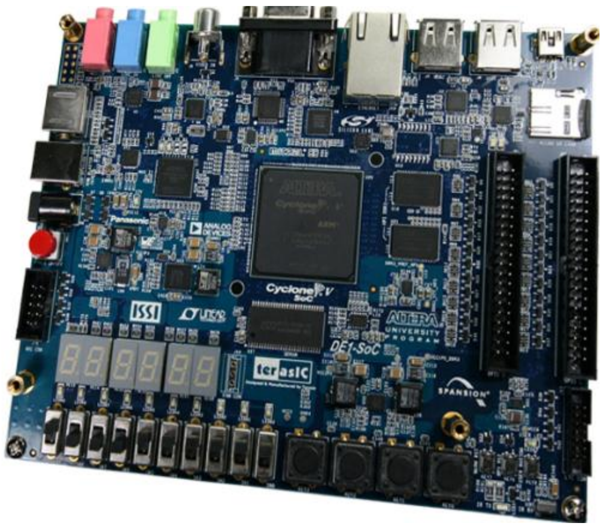


Figure 6. The DE1-Soc by Intel-Altera

The DE1-SoC introduces a robust hardware design platform built around Altera's System-on-Chip (SoC) FPGA, which combines the latest Cortex-A9 dual-core embedded cores with state-of-the-art programmable logic for design flexibility ultimate. Users can now take advantage of the exceptional redundancy associated with a high-performance, low-power processor system. Altera SoC integrates an ARM-based hard processor (HPS) system consisting of processors, peripherals and memory interfaces transparently linked to the FPGA structure using a high-bandwidth interconnect backbone. The DE1-SoC development board is equipped with high-speed DDR3 memory, audio and video capabilities, Ethernet networks and many other exciting applications.

The DE1 board offers 64 MB (32Mx16) SDRAM, 1 GB (2x256Mx16) DDR3 SDRAM on HPS, Dual-core ARM CORTEX-A9 (HPS), and an MICRO SD Card Socket on HPS. 10 User switches (FPGA x10). 11 User LEDs (FPGA x10; HPS x1) are integrated into the DE1 soc board. This board offers 3 oscillators of 50 MHz, 27 MHz and 24 MHz for clock sources. This board offers other tools like VGA output and RS-232 connector ... etc. In addition to these hardware features, the DE1-Soc board has software support for standard I/O interfaces and a control panel to access to the various components of this board. In addition, software is provided for a number of demonstrations that illustrate the different features of the DE1-Soc board. The figure 7 shows the Block Diagram of the DE1-SoC Board.

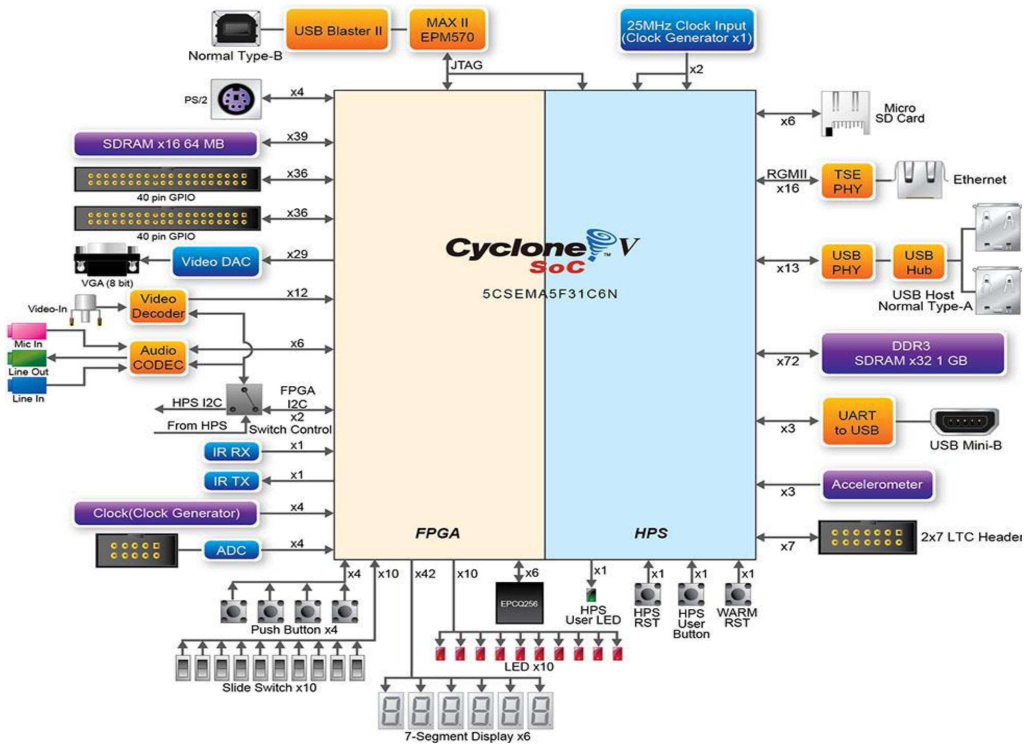


Figure 7. The Block Diagram of the DE1-SoC Board

The present study, proposes a new approach of the heterogeneous implementation of the ADTF filter proposed by Wissam Jenkal in [13]. In this approach, we proposed the ADTF heterogeneous implementation using parallel programming via OpenCL tool.

As you can see in Figure 5, the ADTF architecture is separated into 3 blocks. Then our new proposal is to separate the block so that a part will be processed in the Host (ARM) and the rest in the Device (FPGA). Consequently the implementation of the ADTF algorithm in this new architecture will be optimized, in the resources and also the processing time.

However this approach permits to give flexibility and a possibility of realized of other task in parallel with itself treatment.

V. Results and Discussion

The test of ADTF architecture on the DE1 card developed by Wissam Jenkal in [13], is used to validate the architecture given in VHDL. This test made a comparison between the material (DE1 card) and the simulation tool (ModelSim) results. The general purpose of this validation is to verify the different functionality of the architecture and the synchronization with the different FPGA card elements. The following figure shows a simulation with ModelSim proposed by our team in [13].

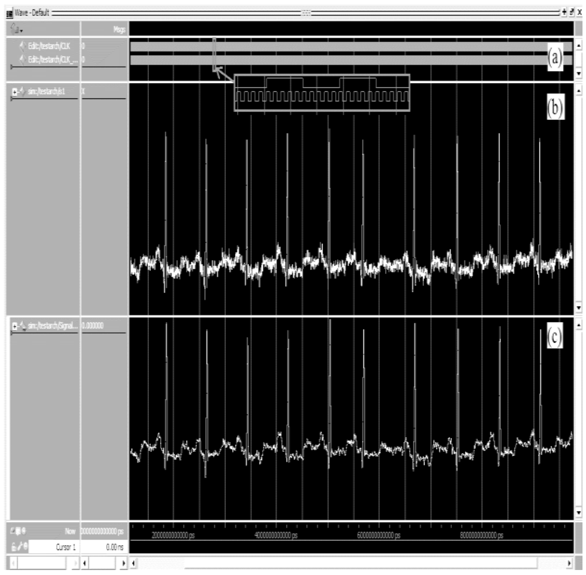


Figure 8. ECG signal denoising using the real-time ADTF architecture via ModelSim: Case of signal 100 with 20 dB of WGN. (a) Clock signals (Clk and Internal Clk), (b) infected signal, (c) signal result corrected with ADTF

The figure 9 shows our proposed architecture for ADTF implementation in the heterogeneous DE1-Soc FPGA card. As you see in the new proposed architecture, the first block RDLM which has a role of emission of the signals is implemented in the part Device (FPGA), then the block of calculates thresholds CFM in the part Host (HPS) which contains ARM cortex-A9. The question that's why we chose the second block specifically for the built-in HPS part? Thanks to the high power of arithmetic calculation in CPUs and the need for arithmetic calculation in this block. We have the possibility of integrating it into the HPS part. The third block for the tests will be implemented in the Device part (FPGA).

VI. Conclusion

In this work, we have proposed a new approach of the heterogeneous implementation of the ADTF algorithm developed by our team in 2015. In this approach, we proposed the ADTF heterogeneous implementation using a parallel programming via OpenCL tool in the DE1 Soc card of the Intel manufacturer Intel-Altera. This heterogeneous architecture will offer an optimization in the resources of our card, even if the VHDL description language, and above all the Structural method is optimized, but the proposal remains the best at the level of resource optimization and processing time.

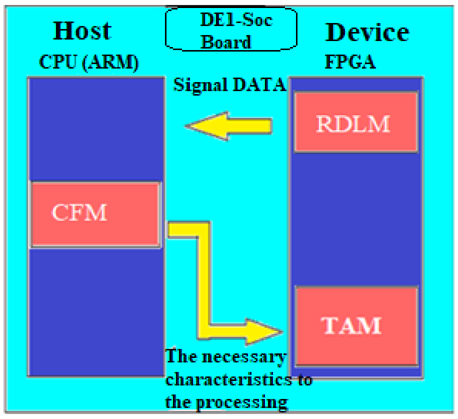


Figure 9. The new optimized architecture of the ADTF algorithm

References

[1] Venes D. Taber's cyclopedic medical dictionary. FA Davis;2013. p. 1086–7.

[2] Betts JG, DeSaix P, Johnson E, Johnson JE, Korol O, Kruse DH, et al. Anatomy and physiology. OpenStax College; 2013.

[3] Jenkal W, Latif R, Toumanari A, Dliou A, El B'charri O, Maoulainine FMR. QRS detection based on an advanced multilevel algorithm. Int J Adv Comput Sci Appl 2016;7 (1):253–60

[4] W. Jenkal, R. Latif, A. Toumanari, A. Dliou, O. El B'charri, F. M. Maoulainine, An efficient algorithm of ECG signal denoising using the adaptive dual threshold filter and the discrete wavelet transform, Biocybernetics and Biomedical Engineering, vol. 36 n. 3, 2016, pp. 499-508.

[5] Gupta V, Chaurasia V, Shandilya M. Random-valued impulse noise removal using adaptive dual threshold

[6] D. O'Loughlin, A. Coffey, F. Callaly, D. Lyons, F. Morgan, Xilinx Vivado high level synthesis: Case studies, 25th IET Irish Signals & Systems Conference 2014 and 2014 China-Ireland International Conference on Information and Communications Technologies (ISSC 2014/CIICT 2014), 2014, pp. 352-356.

[7] P. Eles, K., Kuchcinski, Z. Peng, *System synthesis with VHDL* (Springer Science & Business Media, 2013).

[8] C. H. Roth Jr, L. K. John, *Digital systems design using VHDL* (Cengage Learning, 2016).

[9] F. Vahid, Digital Design with RTL Design, Verilog and VHDL, (John Wiley & Sons, 2010).

[10] P. Wilson, Design recipes for FPGAs: Using Verilog and VHDL, (Newnes, 2015).

[11] Munshi Aaftab. OpenCL Specification Version 1.0. Dec, 2008. http://www.khronos.org/registry/cl/

[12] wissam jenkal,rachid latif, ahmed toumanari, abdelhafid elouardi anas hatim, oussama el'bcharri, real-time hardware architecture of the adaptive dual threshold filter based ecg signal denoising ,july 2018. vol.96. no 14

[13] W. Jenkal, R. Latif, A. Toumanari, A. Dliou, O. El B'charri, An efficient method of ECG signals denoising based on an adaptive algorithm using mean filter and an adaptive dual threshold filter, *International Review on Computers and Software (IRECOS), vol. 10 n. 11,* 2015, pp. 1089-1095.

[14] Gupta, V. Chaurasia, M. Shandilya, Random-valued impulse noise removal using adaptive dual threshold median filter, *Journal of visual communication and image representation, vol. 26,* 2015, pp. 296-304.

[15] Jenkal W, Latif R, Toumanari A, Dliou A, El Bcharri O, Bsiss MA. QRS Detection Using an Effcient Algorithm Based on Wavelet Coefcients. International Review on Computers and Software (IRECOS). 2016; 11(6):479-488

[16] Cohen Jonathan, Garland Michael. Solving computational problems with GPU computing. Computing in Science and Engineering. 2009;11(5):58–63.

[17] Nickolls John, Buck Ian, Garland Michael, Skadron Kevin. Scalable parallel programming with CUDA. ACM Queue. 2008;6(2):40–53

[18] John E. Stone, David Gohara, and Guochun Shi, OpenCL: A Parallel Programming Standard for Heterogeneous Computing Systems,2010 May; 12(3): 66–72.

Rachid Latif received the PhD degree in signal processing in 2000, the Habilitation degree in 2005 from faculty of sciences and Full Professor in Information Technology in 2015, from Ibn Zohr University, Morocco. Currently, he is a Professor with the Department of Industrial & Electrical Engineering, National School of Applied Sciences, Ibn Zohr University, Agadir, Morocco. His research interests include Biomedical signal and image processing, Biomedical instrumentation, Robotics and embedded systems (CPU, OMAP, DSP, FPGA, GPU, CPU+FPGA), Artificial intelligence and information technology and he is working on the modeling, filtering, and analysis of fetal cardiac signals. From 2006 to 2014, he was the head of the signals systems and computer science group (ESSI). He was the head of the LISTI Laboratory, which he founded in December 2014 and the head of Embedded Systems and Biomedical Engineering Master (SEIB).And also he is evaluator expert CNRST Morocco.

Wissam Jenkal was born in Agadir, Morocco in 1991. He received the Master degree in automatic, signal processing and industrial computer from the University of Hassan the First, Faculty of Science and Technology, Settat, Morocco, in 2014. He joined the Laboratory of Systems Engineering and Information Te chnology (LISTI) in the National School ofApplied Sciences, Ibn Zohr University, Agadir, Morocco as a PhD Student in 2015. In 2018, he received the PhD degree in Embedded Systems and Signal Processing. His current research interests include Biomedical Engineering, Embedded systems and Information Technologies.

Amine SADDIK was born in Morocco in 1996. He graduated with a degree in instrumentation and biomedical maintenance, from ESTS at the university of Mohamed V Morocco, in 2017. Currently, he follows studies in 2nd yeara od Master Embedded Systems and Biomedical Engineering at the National School of Applied Sciences, Ibn Zohr University, Agadir, Morocco. His current research interests include biomedical engineering, embedded systems and artificial intelligence.