Speeding up an Adaptive Filter based ECG Signal Pre-processing on Embedded Architectures

Safa Mejhoudi[1], Rachid Latif[2]
Amine Saddik[3], Wissam Jenkal[4]
Laboratory of Systems Engineering and Information Technology, ENSA, Ibn Zohr University, Agadir, Morocco

Abdelhafid El Ouardi[5]
SATIE, Digiteo Labs
Paris-Saclay University
Orsay, France

***Abstract*—Medical applications increasingly require complex calculations with constraints of accelerated processing time. These applications are therefore oriented towards the integration of high-performance embedded architectures. In this context, the detection of cardiac abnormalities is a task that remains a high priority in emergency medicine. ECG analysis is a complex task that requires significant computing time since a large amount of information must be analyzed in parallel with high frequencies. Real-time processing is the biggest challenge for researchers, when talking about applications that require time constraints like that of cardiac activity monitoring. This work evaluates the Adaptive Dual Threshold Filter (ADTF) algorithm dedicated to ECG signal filtering using various embedded architectures: A Raspberry 3B+ and Odroid XU4. The implementation has been based on C/C++ and OpenMP to exploit the parallelism in the used architectures. The evaluation was validated using several ECG signals proposed in MIT-BIH Arrhythmia database with a sampling frequency of 360 Hz. Based on an algorithmic complexity study and a parallelization of the functional blocks which present significant workloads, the evaluation results show a mean execution time of 7.5 ms on the Raspberry 3B+ and 0.34 ms on the Odroid XU4. With an efficient parallelization on the Odroid XU4 architecture, real-time performance can be achieved.**

Keywords*—*ECG signal denoising; ADTF algorithm; OpenMP programming; embedded architectures

I. Introduction

ECG is an essential element in the diagnosis and the detection of cardiovascular disease or also in the monitoring of patients [1]. However, it is often correlated with different types of noise, which generates a distortion of the signal and a loss of valuable information. To simplify the interpretation task, several processing and filtering algorithms are proposed in the literature [2-6]. Digital Filters (FIR and IIR), Empirical Mode Decomposition (EMD), Wavelet Transform denoising techniques as Discrete Wavelet Transform (DWT) and the Adaptive Dual Threshold Filter (ADTF) [7-8].

Using digital filters, some useful information in the signal can be affected, particularly the R wave [3,9]. The EMD and the DWT give satisfying results, but they are characterized by their algorithmic complexity that requires more hardware resources and important computation time [10-11]. The ADTF proposed by W. Jenkal et al. in [7] has a great denoising capacity, especially when the signal is mixed with the high-frequency noises. The advantage of this technique is the very low complexity. Hardware implementation of this algorithm on FPGAs is presented in [8] using Hardware Description Language (VHDL). The author divided the algorithm into 3 main blocks: The first consists of Real-time data loading module with an acquisition frequency of 360 Hz as he uses signals from MIT-BIH database. The second is used for ADTF features calculation. The third is for Test and Assignment. To respect real-time processing, a frequency of 3.6 kHz is used in the second and third blocks, which is ten times greater than the acquisition frequency. This implementation indeed respects real-time, but this is related to the processing frequency which is fixed by the author not given by the architecture.

Abbreviations

***ADTF*:** Adaptive Dual Threshold Filter
***DWT*:** Discrete Wavelet Transform
***DT-WT*:** Dual-Tree Wavelet Transform
***ECG* :** Electrocardiogram
***EMD* :** Empirical Mode Decomposition
***EEMD* :** Ensemble Empirical Mode Decomposition
***EEMD-GA*:** EEMD and Genetic Algorithm
FIR: Finite Impulse Response
***FPGA*:** Field Programmable Gate Array
***GPU*:** Graphics Processing Unit
***Ht*:** Higher Threshold
***IIR*:** Infinite Impulse Response
***IMF* :** Intrinsic Mode Functions
***LA* :** Left Atria
***Lt*:** Lower Threshold
***LV*:** Left Ventricle
***MSE*:** Mean Square Errors
***PRD*:** Percentage Root-mean-square Difference parameter
***RA*:** Right Atria
***RV*:** Right ventricle
***SNR*:** Signal to Noise Ratio
***VHDL*:** VHSIC Hardware Description Language
***WGN*:** White Gaussian Noise

This work deals particularly with the real-time ECG denoising. Real-time systems differ from others by considering time constraints and compliance, which is as important as the result's accuracy. In other words, the system should not just deliver exact results; it should also provide them within a set timeframe. This notion takes on its importance when it comes to human health. So, the design of processing algorithms that meet these constraints is therefore essential. In this context, the processing system is considered a real-time system. It must make the necessary correction for sample *n* before the arrival of sample *(n + 1)*. This constraint implies rigorous requirements in terms of performance and speed. In this work, different signals from the Physionet MIT-BIH arrhythmia database are used, with 360 Hz of sampling frequency to test the reliability of the algorithm. This means that the time constraint is of the order of 2.77 ms. The Matlab simulation gives a processing time average of 150ms for one sample, which is too far from being in real-time.

In order to ensure real-time processing, OpenMP parallel programming is used, which gives excellent results compared to C/C++ naive implementation and Matlab implementation. So, for Desktop with OpenMP programming, just 0.34 ms is needed to process one sample. But desktop is not always a good solution in biomedical monitoring because of its size, weight, and power consumption, especially when the objective is the home monitoring to facilitate access to care for the elderly or in regions lacking medical personnel, prevent hospitalizations and improve patient control and quality of life.

We opted to use Low-cost embedded architectures such as XU4 and Raspberry. The given results show that the optimal choice is the XU4 board with an average processing time of 2.34ms instead of 7.5 ms in the case of Raspberry 3B, which greatly satisfies the time and energy consumption constraints.

The paper is formulated as follow:

1) The first section exposes an overview of ECG signal denoising techniques and related work.

2) The second section depicts a detailed evaluation of the ADTF algorithm and describes its implementation.

3) The third section puts on view the results and discussion of the processing performance of the embedded implementations on different embedded architectures.

4) Lastly, conclusion and future work are the objects of the fourth section.

II. ECG Signal Processing: An Overview

A. ECG Signal Processing

The ECG signal or electrocardiogram is a widely used exam in cardiology field. It describes the electrical activity of the human heart and has a high clinical value for diagnosing cardiac arrhythmias. From the ECG signal processing, several parameters can be extracted. As a rule, different waves' durations and shapes can indicate some cardiac abnormalities [3]. Fig. 1 represents the formation of the ECG signal, which reflects the different deflections and contractions of the heart muscles, making it possible to diagnose a patient's cardiac state. As shown in the Fig. 1, RA, LA, RV, and LV represent respectively the right and left Atria and ventricle.

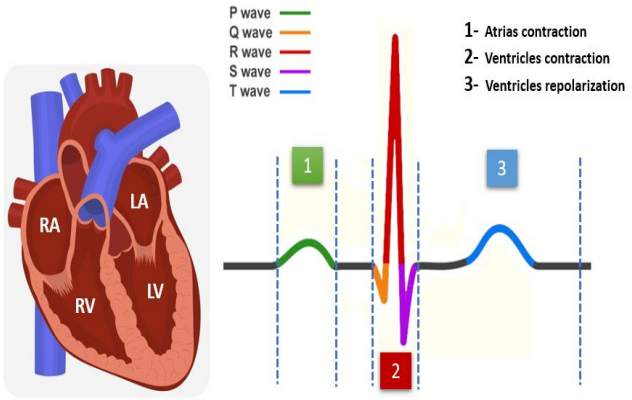

Fig. 1. ECG Signal Formation.

To make the best use of ECG data in large quantities, intelligent diagnostic systems have appeared. These systems can improve the quality of the signal; extract useful information, and offer a diagnosis that can help doctors make the right decisions.

The biomedical engineering revolution forces researchers to enhance the automatic diagnosis by optimization of ECG processing algorithms in order to ensure real-time monitoring of cardiac data [12-16]; and their implementation on embedded systems as recent technological resources [8,17-19].

The ECG signal is characterized by low frequency and a small amplitude. So, it is often affected by various kinds of noise as interference due to electrical appliances, high-frequency noises produced by muscles activity, and low-frequency noises of body movements in relation with respiration, which distorts its morphology, resulting a wrong diagnostic of the heart state of the patients [5,20]. To overcome this problem, the ECG signal must first go through a precise and effective preprocessing step.

The preprocessing step aims to remove or reduce the different noises. In this context, many methods are used such as Digital Filters (FIR/IIR) [21-22], Empirical Mode Decomposition (EMD) and Ensemble EMD (EEMD) based techniques [10, 23-25], Discrete Wavelet Transform (DWT) [26-27, 3], Dual-Tree Wavelet Transform (DT-WT) [28] and Adaptive Filtering [29,8].

B. Related Works based ECG Denoising

Digital filters are represented by Finite Impulse Response (FIR) and Infinite Impulse Response (IIR). They are used to denoise ECG signals. Their names are originally linked to the mathematical definition, and their expressions are given respectively by (1) and (2):

$$Y(n) = \sum_{k=0}^{N-1} \mathrm{b(q)X(n-q)} \quad (1)$$

$$y(n) = \sum_{k=1}^{M} a_k\, y(n-k) + \sum_{k=0}^{N} b_k\, x(n-k) \quad (2)$$

The implementation of FIR filters can be done without feedback as it is shown in Fig. 2 where X(n) presents the input signal, Y(n) is the filtered signal, Z^{-1} operator is a delay in the Z transformation, N is the filter order, and b_q are the coefficients of the filter transfer function. Various windowing techniques are used, as example: Rectangular window, Kaiser

window, Hamming window, Hanning window and Blackman window. IIR filters are designed using filters as Chebyshev filter, Butterworth filter, Inverse Chebyshev filter [30]. The major difference between them is that FIR filters are stable for any input signal. However, IIR filters can alter to unstable due to the feedback as shown in Fig. 3. Where a_n and b_m are the filter transfer function coefficients.

Most of the works used FIR or IIR filters by selecting a bandwidth related to the utile data from the ECG signal. From different papers, it can be deduced that that FIR filter with Kaiser Window eliminates noises from ECG signal with less alteration in the waveform [4,20]. FIR filters' problem is the high computational due to the number of coefficients needed to achieve excellent denoising result and a group of delay in response, which is the main challenge in real-time systems [4].

Wavelet methods have proven to be more common and effective than FIR/IIR filters [31]. Wavelet methods simultaneously characterize time and frequency information. They decompose the signal to different resolutions using low pass filters (H[n]) to get the approximations (A) and high pass filters (g[n]) to get the details (D) as it's explained in Fig. 4. The ECG signals are denoised using thresholding techniques. But they have some limitations since they reduce the signal amplitude, which can affect the R waves.

To overcome this limitation, methods based on EMD are used, the signal is disintegrated into a sequence of intrinsic mode functions (IMFs), and noisy IMFs are removed, but this technique can remove some useful information when eliminating the noisy IMFs. EEMD is used to overcome this problem by removing the mode-mixing [32].

To deal with the problems of complexity in ECG denoising, W. Jenkal et al. developed a new approach inspired from image denoising [33]. This approach is an adaptive dual threshold filter which is dedicated to removing high-frequency noises [34-35]. This method aims to compute three elements for a selecting window (the average of the window, the higher threshold, and the lower threshold) and then the correction of the window's median value using the thresholding. The process is explained in the following block diagram shown in Fig. 5.

The performance evaluation of this method is made in [8] based on the SNRimp result comparison between the ADTF and techniques based on EEMD. This evaluation shows that the ADTF gives very good results compared to the EEMD denoising algorithm presented in [36] and a competitive SNRimp results to the enhanced EEMD method (EEMD-GA) published in [37].

The main characteristic of the ADTF algorithm is its low complexity compared to the cited methods. The ADTF presents a linear complexity C(n) depending just on the signal size n. A comparative study of the complexity between the EMD, the EEMD, and the ADTF methods is presented in [8]. The conclusion of this study shows that ADTF presents a low complexity, unlike EMD and EEMD, which is related to various parameters, namely, the length of the signal, the number of the noisy signals, the number of IMFs, as well as the number of sifting processes. Comparing with the DWT, it also has linear complexity, in the manner of EMD/EEMD, it's related to other parameters not only the size of the signal, we are talking about the wavelet mother's coefficients, the number of decomposition's level and also the thresholding technique [38].

In the next section, a detailed study of the ADTF is presented as well as simulation results.

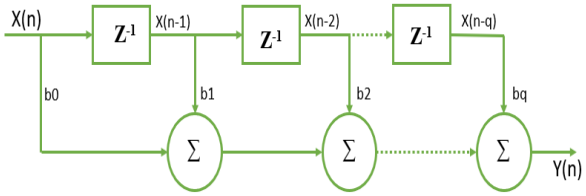

Fig. 2. FIR Filter Conception.

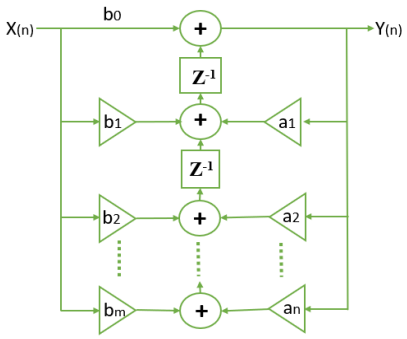

Fig. 3. IIR Filter Conception.

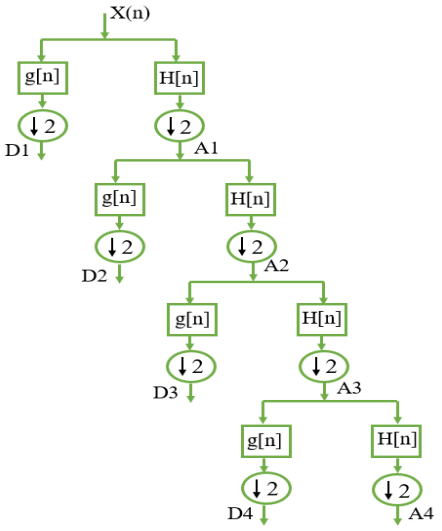

Fig. 4. DWT Decomposition.

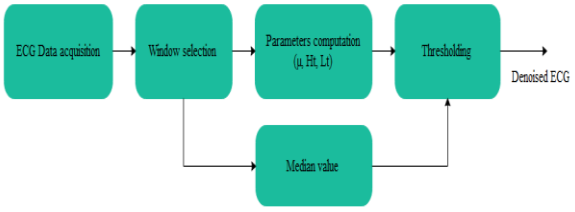

Fig. 5. ADTF Algorithm Overview.

C. The ADTF Technique

As described above, the ADTF algorithm aims to compute the average of the selected window (μ), the lower threshold (Lt), and the higher threshold (Ht). Hereinafter, they are presented respectively by (3), (4), and (5).

$$\mu = \frac{1}{W}\sum_{i=n}^{n+W} \text{Input}(i) \quad (3)$$

$$\text{Lt} = \mu - [(\mu - \text{Min}) * \alpha] \quad (4)$$

$$\text{Ht} = \mu + [(\text{Max} - \mu) * \alpha] \quad (5)$$

Where the window size is W, Input (i) is the noisy signal, and n presents the signal length. Min is the minimum value of the window, Max is the maximum value of the window, and α is the thresholding coefficient with:

$$0 < \alpha < 1$$

α is essentially used to adapt the thresholding. According to the filtering process, α varies between 0% and 100%, lower values are recommended for a high concentration of noise, and higher values are tolerated in the opposite case [7].

The window selection is not arbitrary; it's used to compare the median sample to his left and right regions, as stated in [7], a window of five samples gives the best results in terms of MSE and SNRinput.

Fig. 6 shows the result of the ADTF filtering applied on the signal n° 234 from the MIT-BIH Arrhythmia database corelated with white Gaussian noises of 5 dB, using a window of 5 samples, an α coefficient of 5%. The compilation is done using Matlab R2019a.

The algorithm validation is done using Matlab coding [7], but this remains a functional validation, while in biomedical engineering, Real-time processing is required in most cases, especially ECG signal processing. The time constraint is related to the sampling frequency of the used signals. In the MIT-BIH Arrhythmia database, the signals are sampled 360 Hz which leads to an interval of 2.7 ms between the samples. Using Matlab coding, 150 ms is needed to process and correct each sample. Thusly, with Matlab implementation, the system is too far from being in real-time.

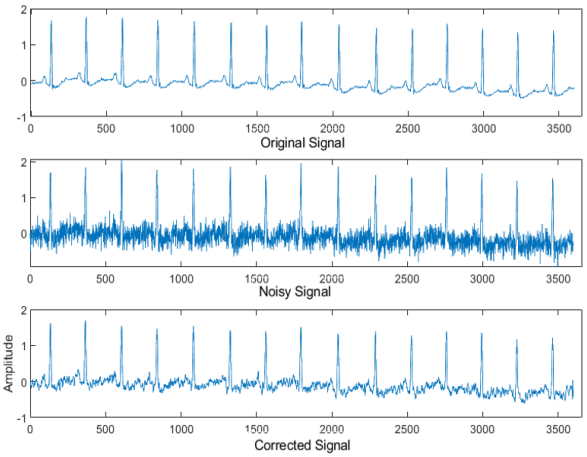

Fig. 6. The Denoising Results of the Signal n°234 of the MIT-BIH Arrhythmia Database Correlated with 5 dB of the WGN.

We have opted for C/C++ optimization using OpenMP parallelization in order to optimize the code. As follows a comparative study of the obtained results using Matlab, C/C++ non-optimized algorithm and OpenMP optimized algorithm on different architectures.

III. Results and Discussion

A. System Specification

In this paper, ECG signals from MIT-BIH Arrhythmia, the international database Physionet, which incorporates 48 half-hour records are used. These signals are converted to numerical values at 360 Hz with a resolution of 11-bits. Different signals with additive White Gaussian Noise (WGN) at SNR levels of 10dB and 20dB are used. Table I presents the used signals for both the Matlab and the C/C++ validation.

The validation of the algorithm was performed using Matlab. In order to ensure a reliable and accurate real-time processing, OpenMP programming is used in the implementation to ensure parallel programming on a shared memory multiprocessor system. The parallelism is achieved by creating a set of threads; these treads execute independently and simultaneously the appointed tasks. Using OpenMP, the program could be optimized to evaluate processing times using three different architectures.

The work is based on Raspberry 3B and XU4 architectures; the choice of these architectures was based on the low energy consumption and also a low weight for embedded application. Despite the desktop giving excellent processing time results, but it's not a good solution for biomedical monitoring because of its size and portability in the real case. Good results can also be given using TX1, TX2, and AGX Xavier boards; notably, with GPU part, we can decrease the processing time [39]. But the major problems are their cost and power consumption. Raspberry and XU4 present a low-cost embedded system with low power computation [40]. The rest of this study will allow to make the right decision by comparing the found processing times for the two architectures. Table II presents the used architectures specifications.

TABLE I. ECG Signal Records

Signal number	The corresponding Signal from MIT-BIH
1	Record n° 100
2	Record n° 100 + 10 dB of WGN
3	Record n° 100 + 20 dB of WGN
4	Record n° 101
5	Record n° 101 + 10 dB of WGN
6	Record n° 101 + 20 dB of WGN
7	Record n° 103
8	Record n° 103 + 10 dB of WGN
9	Record n° 103 + 20 dB of WGN
10	Record n° 113
11	Record n° 113 + 10 dB of WGN
12	Record n° 113 + 20 dB of WGN

TABLE II. DESKTOP, RASPBERRY AND XU4 SPECIFICATIONS

Type	Desktop	Raspberry	XU4
Processor	Intel® Core™ i5-4200M	Broadcom BCM2837B 0	Exynos 5422 big. LITTLE
Cores	Quad	Quad	Octa
CPU	I5 4200M	ARM Cortex-A53 (ARMv8)	ARM Cortex A15/A7
GPU	HD Graphics 4600 /AMD Radeo n R5 M230	Broadcom Videocore-IV	Advanced Mali
Support Language	C/C++/OpenCL/ OpenGL	C/C++	C/C++/OpenCL/Open GL
Frequency	2.50 GHz	1.4GHz	2GHz/1.4GHz
Weight	2.6 kg	50 g	60 g
Energy	90 w	15.5 w	5W
Dimension s	377 x 250 x 34 mm	85 x 56 x 17 mm	82 × 58 × 22 mm

B. *Experimental Results*

To evaluate the denoising performance of the C/C++ code of the ADTF algorithm, Mean Square Errors (MSE), Percentage Root-mean-square Difference parameter (PRD), and Signal to Noise Ratio (SNR) are computed for 12 records of ECG of 10s from the MIT-BIH arrythmia database. Their expressions are presented respectively by (6), (7), and (8).

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\big(Input(i) - Output(i)\big)^2 \quad (6)$$

$$SNRout = 10 \times log10\left(\frac{\sum_{i=1}^{N}(Input(i))^2}{\sum_{i=1}^{N}(Output(i)-Input(i))^2}\right) \quad (7)$$

$$PRD = \sqrt{\frac{\sum_{i=1}^{N}(Input(i)-Output(i))^2}{\sum_{i=1}^{N}(Input(i))^2}} \times 100 \quad (8)$$

Where input(i) represents each input sample and output (i) the filtered sample, N is the size of signal.

The experimental results in Fig. 7, 8, and 9 show that the C/C++ program provides concrete denoising than the Matlab code in terms of good SNR with less MSE and PRD.

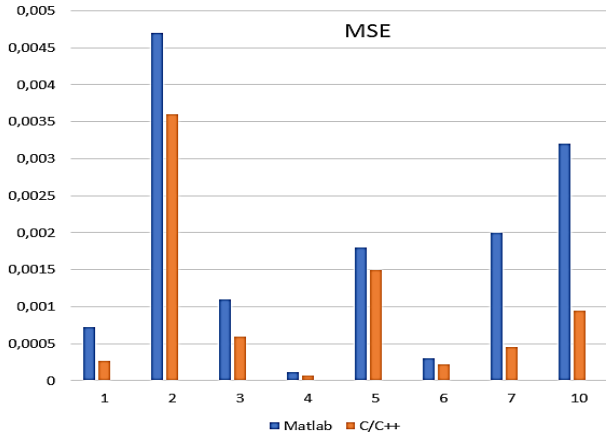

Fig. 7. MSE Comparison of Denoised Signals using Matlab and C/C++.

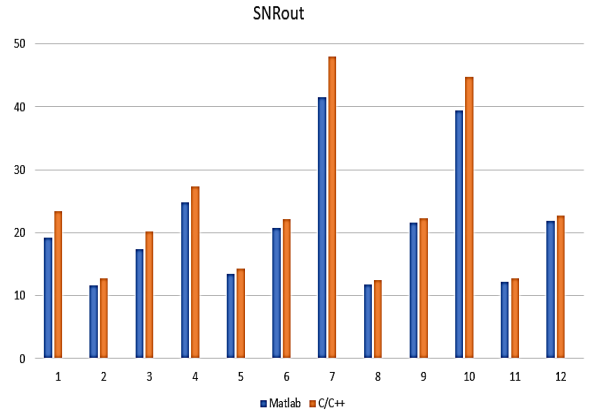

Fig. 8. SNRout Results Comparison of Denoised Signals using Matlab and C/C++.

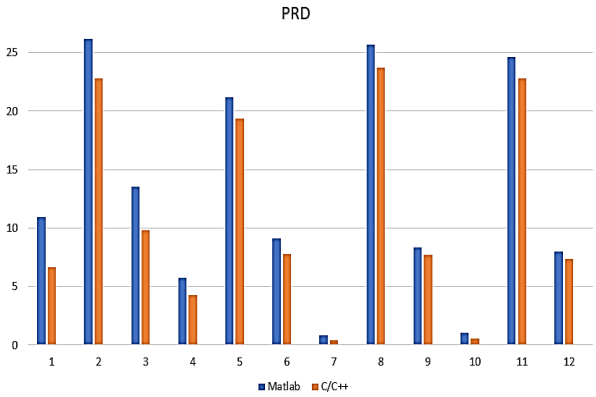

Fig. 9. PRD Comparison of Denoised Signals using Matlab and C/C++.

As shown, the C/C++ gives better performance in noise reduction. As follows, a comparison study of the execution time on Desktop is presented in Table III. It presents the needed time to process each sample of the signal based on nine iterations of time average calculation. The obtained results show that using OpenMP on the desktop, the results showed a ×4 speed-up compared to the time obtained with the naive C/C++ implementation. In Fig. 10, the execution time of Matlab is added to visualize the interest of parallel programming in this case.

TABLE III. EXECUTION TIME IN DESKTOP

C++		Open MP	
Iteration	***Time (ms)***	***Iteration***	***Time (ms)***
1	1.21	1	0.25
2	2.01	2	0.4
3	0.9	3	0.133
4	1.47	4	0.36
5	1.85	5	0.41
6	1.81	6	0.408
7	2.04	7	0.405
8	1.81	8	0.365
9	0.94	9	0.405
Average	**1.56**	**Average**	**0.348**

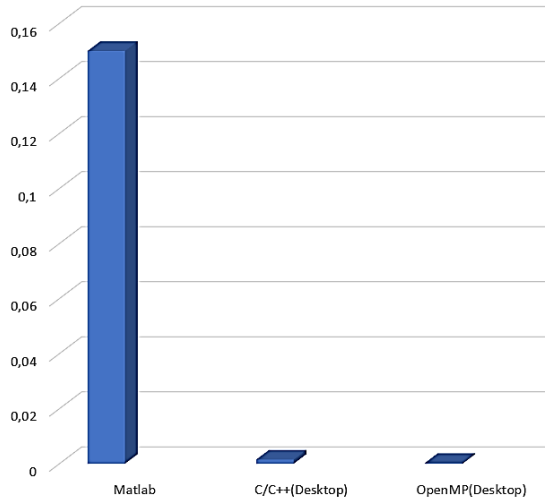

Fig. 10. Processing Times on Desktop using different Tools.

C. Performance Evaluation using Embedded Architectures

Performance evaluation implementing embedded architectures is the main of this work. After proving the algorithm performance in noise reduction, a reliable real-time implementation is necessary.

The Matlab implementation was used for the algorithm validation and results interpretation, but the very high execution times lead us to exclude any software optimization in order to speed up processing in this implementation. Then we opted for C/C++ programming. The implementation is done, as cited above using a Desktop, a Raspberry 3B and XU4 boards. The C/C++ implementations results are too much better than those given by Matlab implementation, but they stay a little far from the real-time as it appears in Fig. 11, with 1.56 ms for Desktop, 9.2ms for Raspberry and 6.8 ms for XU4.

The block diagram of the parallelized algorithm is presented in Fig. 12. The first block is the ECG signal acquisition which is not the subject of this paper. The second aims to divide to the input signal by the thread number (A). Here a test is done; if the number of signal samples is not divisible on A the system searches for the optimal signal size by adding a few samples at the end of the signal. The added samples can be calculated using the values of the last window of the signal, an average of the window can be calculated to replace missing samples. The third block consists of memory allocation and parameters initialization as the α coefficient and the window size.

Block 4 is the core of this work; it aims to execute the denoising procedure. This is where the parallelism is applied, ECG signal is divided by the number of threads, each thread runs the denoising program on a portion of the signal instead of the whole signal. The last blocks present the denoised signal's exploitation step either for additional processing, storage, display, or transmission.

To optimize the given execution times, OpenMP is used. Fig. 13 depicts the pseudo code of the OpenMP-based parallel computation algorithm.

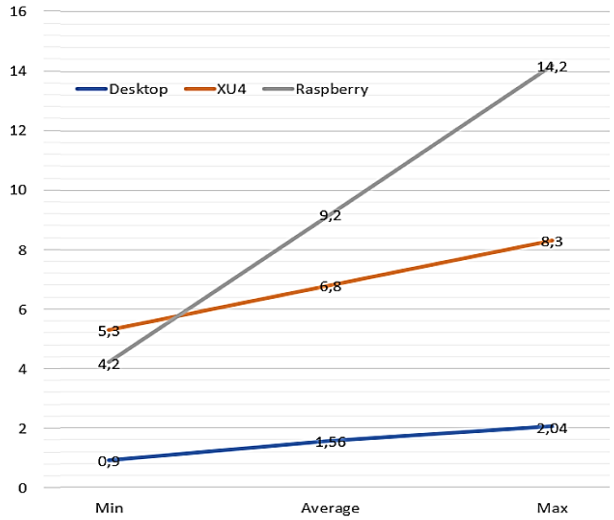

Fig. 11. Min and Max Processing Times using different Architectures.

The first step is to determine the optimal size that can give a divisible value over the number of threads. Thereafter, we set the input and output signals that will be written in the output file. This output file will be plotted subsequently using Matlab in order to display the errors and compute the different evaluation metrics.

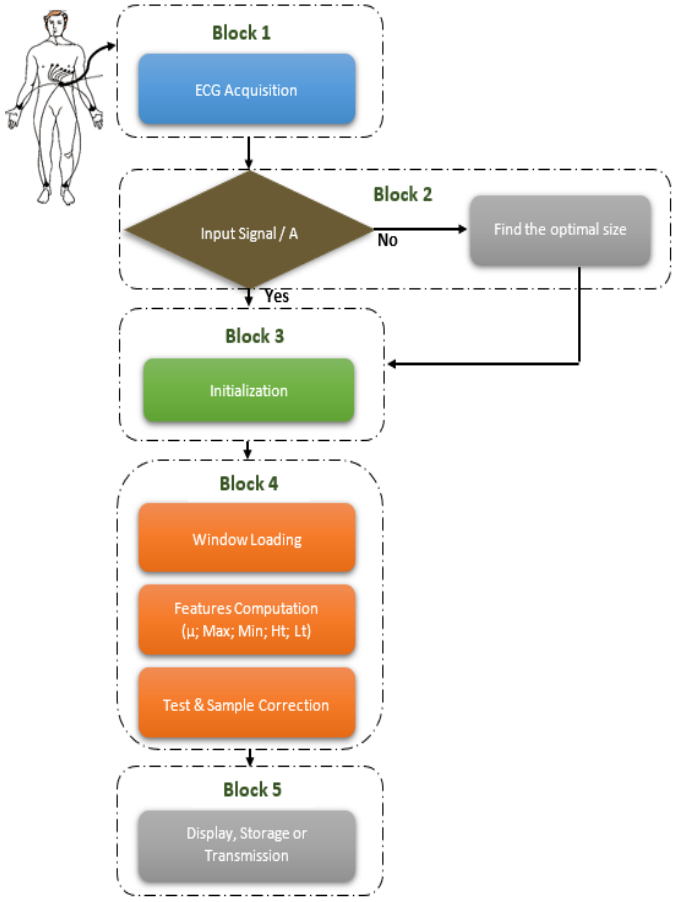

Fig. 12. The Algorithm Block Diagram.

ALGORITHM

```
#define NUMTHREADS
#define NAME_FILE "INPU_ECG"
#define NOM_SORTIE1 "OUTPUT_Signal"
#define Alpha
Input:  ECG Signal.txt
        Find the optimal signal size
        X(i)  <=  Input(i)
        i=1;
        α, W Initialisation
        Threads = NUMTHREADS
        #pragma omp parallel for shared (Min, Max,
        µ, Ht, Lt,) num_threads (threads)
          for i=1 to N-W
              Compute Min, Max, µ, Ht and Lt
              If X (i+W/2) > Ht Output (i+W/2) =Ht;
              If X (i+W/2) < Lt Output (i+W/2) =Lt;
              Else
              Output(i+W/2) = Input(i+W/2)
              i= i+1;
          End for
        Output(N-1) = X(N-1)
        Output(N) = X(N)
End.
```

Fig. 13. Pseudo Program of the Algorithm using OpenMP.

Fig. 14 shows processing time using OpenMP implementation. A time of 7.5 ms is achieved for one sample processing using Raspberry architecture, 2.34 ms using XU4 architecture, and 0.34 ms using the desktop. The time constraint posed by the acquisition system forced us to process each sample with a delay less than 1/360Hz, which implies trying to process each sample within 2.77ms. The results allowed to eliminate the choice of raspberry due to the processing times, which exceed 2.77ms. Despite their low energy consumption and weight, the time evaluation has shown that this architecture cannot process the algorithm in real-time. The desktop gave a very low processing time, 0.34ms, which shows the desktop's high performance, but the drawback here is the high-power consumption, which makes this type of system does not meet the reliability requirement. On the other hand, the XU4 architecture met the time constraint, making it the best choice for this application. In addition, its low power consumption and low weight confirm the choice.

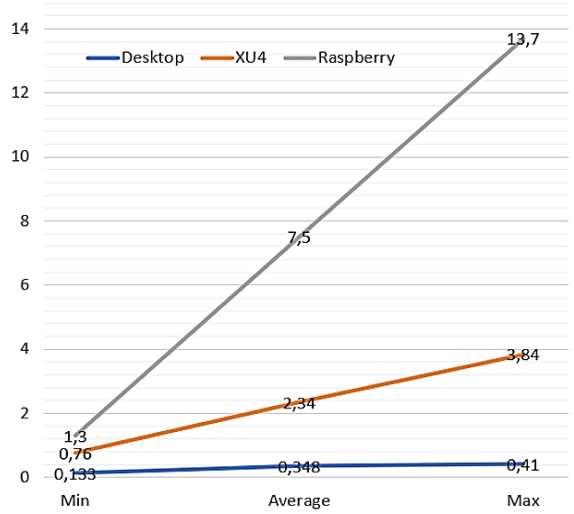

Fig. 14. OpenMP Executing Time.

TABLE IV. DIFFERENT EXECUTING TIMES

Executing time (ms)	Desktop	XU4	Raspberry
C/C++	1.56	6.8	9.2
C/C++ - OpenMP	0.34	2.34	7.5

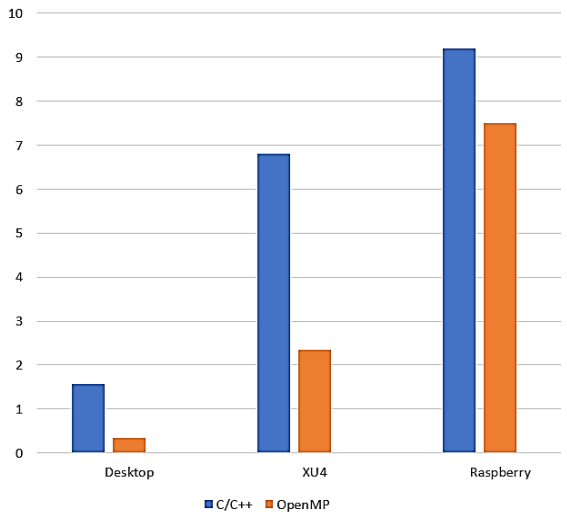

Fig. 15. Mean Processing Times (MS) based on different Architectures.

Table IV and Fig. 15 shows a comparison of all processing times using the different architectures and both C/C++ and OpenMP parallel implementation.

The optimization of the algorithm on the XU4 architecture proves to be very efficient and makes it possible to speed up processing and achieve real-time processing times in addition to its power consumption advantage.

IV. CONCLUSION

In this paper, a complex algorithm-based ECG signals processing is studied in order to meet the requirements of monitoring applications in terms of real time and portability on a low power architecture.

The evaluation of the algorithm using Matlab allowed validation of the algorithm and the evaluation of the different metrics (MSE, PRD, and SNR errors).

The approach followed by the algorithm parallelization on an adequate architecture is effective to process signals at 2.34ms/samples using a 360 Hz frequency acquisition.

This study opens up research perspectives to design a system integrating sensors and a SoC whose architecture is similar to that of the XU4 and which integrates an FPGA in order to carry out on-the-fly signal processing without data storage.

ACKNOWLEDGMENT

We would like to thank the National Centre for Scientific and Technical Research (CNRST) of Morocco for the financial support (scholarship number: 588UIZ2017).

REFERENCES

[1] W. Jenkal, R. Latif, A. Toumanari, A. Dliou, O. El, and F. Mrabih, "QRS Detection Based on an Advanced Multilevel Algorithm,"

International Journal of Advanced Computer Science and Applications (IJACSA)., vol. 7, no. 1, 2016, doi: 10.14569/IJACSA.2016.070135.

[2] S. Elouaham, A. Dliou, R. Latif, and M. Laaboubi, "Filtering of Biomedical signals by using Complete Ensemble Empirical Mode Decomposition with Adaptive Noise," *Int. J. Comput. Appl.*, vol. 149, no. 7, pp. 39–43, 2016, doi: 10.5120/ijca2016911515.

[3] W. Jenkal, R. Latif, A. Toumanari, A. Dliou, O. El B'Charri, and F. M. R. Maoulainine, "An efficient algorithm of ECG signal denoising using the adaptive dual threshold filter and the discrete wavelet transform," *Biocybern. Biomed. Eng.*, vol. 36, no. 3, pp. 499–508, 2016, doi: 10.1016/j.bbe.2016.04.001.

[4] P. C. Bhaskar and M. D. Uplane, "High Frequency Electromyogram Noise Removal from Electrocardiogram Using FIR Low Pass Filter Based on FPGA," *Procedia Technol.*, 2016, doi: 10.1016/j.protcy.2016.08.137.

[5] S. Mejhoudi, R. Latif, A. Elouardi, and W. Jenkal, "Advanced Methods and Implementation Tools for Cardiac Signal Analysis," *Adv. Sci. Technol. Innov.*, pp. 95–103, 2019, doi: 10.1007/978-3-030-05276-8_11.

[6] Z. Wang, F. Wan, C. M. Wong, and L. Zhang, "Adaptive Fourier decomposition based ECG denoising," *Comput. Biol. Med.*, 2016, doi: 10.1016/j.compbiomed.2016.08.013.

[7] W. Jenkal, R. Latif, A. Toumanari, A. Dliou, and 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," *Int. Rev. Comput. Softw.*, vol. 10, no. 11, pp. 1089–1095, 2015, doi: 10.15866/irecos.v10i11.7821.

[8] W. Jenkal, R. Latif, A. Toumanari, A. Elouardi, A. Hatim, and O. El'bcharri, "Real-time hardware architecture of the adaptive dual threshold filter based ECG signal denoising," *J. Theor. Appl. Inf. Technol.*, vol. 96, no. 14, pp. 4649–4659, 2018.

[9] G. Tang and A. Qin, "ECG de-noising based on empirical mode decomposition," *Proc. 9th Int. Conf. Young Comput. Sci. ICYCS 2008*, pp. 903–906, 2008, doi: 10.1109/ICYCS.2008.178.

[10] M. A. Kabir and C. Shahnaz, "Denoising of ECG signals based on noise reduction algorithms in EMD and wavelet domains," *Biomed. Signal Process. Control*, vol. 7, no. 5, pp. 481–489, 2012, doi: 10.1016/j.bspc.2011.11.003.

[11] T. Wang, M. Zhang, Q. Yu, and H. Zhang, "Comparing the applications of EMD and EEMD on time-frequency analysis of seismic signal," *J. Appl. Geophys.*, vol. 83, pp. 29–34, 2012, doi: 10.1016/j.jappgeo.2012.05.002.

[12] O. El B'charri, R. Latif, W. Jenkal, and A. Abenaou, "The ECG Signal Compression Using an Efficient Algorithm Based on the DWT," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 7, no. 3, pp. 181–187, 2016.

[13] A. Giorgio, "A New FPGA-based Medical Device for the Real Time Prevention of the Risk of Arrythmias," *Int. J. Appl. Eng. Res.*, vol. 11, no. 8, pp. 6013–6017, 2016.

[14] Z. Zhang, Z. Li, and Z. Li, "An Improved Real-Time R-Wave Detection Efficient Algorithm in Exercise ECG Signal Analysis," *J. Healthc. Eng.*, vol. 2020, 2020, doi: 10.1155/2020/8868685.

[15] S. Sahoo, P. Biswal, T. Das, and S. Sabut, "De-noising of ECG Signal and QRS Detection Using Hilbert Transform and Adaptive Thresholding," *Procedia Technol.*, vol. 25, no. Raerest, pp. 68–75, 2016, doi: 10.1016/j.protcy.2016.08.082.

[16] T. A. Rashid, C. Chakraborty, and K. Fraser, "Advances in Telemedicine for Health Monitoring: Technologies, Design and Applications," *Adv. Telemed. Heal. Monit. Technol. Des. Appl.*, no. June, 2020, doi: 10.1049/pbhe023e.

[17] H. W. Lim, M. Syafiq, M. Sani, A. Hashim, and Y. W. Hau, "Throb : System-on-Chip based Arrhythmia Screener with Self Interpretation," pp. 30–36, 2015.

[18] W. Shen, D. Wei, W. Xu, X. Zhu, and S. Yuan, "Parallelized computation for computer simulation of electrocardiograms using personal computers with multi-core CPU and general-purpose GPU," *Comput. Methods Programs Biomed.*, vol. 100, no. 1, pp. 87–96, Oct. 2010, doi: 10.1016/J.CMPB.2010.06.015.

[19] L. V. R. Kumari, Y. P. Sai, N. Balaji, and K. Viswada, "FPGA Based Arrhythmia Detection," *Procedia Comput. Sci.*, vol. 57, pp. 970–979, 2015, doi: 10.1016/j.procs.2015.07.495.

[20] B. Chandrakar, O. P. Yadav, and V. K. Chandra, "a Survey of Noise Removal Techniques for Ecg Signals," *Ijarcc*, vol. 2, no. 3, pp. 1354–1357, 2013, [Online]. Available: www.ijarcce.com.

[21] J. M. Łęski and N. Henzel, "ECG baseline wander and powerline interference reduction using nonlinear filter bank," *Signal Processing*, vol. 85, no. 4, pp. 781–793, 2005, doi: 10.1016/j.sigpro.2004.12.001.

[22] Z. Haque, R. Qureshi, M. Nawaz, F. Khuhawar, N. Tunio, M. Uzair, "Analysis of ECG Signal Processing and Filtering Algorithms," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 10, Issue 3, pp. 545-550, 2019, doi: 10.14569/IJACSA.2019.0100370.

[23] P. Nguyen and J. M. Kim, "Adaptive ECG denoising using genetic algorithm-based thresholding and ensemble empirical mode decomposition," *Inf. Sci. (Ny).*, vol. 373, pp. 499–511, 2016, doi: 10.1016/j.ins.2016.09.033.

[24] M. Rakshit and S. Das, "An efficient ECG denoising methodology using empirical mode decomposition and adaptive switching mean filter," *Biomed. Signal Process. Control*, vol. 40, pp. 140–148, 2018, doi: 10.1016/j.bspc.2017.09.020.

[25] G. Han, B. Lin, and Z. Xu, "Electrocardiogram signal denoising based on empirical mode decomposition technique: An overview," *J. Instrum.*, vol. 12, no. 3, 2017, doi: 10.1088/1748-0221/12/03/P03010.

[26] K. J. Bijwe1, S. S. Vasekar, "FPGA Implementation of DWT for ECG Signal Pre-Processing", *International Journal of Engineering Science and Computing* vol. 6, no. 8, pp. 2450–2452, 2016.

[27] E. M. El Hassan and M. Karim, "An FPGA-based implementation of a pre-processing stage for ECG signal analysis using DWT," 2014 *2nd World Conf. Complex Syst. WCCS 2014*, pp. 649–654, 2015, doi: 10.1109/ICoCS.2014.7060929.

[28] O. El B'charri, R. Latif, K. Elmansouri, A. Abenaou, and W. Jenkal, "ECG signal performance de-noising assessment based on threshold tuning of dual-tree wavelet transform," *Biomed. Eng. Online*, vol. 16, no. 1, pp. 1–18, 2017, doi: 10.1186/s12938-017-0315-1.

[29] S. Poungponsri and X. H. Yu, "An adaptive filtering approach for electrocardiogram (ECG) signal noise reduction using neural networks," *Neurocomputing*, vol. 117, pp. 206–213, 2013, doi: 10.1016/j.neucom.2013.02.010.

[30] P. Podder, M. Mehedi Hasan, M. Rafiqul Islam, and M. Sayeed, "Design and implementation of butterworth, chebyshev-i and elliptic filter for speech signal analysis," *arXiv*, 2020, doi: 10.5120/17195-7390.

[31] P. N. Malleswari, C. Hima Bindu, and K. Satya Prasad, "An investigation on the performance analysis of ECG signal denoising using digital filters and wavelet family," *Int. J. Recent Technol. Eng.*, vol. 8, no. 1, pp. 166–171, 2019.

[32] S. Thakran, "A hybrid GPFA-EEMD_Fuzzy threshold method for ECG signal de-noising," *J. Intell. Fuzzy Syst.*, vol. 39, no. 5, pp. 6773–6782, 2020, doi: 10.3233/JIFS-191518.

[33] V. Gupta, V. Chaurasia, and M. Shandilya, "Random-valued impulse noise removal using adaptive dual threshold median filter," *J. Vis. Commun. Image Represent.*, vol. 26, pp. 296–304, 2015, doi: 10.1016/j.jvcir.2014.10.004.

[34] W. Jenkal, R. Latif, A. Elouardi, and S. Mejhoudi, "FPGA Implementation of the Real-Time ADTF process using the Intel-Altera DE1 Board for ECG signal Denoising," *Proc. 2019 IEEE World Conf. Complex Syst. WCCS 2019*, 2019, doi: 10.1109/ICoCS.2019.8930780.

[35] S. Mejhoudi, R. Latif, W. Jenkal, and A. Elouardi, "Real-Time ecg signal denoising using the adtf algorithm for embedded implementation on fpgas," *Proc. 2019 IEEE World Conf. Complex Syst. WCCS 2019*, 2019, doi: 10.1109/ICoCS.2019.8930771.

[36] K. M. Chang, "Arrhythmia ECG noise reduction by ensemble empirical mode decomposition," *Sensors*, Vol. 10, Issue. 6, 2010, pp.6063-80.

[37] P. Nguyen, J. M. Kim, "Adaptive ECG denoising using genetic algorithm-based thresholding and ensemble empirical mode decomposition," *Information Sciences*, Vol. 373, 2016, pp. 499-511.Y.

[38] L. Wu, D. Agrawal, and A. El Abbadi, "A comparison of DFT and DWT based similarity search in time-series databases," pp. 488–495, 2000, doi: 10.1145/354756.354857.

[39] S. Hossain and D-j. Lee, "Deep Learning-Based Real-Time Multiple-Object Detection and Tracking from Aerial Imagery via a Flying Robot with GPU-Based Embedded Devices," *Sensors*, Vol. 19, Issue. 15, 2019, doi: 10.3390/s19153371.

[40] U. Prodhan, T. Saha, R. Shaharin, T. Emon and M. Rahman, "Implementation of Low Cost Remote Primary Healthcare Services through Telemedicine: Bangladesh Perspectives," *International Journal of Advanced Computer Science and Applications (IJACSA)*, 11(11), 2020. http://dx.doi.org/10.14569/IJACSA.2020.0111118.