Low-Frequency and Low-Power NIOS-V Soft-Processor Based System-On-Chip Architecture for Real-Time ECG Signal Denoising Using the DWT-ADTF Process

Original Scientific Paper

Bouchra Bendahane*
Ibn Zohr University, National School of Applied Science, Laboratory of System Engineering and Information Technology,
Agadir, Morocco
bouchra.bendahanne@edu.uiz.ac.ma

Wissam Jenkal
Ibn Zohr University, National School of Applied Science, Laboratory of System Engineering and Information Technology,
Agadir, Morocco
w.jenkal@uiz.ac.ma

Mostafa Laaboubi
Ibn Zohr University, Faculty of Applied Science,
Ait Melloul, Morocco
mostafa.laaboubi@uiz.ac.ma

Boujemaa Nassiri
International University of Agadir – Universiapolis, Laboratory of Interdisciplinary Applied Research,
Agadir, Morocco
boujemaa.nassiri@e-polytechnique.ma

Samir Elouaham
Ibn Zohr University, National School of Applied Science, Laboratory of System Engineering and Information Technology,
Agadir, Morocco
s.elouaham@uiz.ac.ma

*Corresponding author

Abstract *– Several are the embedded architectures that have been introduced to filter ECG signals. However, processing frequencies, consumed hardware, power, and time are still challenging. The elevated computational complexity of the considered algorithms is the major reason. The DWT-ADTF is a recent low-computational complexity process that has been introduced for ECG denoising. This paper proposes an FPGA-based software architecture providing real-time ECG signal filtering with low frequency and power consumption. Our system consists of a Nios-V soft processor interconnected to 69.34 KB of RAM through the AXI4 and Avalon interfaces. The Nios-V is used thanks to its hardware flexibility and five-stage pipeline. Using the C language, our architecture is designed to take advantage of this hardware capacity. The data is loaded and treated through a sample-by-sample and shifting approach. Consequently, the processor's pipeline is well-derived for real-time processing, and no permanent data storage is needed. Thus, under only 50 MHz, our architecture ensures real-time data loading and processing while consuming less hardware and power. The denoising quality is assessed using the average MSE, PRD, and SNRimp. Several noise types and ECG signals from the MIT-BIH Arrhythmia database are involved. The processing frequency and the consumed hardware, power, and time evaluate the efficiency and performance of our architecture. Promising results are achieved compared to other works, which makes our architecture appropriate for real-time ECG monitoring systems. Further, its lower consumption favors more ECG processing integration.*

Keywords: *Denoising, design, ECG, embedded, FPGA, implementation, low-frequency, low-power, Nios-V, real-time, pipeline, soft-processor, software*

Received: January 20, 2026; Received in revised form: March 16, 2026; Accepted: March 30, 2026

1. INTRODUCTION

ECG, or electrocardiography, is the most used sensor for the electrical behavior recording of the heart as well as the related abnormalities' diagnosis [1, 2]. Existing ECGs ensure data recording using single-to twelve-lead-based recordings [3]. An ECG signal can serve in several environments: hospitals, ambulatories, home monitoring, and also remote monitoring purposes [1]. Even with the particular importance and attention that the academic and industrial researchers provide to ECG signal processing, cardiovascular diseases are still the main reason behind human death in the world [1, 4]. The problem comes back to the insufficient effectiveness and real-time capacity of the diagnosis given by existing ECG monitoring systems. Further, the availability of these devices is mainly lim-

ited to hospitals due to power supply needs, while they are needed continuously in other environments such as ambulatories. This imposes the development of more effective, real-time, and low-power devices, ensuring long-term availability of the device. Furthermore, an early and predictive diagnosis is a necessity.

Because of direct dependency in processing quality between all ECG signal processing tasks, the preprocessing is a critical stage. Therefore, more interest has been directed to ECG signal filtering, either algorithms, or embedded architecture development. The ECG signal is very weak in the presence of noise. This occurs because its frequency composition belongs to a very low-frequency range. The corrupting noises vary between baseline wander (BW), electromyogram (EMG), power-line interference (PLI), channel noise, random noise, and electrode motion artifacts [1, 5-7]. The ECG signal preprocessing literature involves standard as well as hybrid-based methods. These include digital filters (FIR and IIR) [8, 9], empirical mode decomposition method (EMD) [10-13], wavelet-based methods [14-18], and models based on artificial intelligence [19, 20].

The continuous development in embedded electronics technology has given powerful devices in terms of parallelism and pipeline-deepening capabilities, serving the increasing demand for efficient computing in different fields. These devices vary between CPUs, GPUs, DSPs, FPGAs, and ASICs, which are the most efficient solutions with no reconfigurability and no programmability [21]. Additionally, there are heterogeneous architectures such as system-on-chip (SoC) FPGAs. Further, soft processor-based intellectual property (IP), like the Intel Nios, overcomes the flexibility issue of hard processors. Furthermore, the field programmable analog array (FPAA) is used for analog circuit development [22]. Based on these devices, several ECG signal filtering methods have been deployed. Given that embedded systems are resource limited and that the ECG-based diagnosis is real-time constrained, low computational complexity algorithms must be considered for implementation. However, existing ECG signal denoising architectures implement high-computation-complexity algorithms. This results in insufficient processing time for real-time denoising and high resource consumption (hardware and power), challenging the integration of other treatments and limiting the availability of the said architectures to hospitals, while ECGs are also needed in other environments like ambulatories.

Jenkal *et al*. suggested in [23, 24] a real-time hardware architecture in the favor of the ADTF algorithm. The architecture works under 3.6 kHz. However, this is a higher frequency compared to the required processing of the ADTF. Some years later, Mejhoudi *et al*. introduced a software system on the basis of multi-core CPUs for the ADTF too [25]. The implementation was performed in the Raspberry Pi 3B+, the Odroid XU4, and a desktop PC with an i5-4200M Core. Thanks to the multiple cores, the preprocessing is ensured in a short time. However, this is performed in a segment-by-segment manner and not sample-by-sample. This means that the data needs to be initially stored, which affects the real-time capacity of the system. Further, such architectures use higher frequencies (gigahertz) and consume more power (watts) because of the unused hardware. Last year, we introduced in [26] an optimized real-time ADTF-based hardware architecture. In that work we dealt with the challenges and limitations in performance and efficiency of the existing embedded system, including anterior architectures of the ADTF. The effectiveness of our architecture was competitive, and it was so efficient and performant.

In [27], an FPGA-based hardware accelerator was proposed for the adaptive max-mean-min algorithm (AMaMeMi) for cleaning physiological signals. Chandra *et al*. introduced in [28] numerous FPGA-based improved architectures for PLI removal using the adaptive LMS filter. In [29] and in [30], FPGA-based and ASIC-based hardware architectures were suggested by Das *et al*. for EMD and EEMD processes. In 2022, Elbedwehy *et al*. introduced a hardware solution for computing the single node reservoir system (SNRC) and the mean filter [31]. FIR filters were also considered by Tripathi *et al*. for an FPGA-based architecture for different noise removal [32]. The WT was presented by Goel *et al*. for a hardware implementation using an operating frequency of 180.278 MHz, where the db3 and db2 mother wavelets were considered [33]. More recently, Janwadkar *et al*. introduced an ASIC-based FIR filter in [34]. The common challenges addressed by these embedded systems are the power, hardware, and time consumption, as well as high processing frequencies.

The DWT-ADTF [15, 16] is one of the effective algorithms that have been suggested in the ECG denoising literature. The said algorithm involves the discrete wavelet transform (DWT) technique, a time-frequency analysis technique utilized in signal and image processing, and the adaptive dual threshold filter (ADTF), an adaptive algorithm recently developed for ECG denoising [23, 24]. The DWT-ADTF takes its particularity from its low computational complexity, which raises its implementability in different embedded system platforms. Consequently, the denoising quality and low computational complexity tradeoff are effectively addressed by the DWT-ADTF method. Therefore, and within the current work, we suggest an embedded architecture implanting the DWT-ADTF algorithm. The said method had known its first embedded system in 2021, where a hardware architecture was presented by Mejhoudi *et al*. in [35]. However, the high peaks correction part was not considered, while it is important in recuperating the original high peaks required for heart rate computing. Further, a processing frequency of 50 kHz was used, which is larger than the needed clock cycles for DWT and ADTF computing in a hardware-based architecture. This means that the power consumption of the architecture will be increased due to the useless clock cycles.

In the current work we suggest a software-based architecture implementing the DWT-ADTF for the ECG signal filtering. The proposed solution uses the Nios-V soft processor, which is the newest generation of Intel soft processors. In contrast to hard processors, soft processors represent a highly flexible solution thanks to their configurability in FPGAs. This means that only required hardware and frequencies will be used regarding the considered application needs. Therefore, the power consumption of the whole system will be significantly optimized. On the other side, the core of the Nios-V g (general purpose) comes with a five-stage pipeline, which increases the core's performance. Furthermore, the instruction set architecture (ISA) of the Nios-V g core is extensible, which raises its deepening with additional hardware accelerators. Consequently, the processor performance will be further increased. To well exploit this hardware capacity for an effective and efficient processing of the ECG signal, we designed an appropriate software architecture. This later ensures real-time processing through a sample-by-sample handling of the ECG data at the level of all processing parts. This manner also allows a good exploitation of the Nios pipeline. Additionally, this avoids the need for any permanent storage of data either before or after processing. Thus, a high amount of unnecessary hardware and consequently of power is reduced. Compared to the considered works, the outcomes of the proposed architecture are significant and convincing.

2. RELATED WORKS

2.1. DWT-ADTF ALGORITHM

The DWT-ADTF is a preprocessing technique recently introduced for ECG signals [15]. It is the result of combining the ADTF, recently introduced in [23, 24], and the DWT, a famous signal processing method. The ADTF development was not only for responding to ECG signal denoising needs but also for embedded system deployment. Therefore, the ADTF has given very promising denoising quality with very low computational complexity. Additionally, the computational complexity of the ADTF depends only on the size of the input data [24], which raises its implementability in embedded architectures.

In light of the ECG signal's non-stationarity and the variety of the corrupting noises, the denoising quality of the ADTF has known considerable improvement by the integration of the DWT. This hybridization has conserved the implementability in embedded systems of the resulting algorithm by maintaining its low computational complexity property. This is clearly shown in Table 1, which presents the execution time comparison between the DWT-ADTF and other well-known techniques. The comparison proves that the lowest time corresponds to the DWT-ADTF. This has resulted thanks to the low computational complexity of the DWT-ADTF. Therefore, the DWT-ADTF presents a particular solution for embedded systems targeting ECG signal denoising.

Table 1. Comparison of the average execution time of DWT-ADTF with other related algorithms

Algorithm	Execution time (s)
DWT-ADTF [16, 36]	0.040
DWT-Thresholding [36]	0.1189
DWT-NLM [36]	0.9986
DWT-VMD-NLM [36]	0.5480
EMD-LM [16]	0.1620
EEMD-LM [16]	13.87005

The ADTF enhancement has been continued, always considering the denoising quality and algorithmic complexity trade-off. Therefore, a very recent version has been introduced in [16]. Competitive results have been achieved, which reflects the importance of the DWT-ADTF as well as the ADTF itself in the ECG signal filtering field. As a consequence, we propose through the current work an embedded system for the DWT-ADTF process presented in [15]. The denoising principle of the DWT-ADTF involves three main stages: two levels of DWT, ADTF application to the reconstructed signal of the DWT, and finally, the high peaks correction of the ADTF signal.

2.1.1. DWT

The DWT is a convolution product of the input signal with the coefficients of the chosen wavelet. This process results in the decomposition of the signal into a collection of approximations (A[m]) and details (D[m]). Approximations group low-frequency elements of the signal at the considered decomposition level, while high-frequency components are given by details (D[m]). Approximations and details are calculated according to the equations (1) and (2), respectively [7, 15]. L(l) is the low-pass filter of the chosen mother wavelet, while H(l) is its high-pass filter. Reconstructing the original signal is feasible using its approximations and details according to the equation (3) [7, 15]. For the DWT-ADTF, the DWT is applied to the noised ECG signal for two decomposition levels. Details D1 and D2 are removed since they concentrate a high amount of high-frequency noises. Therefore, the approximation A2 is twice reconstructed.

$$A[m] = \sum_{l=-\infty}^{l=+\infty} g(l)x(2m-l) \tag{1}$$

$$D[m] = \sum_{l=-\infty}^{l=+\infty} h(l)x(2m-l) \tag{2}$$

$$x[m] = \sum_{l=-\infty}^{l=+\infty} [D(l)h(2l-m)+A(l)g(2l-m)\,] \tag{3}$$

2.1.2. ADTF

The ADTF is an adaptive process of the filtering of ECG signals. It consists of a dual thresholding through a shifting

window [23, 24]. The thresholds computation is based on the minimum, maximum, and mean values of the shifting window [23]. The thresholds are computed according to the equations (4), (5) and (6), where δ is the noised signal, β that is equal to 0.1 is the thresholding coefficient, and S is the size of the window. In the case of the DWT-ADTF algorithm, the ADTF is applied to the DWT output for a sliding window of eleven samples. More details about the ADTF method can be found in [23, 24].

$$Mean = \frac{1}{S} \sum_{i=n}^{n+(S-1)} \delta(i) \tag{4}$$

$$HT = Mean + (Max - Mean) * \beta \tag{5}$$

$$LT = Mean - (Mean - Min) * \beta \tag{6}$$

2.1.3. High peaks correction

This part of the DWT-ADTF algorithm aims at the correction of the high peak amplitude in the ECG signal resulting after the ADTF. The correction principle consists of replacing the QRS regions of the ADTF signal with those of the DWT signal. This is achieved through a sliding window of 1800 samples and a thresholding process, where the threshold is 70% of the window's maximum.

2.2. MAIN RELATED WORKS

Thanks to the continuous development in embedded electronic technology, powerful devices have been introduced. Therefore, numerous embedded architectures for ECG signal filtering have been introduced. A hardware architecture in favor of the ADTF process was suggested by Jenkal *et al*. in [23, 24] for real-time filtering. The proposed solution is a structural and pipelined architecture working under 3.6 kHz. This is quite sufficient to fulfill the real-time processing of the ECG signals. However, it is larger than the required processing of the ADTF, so the unused cycles contribute to increasing power consumption.

In 2021, the ADTF has known another embedded system suggestion. It is a multi-core CPU-based software architecture that was introduced by Mejhoudi *et al*. in [25]. The proposed solution performed many threads on multiple segments of the input ECG data. Thus, the input data needs to be stored initially, and the number of threads is user-chosen. The software application was developed using C/C++ and the parallel programming tool OpenMP. The implementation was performed in the Raspberry Pi 3B+, the Odroid XU4, and a desktop PC with an i5-4200M processor. Such architectures reduce significantly the processing time. However, the suggested architecture handles the ECG signal as segment-by-segment, which affects the real-time capacity since the data needs to be stored before. Additionally, these architectures use gigahertz and come with additional hardware for communication and transmission to serve different applications. However, these may not be used, which leads to unnecessary power consumption. On the other hand, a desktop PC is not an embedded system since it can't be integrated into a larger mechanical or electrical system. Furthermore, the Raspberry B3+ and Odroid XU4 are single-board microcomputers designed for specific computing tasks within a larger mechanical or electrical system. However, their ISAs lack hardware flexibility, where they are fixed and can't be extended. This means that the whole system maximum performance is software-driven within the existing hardware.

In [26], we have introduced an optimized nonstructural hardware architecture for the ADTF. The work consisted of a real-time deep pipelined architecture of six stages working under 1.4 kHz. Significant efficiency and performance have been marked for the suggested system compared to other existing architectures that are based on well-known methods such as the FIR/IIR and EMD.

The DWT-ADTF algorithm had its first known embedded system in 2021 [35], where Mejhoudi *et al*. proposed an FPGA-based hardware architecture. The introduced solution is a structural architecture, denoising the ECG data within only a DWT and an ADTF processing, without correcting higher peaks attenuated by the ADTF. This significantly influences the denoising effectiveness of the architecture. 50 kHz was the frequency used for data processing. The said frequency is sufficient to achieve real-time ECG signal preprocessing. However, it is larger than the required clock cycles for both DWT and ADTF operations under a hardware-based architecture. Therefore, unused clock cycles will increase the system's power consumption.

In [27], a software-hardware (SW/HW) architecture was proposed for BW removal, where the max-mean-min algorithm (MaMeMi) was considered. The proposed solution suits different physiological signals, including ECG. The MaMeMi filter presents a cutoff frequency of 17 Hz, which may affect informative low-frequency components of the ECG data. To resolve this, the authors suggested an adaptive MaMeMi (AMaMeMi) to reduce the cutoff frequency to the 0-1 Hz range independently of user input. Therefore, one of the filter parameters is value-fixed, and the other is proportionally linked to the input signal amplitude. However, the introduced filter is not fully adaptive, as a manual mode is enabled for the filter parameter evaluation by the user. This may influence the cutoff frequency of the filter and require an expert user to evaluate its parameters. The MaMeMi represented the hardware part of the SW/HW architecture, while the software side concerned the control mode of the filter parameter computing. The accelerator was implemented in a Xilinx Zynq-7000 SoC FPGA.

In [33], Goel *et al*. address the challenge of removing PLI from ECG signals. The core algorithm was based on the wavelet packet transform (WPT) and utilized a thresholding technique. The wavelet mothers db3 and db2 were used within a two-level decomposition/reconstruction. The Donoho soft thresholding process was adopted. However, standard hard/soft thresholds cause oscillation or constant deviation, distorting the recon-

structed signal. The main contribution of the work is a hardware-efficient architecture designed for real-time operation on FPGAs. This architecture was designed to process data sample-by-sample rather than frame-by-frame, which is essential for true real-time processing. However, high operation frequencies are used, which will lead to high power consumption. Further, the architecture assumes stationary noise, while real-world ECG has complex, non-stationary noise. In 2022, Elbedwehy *et al*. introduced a single-node reservoir computing system (SNRC) and mean-based filter in [31]. The architecture dealt with the PLI and EMG noises within a floating-point data representation of sixteen bits.

Even with the numerous architectures given in the ECG denoising literature, the hardware, power, and time consumption, the processing frequencies, and the denoising quality are still a challenging trade-off.

3. METHODS AND MATERIALS

3.1. PROPOSED METHOD

The suggested solution is a RISC-V-based software architecture developed with the C language. The Nios processor is a soft processor. This means that it does not physically exist, but it is created by configuring the FPGA logic using a dedicated ecosystem. This provides high flexibility in the logic's use and distribution according to the application needs. Thanks to the RISC-V ISA's flexibility, modularity, and open-source property, more instructions and custom hardware accelerators can be integrated, extending the pipeline depth of the Nios core. Therefore, the proposed DWT-ADTF software architecture is designed to take advantage of the existing pipeline's five stages so that maximum performance can be achieved. Additionally, the proposed DWT-ADTF software architecture is designed to make easier the integration of custom hardware accelerators later.

3.1.1. Configured hardware architecture

The hardware architecture consists of the Nios-V g core and an on-chip RAM memory component. The interconnection is ensured within a set of standard interfaces such as the Avalon MM (Memory Mapped) and the AXI4 (AMBA Advanced Extensible Interface). Fig. 1 gives a macro view of the whole system hardware.

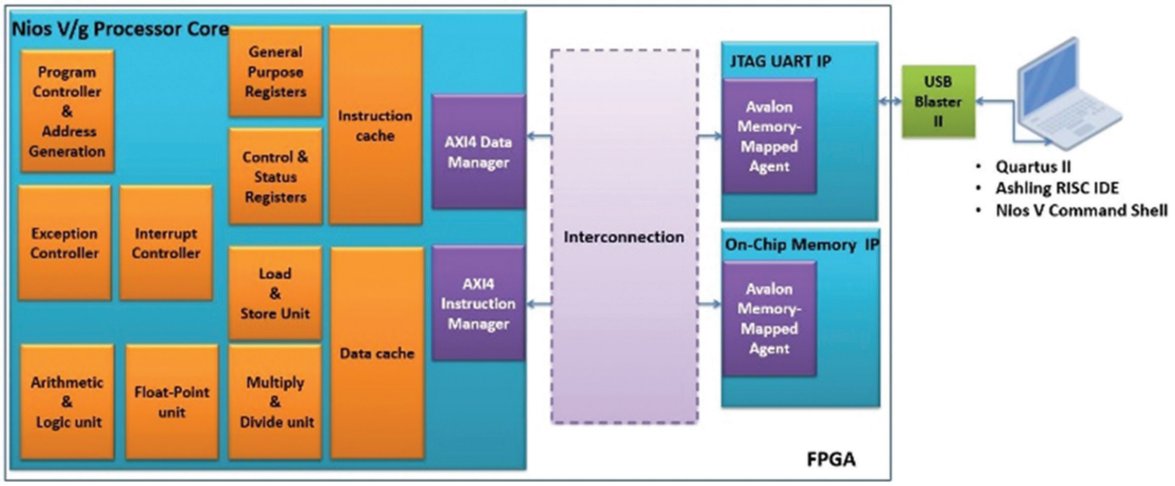


Fig. 1. A macro view of the configured Nios-V-based hardware architecture

Thanks to the fluid data loading and processing ensured by the proposed software architecture, no permanent data storage is needed. Therefore, fewer memory RAM cells are required for data preparation, while more cells can be added to caches so that the system performance is increased. Consequently, only 69.34 KB of RAM and 4 KB of cache are needed.

3.1.2. Proposed software architecture

The software architecture is designed to take advantage of the soft-processor flexibility and the five stages of the Nios core pipeline, avoiding any permanent data storage, reducing memory access needs, and ensuring real-time data loading and processing. Therefore, the data is handled sample-by-sample rather than frame-by-frame, which is essential for true real-time processing.

To represent the ECG data, we used 32-bit floating-point representation. The MIT-BIH Arrhythmia database provides ECG recordings sampled at 360 Hz with 11-bit resolution over a ±5 mV range. The analog-to-digital conversion uses a baseline of 1024 ADC units (corresponding to 0 mV) with a gain of 200 ADC units per mV. Therefore, raw samples are converted to mV using the equation (7). This conversion yields signals in the approximate range of ±5 mV, with the baseline centered at 0 mV.

$$value(mV) = \frac{raw\ value - 1024}{200} \quad (7)$$

For databases other than MIT-BIH Arrhythmia, the data should be converted according to the used ADC conditions. Concerning the size of each of the three windows used by our system, the first one used for the

convolution is 8 samples which, corresponds to the number of the DWT filter coefficients (db4 wavelet). This window size is database independent. In the ADTF step, a window of 11 samples is used. This size is empirically chosen, where lower MSE is achieved regarding other sizes, as reported by the developers of the ADTF and DWT-ADTF algorithms in the references [23] and [15]. For the high peaks correction, the used window is of 1800 samples (5s). This corresponds approximately to six normal heart beats. The time duration of one normal heartbeat is approximately 0.833 s, so for a 360 Hz sampling rate, six heartbeats correspond to 0.833 s * 360 * 6 = 1800 samples. For other databases, this window size can be adapted according to their sampling rates (e.g., for 250 Hz, the window size should be 1250 samples).

The DWT part involves four convolutional computations. Therefore, four memory regions are allocated for data preparation. Eight elements is the size required for each region since the db4 wavelet mother is used. For the next processing of incoming samples, the previous data are one sample shifted so that the new raw sample is integrated. In this way, convolution computing is continuously ensured without any permanent data storage. On the other hand, the down/up sampling operations occur in a switching manner. This means that one of the two occurs in each processing cycle (newly acquired ECG sample). This reduces the required clock cycles since the up-sampling (zero value insertion) will occur when the down-sampling will not. Fig. 2 details the processing principle of the DWT.

The ADTF is based on a windowing process of eleven samples. Therefore, a memory region of eleven elements is allocated for data preparation. Continuous processing is ensured always by shifting the existing data and inserting new samples. Fig. 3 details the processing principle of the ADTF part.

The high-peaks correction part is based on a windowing and thresholding treatment. Therefore, a nine hundred elements-based memory region is allocated to prepare data. No arithmetic is applied to the incoming data. However, the corrected data is a switch between DWT and ADTF data regarding the threshold-based test. To manage corrected data outputting without any overlapping with the threshold-based test, we separated the two. In addition, the threshold-based test is controlled (activated/deactivated) with a delay. In this way, the corrected data is passed serially to the out, while incoming data are welcome, shifting existing ones. In this manner, the next sample for the threshold-based test is put in the appropriate memory cell before the activation of this latter, and the process goes regularly. Fig. 4 details the processing principle of this part.

The proposed software architecture has well exploited the five-stage pipeline of the Nios processor by avoiding the pipeline stalls. Fig. 5, Fig. 6, and Fig. 7 show the pipeline diagrams of the main treatment parts of our system. This concerns the shifted data loading, which prepares data for the DWT, ADTF, and the high peaks correction; the convolution applied in the DWT; and the nested branches used mainly in the DWT and the high peaks correction.

3.2. MATERIALS AND PROOF OF CONCEPTS

We validated the suggested DWT-ADTF architecture within two verification approaches: denoising quality and architectural performance. The filtering effectiveness of our architecture is qualitatively and quantitatively assessed in reference to the original software version of the DWT-ADTF process using MATLAB, as well as in comparison with some related and recent works. Therefore, a collection of clinical ECG signals of the MIT-BIH Arrhythmia database are used. The used ones are the signal 100, 101, 103, 113, 115, 117, 119, and 122. These signals are contaminated with different noise types, namely white Gaussian (WGN), blue, and pink noises. The WGN intensity across frequencies is regularly distributed. In contrast, blue and pink noise intensities present a non-regular distribution. The use of noise variety evaluates the robustness of our architecture against such noise distributions.

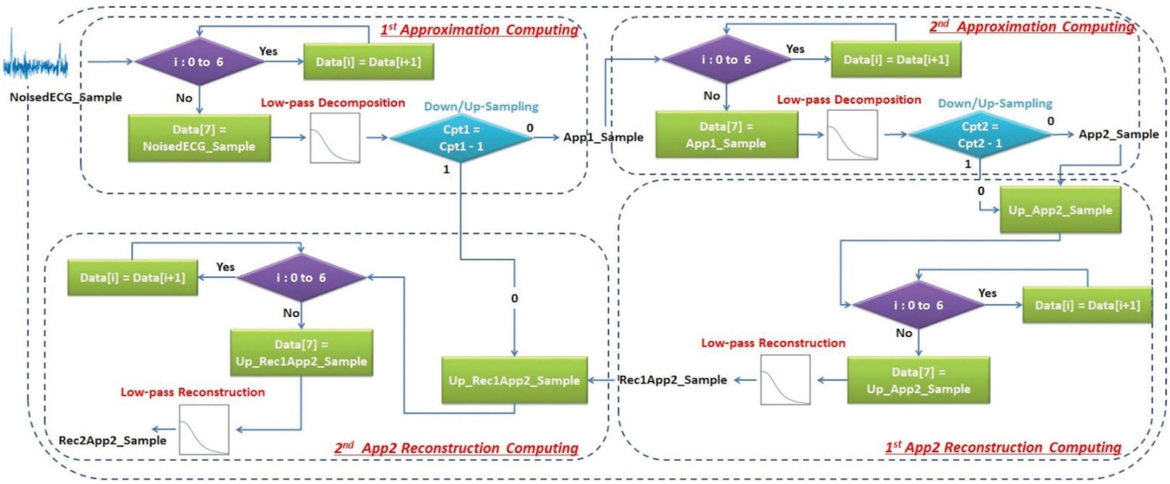


Fig. 2. Proposed software architecture for the DWT

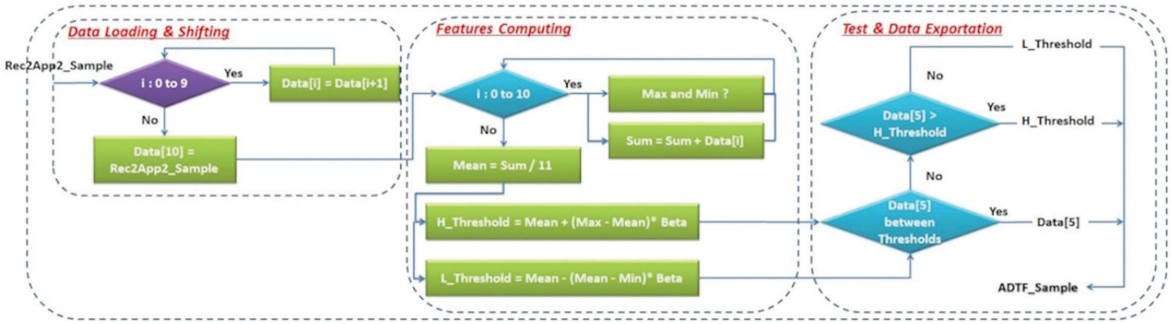


Fig. 3. Proposed software architecture for the ADTF

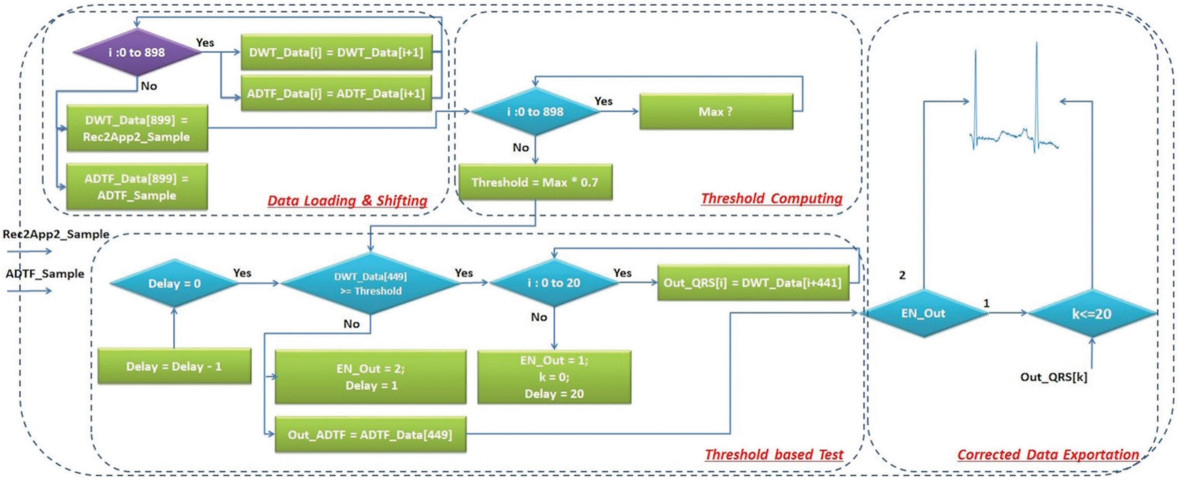


Fig. 4. Proposed software architecture for the high peaks correction

The robustness of the proposed system toward noise variety is quantified based on a set of benchmarks, namely, the percentage root difference (PRD), mean square error (MSE), and the signal-to-noise ratio improvement (SNRimp). These metrics are computed according to the equations (8), (9), and (10), where x is the authentic signal, δ is the corrupted signal, and y is the corrected signal. The outcomes can be interpreted as effective for lower MSE and PRD and for higher SNRimp. The suggested architecture is deployed in the Cyclone V DE10 SoC FPGA using the Quartus Prime Lite edition. The architectural efficiency and performance of our suggested embedded system are assessed on the basis of the consumed time, hardware, and power as well as the required processing frequency. Further, these benchmarks are given in comparison with other related and recent works.

Clock Cycles	Fetch		Decode		Execute		Memory		Write Back	
M	PC=A	**Initialize** i=0								
M+1	PC=B	Loop condition i<7	PC=A	Decode i=0						
M+2	PC=C	**Load** Shift_Data [i+1]	PC=B	Decode i<7 (True)	PC=A	Execute i=0				
M+3	PC=D	**Store** to Shift_Data [i]	PC=C	Decode **Load** Shift_Data [i+1]	PC=B	Execute i<7 (Branch resolution)	PC=A	Complete (Branch continuing)	PC=A	Write i=0
. . . . .	PC=E	**Increment** i	PC=D	Decode **Store** to Shift_Data [i]	PC=C	Execute **Load** Shift_Data [i+1] (Address calc)	PC=B	Complete	PC=B	Complete
	PC=F	Loop branch back (predict taken)	PC=E	Decode **Increment** i	PC=D	Execute **Store** Shift_Data [i] (Address calc)	PC=C	**Load from memory**	PC=C	Waiting
	PC=G	Loop condition i<7 ()	PC=F	Decode branch	PC=E	Execute i++	PC=D	**Store to memory**	PC=C	Write loaded value
	PC=H	**Load** Shift_Data [i+1] (next element)	PC=G	Decode i<7	PC=F	Execute branch (predict taken)	PC=E	**Update i value**	PC=D	Write stored value
		....		....		....		....		....

Fig. 5. Pipeline diagram of the shifted data loading

Clock Cycles	Fetch		Decode		Execute		Memory		Write Back	
M	PC=A	**Load** Shift_Data[0]								
M+1	PC=B	**Load** Filter[0]	PC=A	Decode **Load**						
M+2	PC=C	**Multiply** R0 * F0	PC=B	Decode **Load**	PC=A	Execute **Load**				
M+3	PC=D	**Load** Shift_Data[1]	PC=C	Decode **Multiply** R0 * F0	PC=B	Execute **Load**	PC=A	**Load from memory**		
. . . . .	PC=E	**Load** Filter[1]	PC=D	Decode **Load** Shift_Data[1]	PC=C	Execute **Multiply** R0 * F0 (multi-cycles)	PC=B	**Load from memory**	PC=A	**WB** Shift_Data [0]
	PC=F	**Multiply** R1 * F1	PC=E	Decode **Load** Filter[1]	PC=D	Execute **Load**	PC=C	**Write** product	PC=B	**WB** Filter [0]
	PC=G	**Add** R1 * F1 to R0 * F0	PC=F	Decode **Multiply** R1 * F1	PC=E	Execute **Load**	PC=D	**Load from memory**	PC=C	**WB** product
	. . .	. . .	PC=G	Decode **Add** R1 * F1 to R0 * F0	PC=F	Execute **Multiply** R1 * F1 (multi-cycles)	PC=E	**Load from memory**	PC=D	**WB** Shift_Data [1]
	. . .	. . .	. . .	. . .	PC=G	Execute **Add** R1 * F1 to R0 * F0	PC=F	**Write** product	PC=E	**WB** Filter [1]
		....		....		....		....		....

Fig. 6. Pipeline diagram of the convolution product (DWT)

Clock Cycles	Fetch		Decode		Execute		Memory		Write Back	
M	PC=A	App1_Conv								
M+1	PC=B	**if** (cpt_App1==0)	PC=A	App1_Conv						
M+2	PC=C	App1_DwnSampl= App1_Conv	PC=B	**if** (cpt_App1==0)	PC=A	App1_Conv (multi-cycle)				
M+3	PC=D	Rec1App2_UpSampl = App1_Conv	PC=C	App1_DwnSampl= App1_Conv	PC=B	**if** (cpt_App1==0) (Taken)	PC=A	App1_Conv		
. . . . .	PC=E	cpt_App1= 2	PC=D	Rec1App2_UpSampl = App1_Conv	PC=C	App1_DwnSampl = App1_Conv	PC=B	**if** (cpt_App1==0)	PC=A	App1_Conv
	PC=F	App2_Conv	PC=E	cpt_App1= 2	PC=D	Rec1App2_UpSampl = App1_Conv	PC=C	App1_DwnSampl = App1_Conv	PC=B	**if** (cpt_App1==0)
	PC=G	**else** (cpt_App1==0)	PC=F	App2_Conv	PC=E	cpt_App1= 2	PC=D	Rec1App2_UpSampl = App1_Conv	PC=C	App1_DwnSampl = App1_Conv
	PC=H	**Flush**	PC=G	**else** (cpt_App1==0)	PC=F	App2_Conv (multi-cycle)	PC=E	cpt_App1= 2	PC=D	Rec1App2_UpSampl = App1_Conv
	....	if (cpt_App2==0)	PC=H	**Flush**	PC=G	**else** (cpt_App1==0) (Not Taken)	PC=F	App2_Conv	PC=E	cpt_App1= 2
	....	....	....	if (cpt_App2==0)	PC=H	**Flush**	PC=G	**else** (cpt_App1==0)	PC=F	App2_Conv
	....	....	....	....	....	if (cpt_App2==0) (Taken)	....	**Flush**	....	**else** (cpt_App1==0)

Fig. 7. Pipeline diagram of the nested branches management in the DWT and high peaks correction

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y(i)-x(i))^2 \quad (8)$$

$$DRD = 100\sqrt{\left(\frac{\sum_{i=1}^{i=N}(y(i)-x(i))^2}{\sum_{i=1}^{i=N}x(i)^2}\right)} \quad (9)$$

$$SNR\ imp = 10\log_{10}\frac{\sum_{i=1}^{i=N}(y(i)-z(i))^2}{\sum_{i=1}^{i=N}(y(i)-x(i))^2} \quad (10)$$

4. RESULTS AND DISCUSSION

4.1. DENOISING EFFECTIVENESS EVALUATION

As we have mentioned, the proposed DWT-ADTF software architecture is tested using several original ECG signals corrupted with different noise types. In this subsection we prove qualitatively and quantitatively the denoising efficacy of our proposed architecture.

4.1.1. Qualitative results

The denoising effectiveness of our architecture is qualitatively proved for some ECG signal cases. Fig. 8 displays the signal 115 denoised by our architecture for 5 dB of blue noise. The figure presents the corrected signal in parallel with its corrupted and original versions.

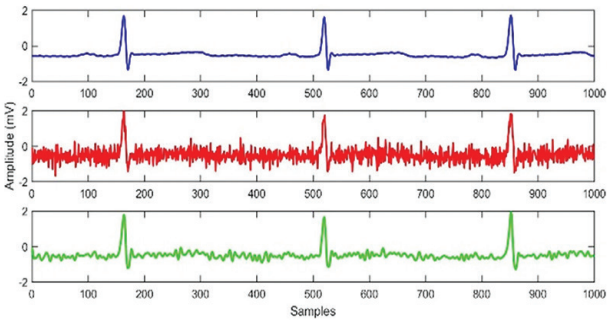


Fig. 8. Denoising of the signal 115 by our DWT-ADTF architecture for 5 dB of blue noise, from bottom to top: corrected, noised, and original signal

Such an input SNR value reflects intensive noise presence. According to Fig. 8, this intensity is well reduced by the proposed architecture. Another denoising case is given in Fig. 9, which presents the case of the signal 103 filtering. This case corresponds to 10 dB of WGN, where a significant correlation is remarked between the reference signal and that corrected by our embedded system.

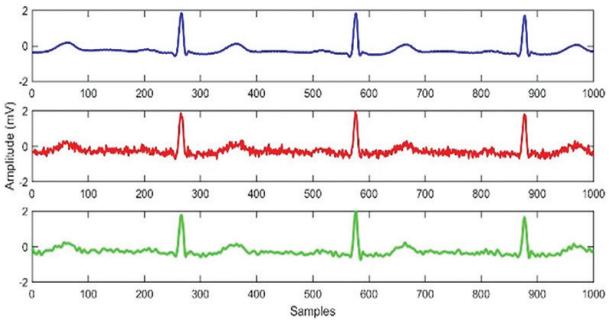


Fig. 9. Denoising of the signal 103 by our DWT-ADTF architecture for 10 dB of white Gaussian noise

The QRS is the main characteristic in an ECG signal. Fig. 10 focuses the view on the QRS region in the case of the signal 115. It presents the correction of the white, blue, and pink noise for different input SNR, namely 5, 10, 15, and 20 dB. Each plot case joins the original signal, its noised, and its denoised versions.

This gives a clear qualitative evaluation between the corrected and the original versions of the considered ECG signal. Regarding these outcomes, a high quality of noise correction is given by the suggested architecture, which proves its robustness against noise variety.

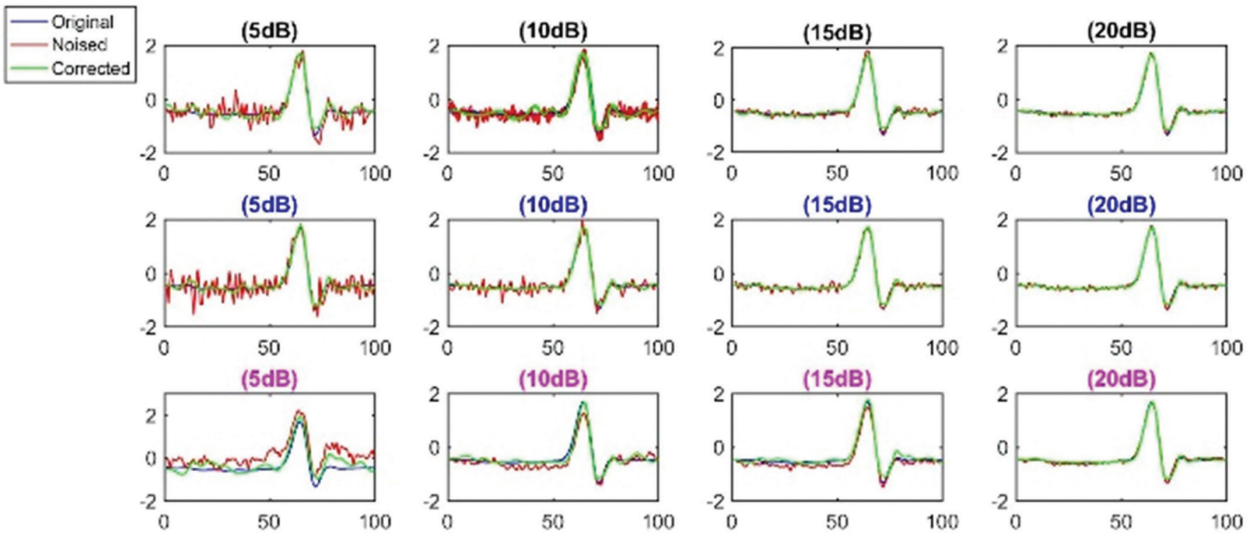


Fig. 10. Denoising of the signal 115 (QRS region) by the proposed DWT-ADTF architecture for different input SNR cases of additive synthetic noises, from top to bottom: white noise, blue noise, and pink noise

4.1.2. Quantitative results

The denoising efficacy of the introduced architecture is quantified by the averaged MSE, PRD, and SNRimp benchmarks for the noise types and ECG signals previously mentioned. Fig. 11 presents the average SNRimp given by the proposed architecture in comparison with the MATLAB DWT-ADTF and other relevant and recent architectures in the WGN case. Similarly, Fig. 12 provides the comparison of the average MSE metric. In the same manner, Fig. 13 shows the comparison results of the average SNRimp in the blue noise case, while Fig. 14 presents the average PRD for the case of pink noise. According to the aforementioned benchmarks' results, the proposed architecture gives competitive outcomes at the level of SNRimp and PRD.

In particular, the lowest MSE values are those related to our proposed system. Therefore, these results prove the effectiveness of the introduced solution.

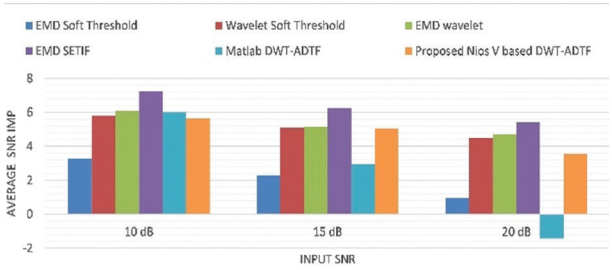


Fig. 11. Average SNRimp comparison between the proposed Nios-V-based system and other related methods [38] in the case of WNG corruption

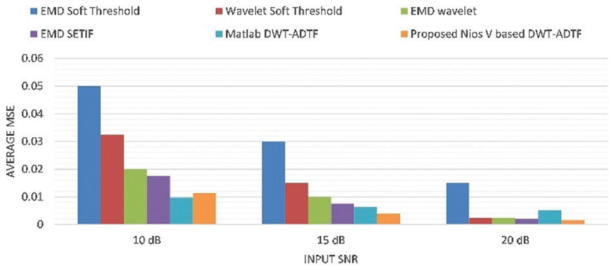


Fig. 12. Average MSE comparison between the proposed Nios-V-based system and other relevant methods [38] in the case of WNG corruption

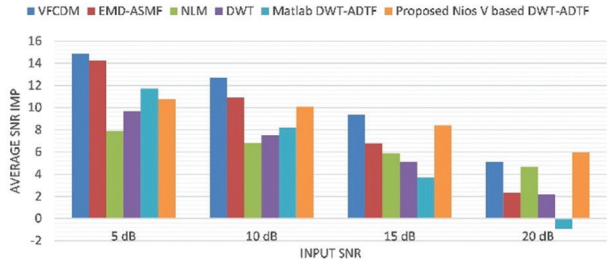


Fig. 13. Comparison of the average SNRimp of the suggested Nios-V-based system and other relevant methods [37] for blue noise corruption

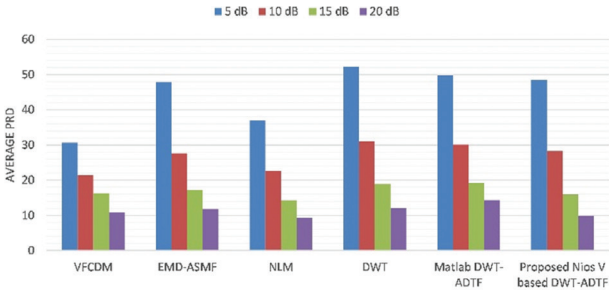


Fig. 14. Comparison of the average PRD of the suggested Nios-V-based system and other relevant methods [37] for pink noise corruption

4.2. PERFORMANCE AND EFFICIENCY EVALUATION

The proposed DWT-ADTF architecture is synthesized under the Intel Quartus Prime Lite edition 23.1, a free edition that includes the required tools for architecture synthesis and implementation. Fig. 15 shows the implementation of our architecture in the Intel Cyclone V DE10 Soc FPGA. The register transfer level (RTL) of the proposed software-based architecture is shown in Fig. 16.

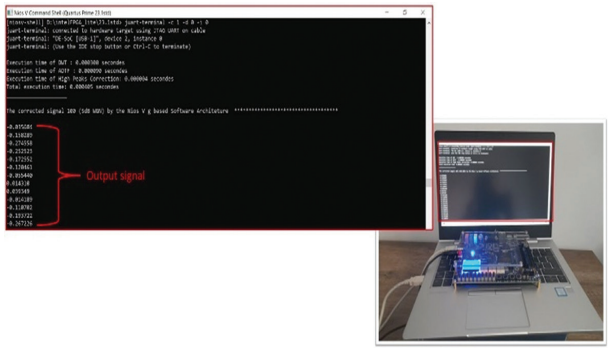


Fig. 15. The used materials for the implementation of our embedded system: desktop PC and Cyclone V DE10 FPGA

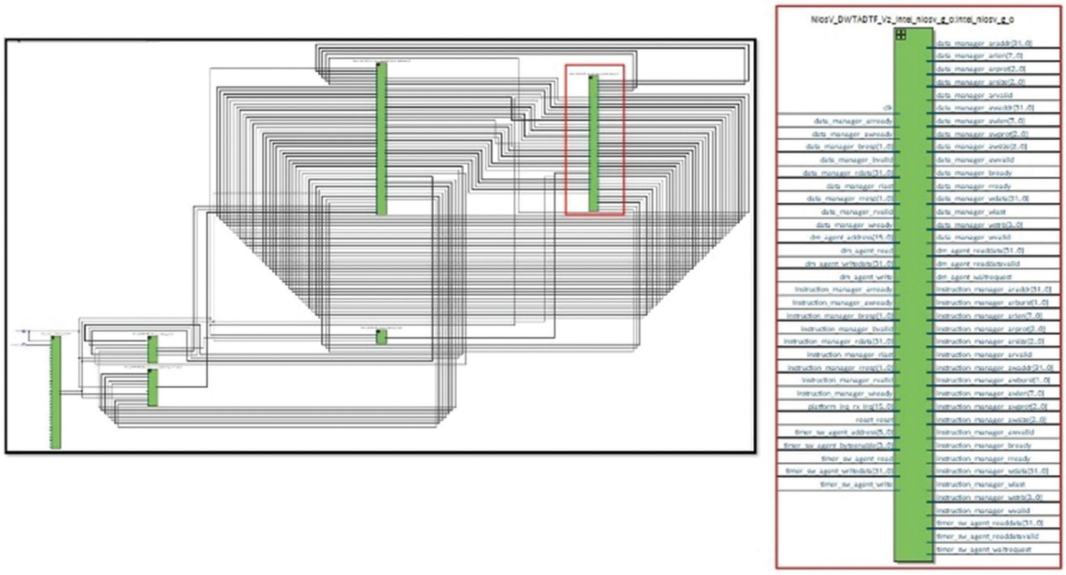


Fig. 16. RTL of the configured Nios-V-based architecture

The performance of the introduced DWT-ADTF architecture is assessed through the demanded hardware, time, and power. Table 2 presents the time consumed by each part of our architecture as well as that consumed by the whole architecture, where 0.41 ms is consumed in total.

In Table 3 we present the resources consumed by the proposed architecture, where a total of 3945 registers, 9% of logical elements, 6% of DSPs, and 69.34 KB of RAM memory are needed by the proposed solution.

The introduced architecture is compared to various recent ECG signal filtering architectures in terms of the used processing frequency, consumed power, and time, as shown in Table 4. Additionally, the SNRimp and SNRout benchmarks are also considered for a complete comparison. We would like to note that the power consumption reported for our proposed system corresponds only to the power dissipated by the Nios core and the interconnected On-Chip memory, while the other FPGA components are not considered. This is because our objective is the optimization of the digital logic of our architecture using FPGAs, not implementing it in an FPGA. Additionally, Table 4 reports different architecture types (software, hardware, and software/hardware). While it is fair to compare the same types of architecture, this comparison aims to highlight the influence of the underlying algorithms on the efficiency of the resulting architectures, even with hardware-based ones.

According to the presented results, our proposed solution provides promising outcomes, achieving an SNRimp and SNRout in dB of 10.74 and 15.45, respectively. These are the higher values compared to the considered works. This is reached in only 0.41 ms of time, with only 0.97 mW of dynamic power, and under a processing frequency of 50 MHz.

Table 2. Consumed time by each part of the proposed DWT-ADTF architecture (ms)

System part	Execution time (ms)
DWT	0.30
ADTF	0.09
High peaks correction	0.005
Full system	0.41

Table 3. Resources consumption by the proposed DWT-ADTF architecture

Resources type	Consumption
Registers	3945
Logical elements (%)	9
DSPs (%)	6
RAM (KB)	69.34

Table 4. Comparison of the performance, efficiency and effectiveness of the suggested architecture with other relevant works

	Algorithm	Architecture	Power	Frequency	Ex. Time (ms)	SNRimp (dB)	SNRout (dB)
Proposed	DWT-ADTF	software	0.97 mW (*)	50 MHz	0.41	10.74	15.45
[15][16]	DWT-ADTF	software	-	-	40	9.17	-
[25]	ADTF	software	5 /15.5 /90 W	1.4/2.5 GHz	2.34 /7.5/ 0.34	-	12.5
[27]	MaxMeanMin	software /hardware	0.7/ 12 mW (HW)	500 Hz/100 MHz	-	0.78	10.15
[29]	EMD	hardware	41.93 mW	50 MHz	0.0662/ 0.703 (1000 data)	-	-
[30]	EMD/EEMD	hardware	-	50 MHz	0.0253 (1000 data)	-	-
[32]	FIR filter	hardware	140 mW	-	-	-	-
[33]	Wavelet Transform	hardware	-	177.787/ 195.4 MHz	-	-	-
[39]	CS-DAE	hardware	1.65 W	-	-	10.5	-
[40]	FIR filter	hardware	89 mW	50 MHz	-	-	6.5813

(*) This estimate corresponds to the power dissipated by the Nios core and the On-Chip memory, not to all the FPGA board components

The proposed architecture provides real-time ECG signal filtering since 0.41 ms is very short compared to the 2.33 ms, which is the acquisition rate of the signals of MIT-BIH Arrhythmia. On the other side, the low consumption of the proposed system in terms of hardware enables the integration of further ECG processing, such as feature extraction and classification, for a complete diagnosis in a single electronic chip.

Furthermore, the low consumption of our architecture due to low hardware needs and the low processing frequency reduces the need for a regular power supply. Consequently, this raises its long-term availability inside and outside hospitals.

5. CONCLUSION

This research paper discussed the development of a real-time FPGA-based embedded system for the DWT-ADTF method. By reason of its five-stage pipeline, high flexibility, and the extensibility of its ISA, the Nios-V soft processor was used. Under a processing frequency of 50 MHz, the five stages of the Nios core pipeline were effectively exploited through the proposed C language-based software architecture. The ECG data was loaded and processed within a sample-by-sample and shifting rather than a frame-by-frame approach. Thus, the ECG signal is preprocessed in real-time, and no permanent data storage is needed.

We verified the architecture's effectiveness, robustness, performance, and efficiency using various noise types and different ECG signals of the MIT-BIH Arrhythmia database. This was quantified using dedicated benchmarks, including the MSE, PRD, SNRimp, hardware, power, and time consumption, as well as the processing frequency, where the results were compared to other related works. The outcomes of the proposed architecture were promising, as it gave competitive SNR improvement and PRD, while its MSE values were the lowest. The architectural metrics proved the low consumption of our proposed system in terms of time, hardware, and power, where its results outperformed existing architectures. Under a processing frequency of 50 MHz, our architecture implemented less hardware, consuming only 0.41 ms of execution time and 0.97 mW of power. This enables the integration of further treatments in the same electronic chip, namely, feature extraction and disease classification, and raises the long-term availability of such service inside and outside hospitals. Consequently, the proposed architecture represents an appropriate denoising solution that can be integrated into ECG monitoring systems.

The achieved execution time is significant and largely sufficient for denoising task accomplishment. However, more processing stages are required for a complete heart state diagnosis. This means that the denoising time needs to be further reduced so that other processing can be added later. Therefore, and thanks to the extensibility of the Nios-V ISA, we look to accelerating the current architecture. The acceleration will concern the DWT part since it is the most time-consuming process of the whole system. Furthermore, the current architecture deals only with high-frequency noise. Therefore, the denoising effectiveness of our architecture will be improved so that low-frequency noise will also be suppressed.

FUNDING

No particular grant from any governmental or private funding organization was received for this study.

CONFLICT OF INTEREST

The Authors declare no conflict of interest.

DATA AVAILABILITY

The MIT-BIH Arrythmia database on the PhysioNet website (https://physionet.org) has publicly accessible data supporting the findings of this study.

ACKNOWLEDGMENT

The corresponding author expresses her acknowledgment to each one of the co-authors for his participation leading to the success of the research presented by this manuscript.

DECLARATION OF THE USE OF AI TOOLS

The authors used QuillBot, version 4.69.0, for correcting the grammatical errors, and no AI tool was used for the writing. All AI-assisted content was reviewed and edited by the authors, who take full responsibility for the integrity and accuracy of this work. No AI tool is listed as an author, and AI tools are not cited as primary sources.

REFERENCES

[1] M. A. Serhani, H. T. El Kassabi, H. Ismail, A. N. Navaz, "ECG Monitoring Systems: Review, Architecture, Processes, and Key Challenges", Sensors, Vol. 20, 2020. doi: 10.3390/s20061796.

[2] A. I. Paganelli, A. G. Mondejar, A. C. da Silva, G. Silva-Calpa, M. F. Teixeira, F. Carvalho, A. Raposo, M. Endler, "Real-time Data Analysis in Health Monitoring Systems: A Comprehensive Systematic Literature Review", Journal of Biomedical Informatics, Vol. 127, 2022, pp. 104-109. doi: 10.1016/j.jbi.2022.104009.

[3] M. Vozda, M. Cerny, "Methods for derivation of orthogonal leads from 12-lead electrocardiogram: A review", Biomedical Signal Processing and Control, Vol. 19, 2015, pp. 23-34. doi: 10.1016/j.bspc.2015.03.001

[4] N. Faruk *et al*. "A comprehensive survey on low-cost ECG acquisition systems: Advances on design specifications, challenges and future direction", Biocybernetics and Biomedical Engineering, Vol. 41, 2021, pp. 474-502. doi: 10.1016/j.bbe.2021.02.007.

[5] V. A. Ardeti, V. R. Kolluru, G. T. Varghese, R. K. Patjoshi, "An overview on state-of-the-art electrocardiogram signal processing methods: Traditional to AI-based approaches", Expert Systems with Applications, Vol. 217, 2023, pp. 119-561. doi: 10.1016/j.eswa.2023.119561.

[6] S. Luo, P. Johnston, "A review of electrocardiogram filtering", Journal of Electrocardiography, Vol. 43, 2010, pp. 486-496. doi: 10.1016/j.jelectrocard.2010.07.007.

[7] S. Chatterjee, R. S. Thakur, R. N. Yadav, L. Gupta, D. K. Raghuvanshi, "Review of noise removal techniques in ECG signals", IET Signal Processing, Vol. 14, No. 9, 2020, pp. 569-590. doi: 10.1049/iet-spr.2020.0104.

[8] S. Cuomo, G. De Pietro, R. Farina, A. Galletti, G. Sannino, "A Novel O(n) Numerical Scheme for ECG Signal Denoising", Procedia Computer Science, Vol. 51, 2015, pp. 775-784. doi: 10.1016/j.procs.2015.05.198.

[9] M. Liu, H. Hao, P. Xiong, F. Lin, Z.G. Hou, X. Liu, "Constructing a Guided Filter by Exploiting the Butterworth Filter for ECG Signal Enhancement", Journal of Medical and Biological Engineering, Vol. 38, 2017, pp. 980-992. doi: 10.1007/s40846-017-0350-1.

[10] M. Blanco-Velasco, B. Weng, K. E. Barner, "ECG signal denoising and baseline wander correction based on the empirical mode decomposition", Computers in Biology and Medicine, Vol. 38, 2007, pp. 1-13. doi: 10.1016/j.compbiomed.2007.06.003.

[11] W. Huang, N. Cai, W. Xie, Q. Ye, Z. Yang, "ECG Baseline Wander Correction Based on Ensemble Empirical Mode Decomposition with Complementary Adaptive Noise", Journal of Medical Imaging and Health Informatics, Vol. 5, 2015, pp. 1796-1799. doi: 10.1166/jmihi.2015.1647.

[12] M. A. Colominas, G. Schlotthauer, M. E. Torres, "Improved complete ensemble EMD: A suitable tool for biomedical signal processing", Biomedical Signal Processing and Control, Vol. 14, 2014, pp. 19-29. doi: 10.1016/j.bspc.2014.06.009.

[13] L. Yao, Z. Pan, "A new method based CEEMDAN for removal of baseline wander and powerline interference in ECG signals", International Journal for Light and Electron Optics, Vol. 223, 2020, pp. 165-566. doi: 10.1016/j.ijleo.2020.165566.

[14] O. E. B'charri, R. Latif, K. Elmansouri, A. Abenaou, W. Jenkal, "ECG signal performance de noising assessment based on threshold tuning of dual tree wavelet transform", Biomedical Engineering Online, Vol. 16, 2017, pp. 16-26. doi: 10.1186/s12938-017-0315-1.

[15] W. Jenkal, R. Latif, A. Toumanari, A. Dliou, O. E. B'charri, F. M.R. 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, 2016, pp. 499-508. doi: 10.1016/j.bbe.2016.04.001.

[16] W. Jenkal, R. Latif, M. Laaboubi, "ECG Signal Denoising Using an Improved Hybrid DWT ADTF Approach", Cardiovascular Engineering and Technology, Vol. 15, 2023, pp. 77-94. doi: 10.1007/s13239-023-00698-8.

[17] A. Kumar, H. Tomar, V. K. Mehla, R. Komaragiri, M. Kumar, "Stationary wavelet transform based ECG signal denoising method", ISA Transactions, Vol. 114, 2021, pp. 251-262. doi: 10.1016/j.isatra.2020.12.029.

[18] S. Elouaham, A. Dliou, W. Jenkal, M. Louzazni, H. Zougagh, S. Dlimi, "Empirical Wavelet Transform Based ECG Signal Filtering Method", Journal of Electrical and Computer Engineering, 2024. doi: 10.1155/2024/9050909.

[19] P. Bing, W. Liu, Z. Zhang, "DeepCEDNet: An Efficient Deep Convolutional Encoder-Decoder Networks for ECG Signal Enhancement", IEEE Access, Vol. 9, 2021, pp. 56699-56708. doi: 10.1109/ACCESS.2021.3072640.

[20] A. Rasti-Meymandi, A. Ghaffari, "A Deep Learning-Based Framework for ECG Signal Denoising based on Stacked Cardiac Cycle Tensor", Biomedical Signal Processing and Control, Vol. 71, 2022, pp. 103-275. doi: 10.1016/j.bspc.2021.103275.

[21] B. Peccerillo, M. Mannino, A. Mondelli, S. Bartolini, "A Survey on Hardware Accelerators: Taxonomy, Trends, Challenges, and Perspectives", Journal of Systems Architecture, Vol. 129, 2022, pp. 102-561. doi: 10.1016/j.sysarc.2022.102561.

[22] M. S. Diab, S. A. Mahmoud, "Survey on Field Programmable Analog Array Architectures Eliminating Routing Network", IEEE Access, Vol. 8, 2020, pp. 220779-220794. doi: 10.1109/ACCESS.2020.3043292.

[23] W. Jenkal, R. Latif, A. Toumanar, A. Elouardi, A. Hatim, O. EL'Bchari, "Real-Time Hardware Architecture of the Adaptive Dual Threshold Filter Based ECG Signal Processing", Journal of Theoretical and Applied Information Technology, Vol. 96, No. 14, 2018.

[24] W. Jenkal, R. Latif, A. Elouardi, S. Mejhoudi, "FPGA Implementation of the Real-Time ADTF process using the Intel-Altera DE1 Board for ECG signal Denoising", Proceedings of the 4th World Conference on Complex Systems, Ouarzazate, Morocco, 22-25 April 2019. doi: 10.1109/ICoCS.2019.8930780.

[25] S. Mejhoudi, R. Latif, A. Saddik, W. Jenkal, "Speeding up an Adaptive Filter based ECG Signal Preprocessing on Embedded Architectures", International Journal of Advanced Computer Science and Applications, Vol. 12, 2021. doi: 10.14569/IJACSA.2021.0120544.

[26] B. Bendahane, W. Jenkal, M. Laaboubi, B. Nassiri, "Low-Frequency and Low-Power Optimized FPGA Architecture of Adaptive Dual Threshold Filter for Real-Time Electrocardiogram Signal Denoising", Jordan Journal of Electrical Engineering, Vol. 11, 2025, pp. 470-490. doi: 10.5455/jjee.204-1759139835.

[27] R. Mittal, A. A. Prince, S. Nalband, F. Robert, A. R. J. Fredo, "Low-Power Hardware Accelerator for Detrending Measured Biopotential Data", IEEE Transactions on Instrumentation and Measurement, Vol. 70, 2020. doi: 10.1109/TIM.2020.3018235.

[28] M. Chandra, P. Goel, A. Anand, A. Kar, "Design and analysis of improved high-speed adaptive filter architectures for ECG signal denoising", Biomedical Signal Processing and Control, Vol. 63, 2021, pp. 102-221. doi: 10.1016/j.bspc.2020.102221.

[29] K. Das, S. Pradhan, "An efficient hardware realization of EMD for real-time signal processing applications", International Journal of Circuit Theory and Applications, Vol. 48, 2020, pp. 2202-2218. doi: 10.1002/cta.2860.

[30] K. Das, S. N. Pradhan, "Field-programmable gate array-based design for real-time computation of ensemble empirical mode decomposition", International Journal of Circuit Theory and Applications, Vol. 49, 2021, pp. 2312-2328. doi: 10.1016/j.micpro.2022.104549.

[31] A. N. Elbedwehy, A. M. El-Mohandes, A. Elnakib, M. E. Abou-Elsoud, "FPGA-based reservoir computing system for ECG denoising", Microprocessors and Microsystems, Vol. 91, 2022, pp. 104-549. doi: 10.1016/j.micpro.2022.104549.

[32] K. Tripathi, H. Sohal, S. Jain, "Design and Implementation of Robust Low Power ECG Pre-processing Module", IETE Journal of Research, Vol. 2716-2722, No. 4, 2020, pp. 2716-2722. doi: 10.1080/03772063.2020.1725660.

[33] P. Goel, M. Chandra, A. Anand, A. Kar, "An improved wavelet-based signal-denoising architecture with less hardware consumption", Applied Acoustics, Vol. 156, 2019, pp. 120-127. doi: 10.1016/j.apacoust.2019.07.013.

[34] S. Janwadkar, R. Dhavse, "ASIC implementation of ECG denoising FIR filter by using hybrid Vedic-Wallace tree multiplier", International Journal of Circuit Theory and Applications, Vol. 52, No. 4, 2023, pp. 1621-1646. doi: 10.1002/cta.3844.

[35] S. Mejhoudi, R. Latif, W. Jenkal, A. Saddik, A. Elouardi, "Hardware Architecture for Adaptive Dual Threshold Filter and Discrete Wavelet Transform based ECG Signal Denoising", International Journal of Advanced Computer Science and Applications, Vol. 12, No. 11, 2021. doi: 10.14569/IJACSA.2021.0121106.

[36] B. Bendahane, W. Jenkal, M. Laaboubi, "Hardware Software Co-Design Approach for ECG Signal Analysis", Proceedings of Advances in Machine Intelligence and Computer Science Applications, Khouribga, Morocco, 23-25 February 2023, pp. 199-210. doi: 10.1007/978-3-031-29313-9_18.

[37] M. Hossain, S. K. Bashar, J. Lazaro, N. Reljin, Y. Noh, K. H. Chon, "A Robust ECG Denoising Technique Using Variable Frequency Complex Demodulation", Computer Methods and Programs in Biomedicine, Vol. 200, 2020, pp. 105-856. doi: 10.1016/j.cmpb.2020.105856.

[38] D. Zhang, S. Wang, F. Li, S. Tian, J. Wang, X. Ding, R. Gong, "An Efficient ECG Denoising Method Based on Empirical Mode Decomposition, Sample Entropy, and Improved Threshold Function", Wireless Communications and Mobile Computing, Vol. 2020, 2020. doi: 10.1155/2020/8811962.

[39] L. Shin-Chi, W. Szu-Ting, M. S. M. Salahuddin, L. Jia-He, H. Shih-Chang, C. Chuan-Yu, S. Ming-Hwa, "VLSI Architecture Design for Compact Shortcut Denoising Autoencoder Neural Network of ECG Signal", IEEE Transactions on Circuits and Systems, Vol. 72, No. 4, 2025, pp. 1621-1633. doi: 10.1109/TCSI.2025.3533544.

[40] P. Bhaskar, M. Uplane, "FPGA Based Notch Filter to Remove PLI Noise from ECG", International Journal on Recent and Innovation Trends in Computing and Communication, Vol. 3, No. 4, 2015, pp. 2246-2250. doi: 10.17762/ijritcc.v3i4.4221.