Design of DNN-Based Low-Power VLSI Architecture to Classify Atrial Fibrillation for Wearable Devices

Rushik Parmar, Meenali Janveja, *Student Member, IEEE*, Jan Pidanic, *Senior Member, IEEE*, and Gaurav Trivedi, *Member, IEEE*

***Abstract*— Atrial fibrillation (AF) is a recurrent and life-threatening disease leading to rapid growth in the mortality rate due to cardiac abnormalities. It is challenging to manually diagnose AF using electrocardiogram (ECG) signals due to complex and varied changes in its characteristics. In this article, for the first time, an end-to-end edge-enabled machine learning-based VLSI architecture is proposed to classify ECG excerpts having AF from normal beats. Researchers have found that abnormal atrial activity is confined to the low-frequency range through the decades. Therefore, in the proposed work, this frequency band is directly analyzed for AF detection, which has not previously been discussed. The proposed architecture is implemented using 180-nm bulk CMOS technology consuming 11.098 μW at 25 kHz and exhibits an accuracy of 92.37% for class-oriented classification and 81.60% for subject-oriented classification. The low-power realization of the proposed design, as compared to the state-of-the-art methods, makes it suitable to be used for wearable devices.**

***Index Terms*— Application specific integrated circuit (ASIC), atrial fibrillation (AF), deep neural network (DNN), electrocardiogram (ECG) signal, wavelet transform, wearable devices.**

I. Introduction

ATRIAL fibrillation (AF) is the most common type of cardiac arrhythmia due to rising incidences and prevalence worldwide. AF is an abnormal heart rhythm caused due to the irregular contraction of the atria. Moreover, the likelihood of developing AF increases with age [1]. Contemporary research shows that around 30% of patients with ischemic heart disease are also diagnosed with AF [2]. AF increases the risk of other heart-related diseases, such as congenital heart disease, myocardial infarction, congestive heart failure, etc. AF also can lead to noncardiac diseases, viz. cognitive dysfunction, depression, and chronic kidney disease making a timely diagnosis of AF imperative to increase life expectancy [3], [4]. We aim to develop biomedical signal processing architectures for wearable or hand-held devices that can continuously analyze and detect AF in real time.

Previously reported research primarily focuses on the detection and classification of different types of cardiac arrhythmia as per Association for the Advancement of Medical Instrumentation (AAMI) standards, viz. detection of normal beat (N), supraventricular ectopic beat (SVEB), ventricular ectopic beat (VEB), fusion beat (F), and unknown beat (U). It is important to note that previous works have not included AF while classifying arrhythmia. Massachusetts Institute of Technology-Beth Israel Hospital (MIT-BIH) arrhythmia database [23] is one of the standard datasets used by researchers for detecting and classifying different types of cardiac arrhythmia. It is worth mentioning that the previously mentioned dataset lacks subjects with AF episodes and consists of 47 subjects, out of which only eight have suffered from AF episodes. The MIT-BIH atrial fibrillation (AF) database [19] was released to support research in AF classification, which consists of 25 ECG recordings of subjects diagnosed with AF. Our previous study [24] targeted real time classification of cardiac arrhythmia but did not include classification of AF, as diagnosing AF is challenging, especially at an early stage due to its asymptomatic nature. Note that AF remains undetected during the diagnosis with conventional ECG monitoring devices. The current method of diagnosis includes a medical expert inspecting and interpreting ECG data, which is a tedious and time-consuming task. Moreover, the scarcity of medical expertise in developing countries increases the mortality rate. Therefore, it is imperative to design smart wearable devices equipped with artificial intelligence to analyze ECG for timely detection of AF without any medical expert intervention. It is important to mention that these devices should have low-power consumption, low computations, and a high detection accuracy to operate in real time while avoiding false alarms and unnecessary visits to the hospital.

In recent years, researchers have used two approaches to detect AF. The first approach involves using handcrafted features from ECG data like a spectrogram [8], P-wave morphology [9], RR-interval series [22], entropy [22], and heart rate variability (HRV) [10]. With these features, studies have also explored the use of dominant atrial frequency [5], [6], [7] to detect AF. It is important to note that feature extraction plays a vital role in determining the computation

Manuscript received 15 September 2022; revised 21 December 2022; accepted 6 January 2023. This work was supported in part by Electronics and ICT Academy at IIT Guwahati funded by the Ministry of Electronics and Information Technology, India and in part by the Programme INTER-EXCELLENCE (LTAIN19100) funded by the Ministry of Education, Youth and Sports, Czech Republic, "Artificial Intelligence Enabled Smart Contact-less Technology Development for Smart Fencing" under Project LTAIN19100. *(Corresponding author: Rushik Parmar.)*

Rushik Parmar, Meenali Janveja, and Gaurav Trivedi are with the Department of Electronics and Electrical Engineering, IIT Guwahati, Guwahati 781039, India (e-mail: rushik_parmar@iitg.ac.in; meena176102001@iitg.ac.in; trivedi@iitg.ac.in).

Jan Pidanic is with the Department of Electrical Engineering, Faculty of Electrical Engineering and Informatics, University of Pardubice, 532 10 Pardubice, Czech Republic (e-mail: jan.pidanic@upce.cz).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TVLSI.2023.3236530.

Digital Object Identifier 10.1109/TVLSI.2023.3236530

cost of the classifier. Complex features, such as spectrogram, entropy, power spectral density, and root mean square demand higher resources leading to higher area utilization and power consumption. The second approach relies on the capabilities of ML models and directly uses raw ECG data to detect AF events [21]. Although ML algorithms can detect different patterns in a signal with high accuracy, they have various pitfalls, including high-computational costs and increased sensitivity to noise. This makes using raw data as a feature vector for ML model inefficient in smart wearable devices. On the contrary, a feature-based approach with selected features and an efficient feature extraction methodology can drastically reduce model size and complexity. This helps developers to reduce computations, realizing low-power wearable devices. With this, it is important to mention that previously reported the state-of-art methods primarily focus on software implementation of the classifier using complex features. These methods exhibit higher area utilization and power consumption. Andersson et al. [22] have proposed an application specific integrated circuit (ASIC) for AF detection. The model proposed in [22] accepts an RR-interval series as input to extract several features and then uses a threshold detector to classify a beat as normal or AF. However, this model does not extract the RR-interval series from ECG data and relies on an external processing unit for ECG data processing. Moreover, the detector uses nontunable hard-coded thresholds, which might not adapt to varied ECG signals in real-world applications. Therefore, the need of the hour is an edge-enabled low-power and computationally efficient VLSI architecture of an AF classifier. Hence, to overcome the above-mentioned issues, we propose an end-to-end AF classifier that accepts raw ECG data to extract wavelet coefficients using a simplified technique to classify AF beats from the normal beats.

The rest of this article is organized as follows. Section II describes the contribution of this article. Section III depicts the implementation details of feature extraction and the machine learning model. Experimental analysis and results are presented in Section IV. Finally, Section V concludes the manuscript.

II. Novelty of the Proposed Work

1) In this article, for the first time, an end-to-end VLSI architecture is proposed for AF classification. The previously reported the state-of-the-art methods were primarily implemented on a software platform or relied on an external processing unit for ECG data processing [22]. The proposed classifier accepts raw ECG data for analysis and classification of AF, making it an end-to-end classifier.
2) The proposed classifier uses low-frequency wavelet coefficients for AF classification instead of a complete spectrum. The use of these coefficients reduces the size of the feature vector, which in turn reduces the complexity and size of the machine learning model.
3) A novel wavelet transform approach using integer Haar wavelet is proposed in this article. It simplifies the implementation of wavelet transform from Mallat's decomposition to simple operations, such as add, subtract, and shift. This significantly reduces the computation cost.
4) The neural network architecture employs an improved Mitchell's multiplier. It uses two error reduction techniques to reduce the average error of Mitchell's algorithm from 3.88% to 1.79%.

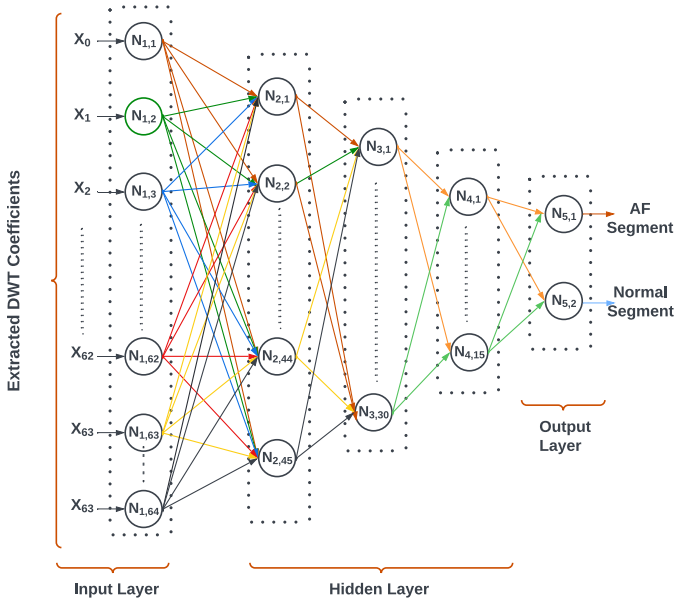


Fig. 1. Proposed neural network for AF classification. $N_{i,j}$ represents jth neuron of ith layer. X_i depicts input feature.

III. Proposed Design of the AF Classifier

A typical system for the detection of AF using ECG consists of two blocks. The first block is the feature extraction block, and the second is the classifier block, which can process features and perform the classification. Fig. 1 showcases the proposed neural network, and Fig. 2 illustrates the complete architecture of the proposed AF classification algorithm. The details of the work proposed in this manuscript are described below.

A. Feature Extraction Block

Wearable applications require low-energy dissipation in both idle and active modes. Idle energy is dominated by the leakage current drawn by memories, whereas active energy is minimized by reducing complexity. Thus, an efficient feature extraction block is proposed in this article, which does not require storing the entire ECG samples. Moreover, the computational complexity of the feature extraction block is reduced by simplifying the wavelet transform, as explained in the subsequent section. The variations in the ECG due to abnormal atrial activity are usually confined to the signal's low-frequency range (less than 12 Hz) [12]. Discrete wavelet transform (DWT) is used for the analysis of the low-frequency ECG signal components. It is preferred over Fourier transform due to the salient features mentioned below.

1) Adaptive time-frequency windows.
2) Lower aliasing distortion for signal processing applications.
3) Reduced computational complexity.
4) Efficient VLSI implementation.

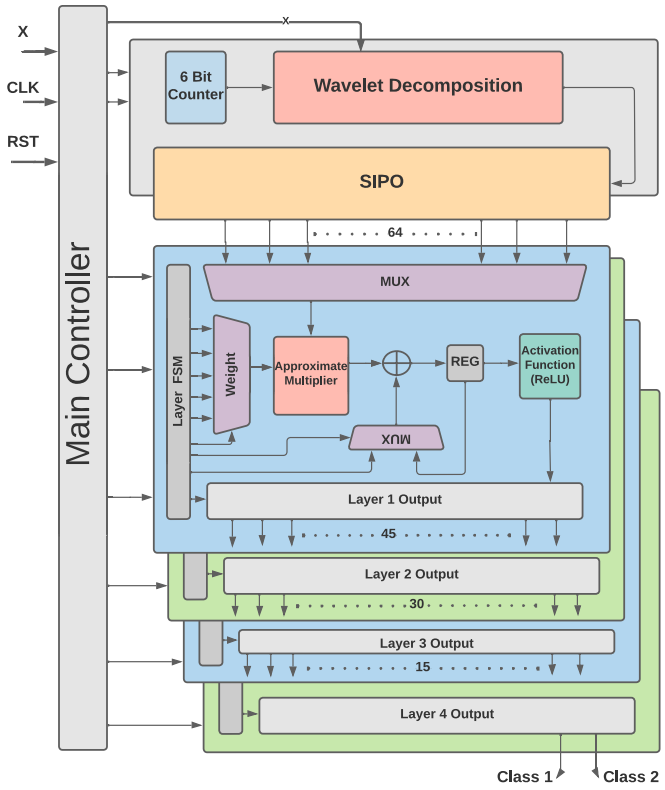


Fig. 2. Proposed architecture of AF classifier.

DWT is a multiresolution analysis tool initially used for image and speech compression. It is now widely used to analyze nonstationary signals, such as ECG [15]. Using DWT, a signal can be decomposed into different subbands to extract local spectral and temporal information simultaneously using a mother wavelet. Among all the mother wavelets, "integer Haar" is the simplest wavelet function. Our previous work proposed using DWT with "integer Haar" as a mother wavelet [15] for the following reasons: First, it has the simplest filter coefficients, making it efficient for hardware implementation. Second, it is fast and memory efficient. Third, it does not exhibit edge effects. Hence, in the proposed work, the analysis of low-frequency ECG signal components, which contain critical information about AF, is performed using DWT employing "integer Haar" as the mother wavelet. Using DWT decomposition with integer Haar, we extract wavelet coefficients, which provide information about various frequency bands. Later, the wavelet coefficients of the ECG signal in the relevant frequency band, having information about AF, are taken as an input feature vector for the deep neural network (DNN) to classify ECG waves as normal or AF.

To extract decomposed wavelet coefficients in different frequency bands, the input signal needs to be passed through a chain of low-pass and high-pass filters as per Mallat's decomposition algorithm [16] shown in Fig. 3. The output of each level is required to be downsampled by two to remove redundant information. The coefficients of low- and high-pass filters are calculated according to the mother wavelet. However, the integer Haar wavelet, an approximation to the Haar

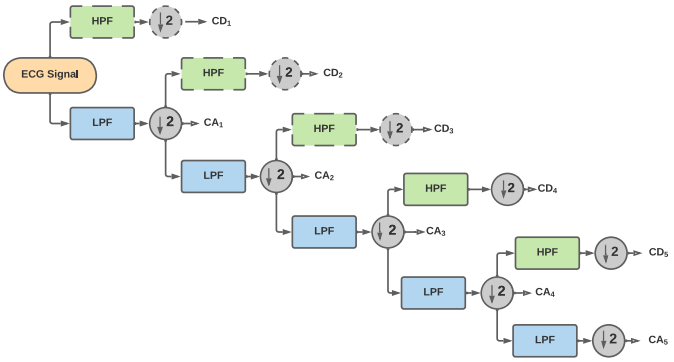


Fig. 3. Mallat's wavelet decomposition method.

wavelet [15], is utilized among all the wavelet functions in the filters. The transfer functions of the low-pass and high-pass filters for Haar wavelet are illustrated by the following:

$$\mathrm{CA}[n] = \frac{1}{\sqrt{2}}X[2n] + \frac{1}{\sqrt{2}}X[2n+1] \tag{1}$$

$$\mathrm{CD}[n] = \frac{1}{\sqrt{2}}X[2n] - \frac{1}{\sqrt{2}}X[2n+1]. \tag{2}$$

Although the Haar wavelet is the simplest among all the mother wavelets, but it involves floating-point arithmetic, making it computationally expensive. To overcome this, integer Haar wavelet, an approximation to haar wavelet is proposed [15]. Further, the transfer functions of the low-pass and high-pass filters, CA[n] and CD[n] of the integer Haar wavelet, can be stated by the following:

$$\mathrm{CA}[n] = \left\lfloor \frac{1}{2}X[2n] + \frac{1}{2}X[2n+1] \right\rfloor \tag{3}$$

$$\mathrm{CD}[n] = X[2n] - X[2n+1]. \tag{4}$$

It is to mention that (3) and (4) can be implemented using simple add and right shift ($\gg$1) operations represented by (5). This avoids typical floating point arithmetic utilized in the conventional Haar wavelet transform

$$\mathrm{CA}[n] = (X[2n] + X[2n+1]) \gg 1. \tag{5}$$

As mentioned above, the approximation of filter coefficients of the conventional Haar wavelet to the integer Haar wavelet helps in reducing the computational complexity, making the integer Haar wavelet an adequate transform for extracting AF frequency bands. In the proposed work, an ECG signal of a length of 500 samples is considered at a time. Its wavelet coefficients up to the 5th scale are calculated, as they contain the frequency components essential for AF detection. The conventional implementation of Mallat's algorithm up to the 5th scale requires the realization of seven filters, which is resource intensive. Thus, in the proposed work, we present filter-less feature extraction method employing a simplified wavelet decomposition scheme described below.

Proposed Wavelet Decomposition: Several VLSI architectures for the discrete wavelet transform have been proposed in the literature [27], [28]. Parhi and Nishitani [27] have proposed a folded three-level DWT architecture for an efficient wavelet decomposition, which is shown in Fig. 4. It employs several

TABLE I
CALCULATION OF THE COEFFICIENTS USING INTEGER HAAR WAVELET

Input	Coefficient	j=0	j=1	j=2	j=k
x[n]	$CD_1[j]$	$x[0]-x[1]$	$x[2]-x[3]$	$x[4]-x[5]$	$x[2*k]-x[2*k+1]$
	$CA_1[j]$	$\frac{x[0]+x[1]}{2}$	$\frac{x[2]+x[3]}{2}$	$\frac{x[4]+x[5]}{2}$	$\frac{x[2*k]+x[2*k+1]}{2}$
CA_1	$CD_2[j]$	$\frac{x[0]+x[1]-x[2]-x[3]}{2}$	$\frac{x[4]+x[5]-x[6]-x[7]}{2}$	$\frac{x[8]+x[9]-x[10]-x[11]}{2}$	$\frac{\sum_{i=4*k}^{i=4*k+1} x[i]-\sum_{i=4*k+2}^{i=4*k+3} x[i]}{2}$
	$CA_2[j]$	$\frac{x[0]+x[1]+x[2]+x[3]}{4}$	$\frac{x[4]+x[5]+x[6]+x[7]}{4}$	$\frac{x[8]+x[9]+x[10]+x[11]}{4}$	$\frac{\sum_{i=4*k}^{i=4*k+3} x[i]}{4}$
CA_2	$CD_3[j]$	$\frac{\sum_{i=0}^{i=3} x[i]-\sum_{i=4}^{i=7} x[i]}{4}$	$\frac{\sum_{i=8}^{i=11} x[i]-\sum_{i=12}^{i=15} x[i]}{4}$	$\frac{\sum_{i=16}^{i=19} x[i]-\sum_{i=20}^{i=23} x[i]}{4}$	$\frac{\sum_{i=8*k}^{i=8*k+3} x[i]-\sum_{i=8*k+4}^{i=8*k+7} x[i]}{4}$
	$CA_3[j]$	$\frac{\sum_{i=0}^{i=7} x[i]}{8}$	$\frac{\sum_{i=8}^{i=15} x[i]}{8}$	$\frac{\sum_{i=16}^{i=23} x[i]}{8}$	$\frac{\sum_{i=8*k}^{i=8*k+7} x[i]}{8}$
CA_3	$CD_4[j]$	$\frac{\sum_{i=0}^{i=7} x[i]-\sum_{i=8}^{i=15} x[i]}{8}$	$\frac{\sum_{i=16}^{i=23} x[i]-\sum_{i=24}^{i=31} x[i]}{8}$	$\frac{\sum_{i=32}^{i=39} x[i]-\sum_{i=40}^{i=47} x[i]}{8}$	$\frac{\sum_{i=16*k}^{i=16*k+7} x[i]-\sum_{i=16*k+8}^{i=16*k+15} x[i]}{8}$
	$CA_4[j]$	$\frac{\sum_{i=0}^{i=15} x[i]}{16}$	$\frac{\sum_{i=16}^{i=31} x[i]}{16}$	$\frac{\sum_{i=32}^{i=47} x[i]}{16}$	$\frac{\sum_{i=16*k}^{i=16*k+15} x[i]}{16}$
CA_4	$CD_5[j]$	$\frac{\sum_{i=0}^{i=15} x[i]-\sum_{i=16}^{i=31} x[i]}{16}$	$\frac{\sum_{i=32}^{i=47} x[i]-\sum_{i=48}^{i=63} x[i]}{16}$	$\frac{\sum_{i=64}^{i=79} x[i]-\sum_{i=80}^{i=95} x[i]}{16}$	$\frac{\sum_{i=32*k}^{i=32*k+15} x[i]-\sum_{i=32*k+16}^{i=32*k+32} x[i]}{16}$
	$CA_5[j]$	$\frac{\sum_{i=0}^{i=31} x[i]}{32}$	$\frac{\sum_{i=32}^{i=63} x[i]}{32}$	$\frac{\sum_{i=64}^{i=95} x[i]}{32}$	$\frac{\sum_{i=32*k}^{i=32*k+31} x[i]}{32}$

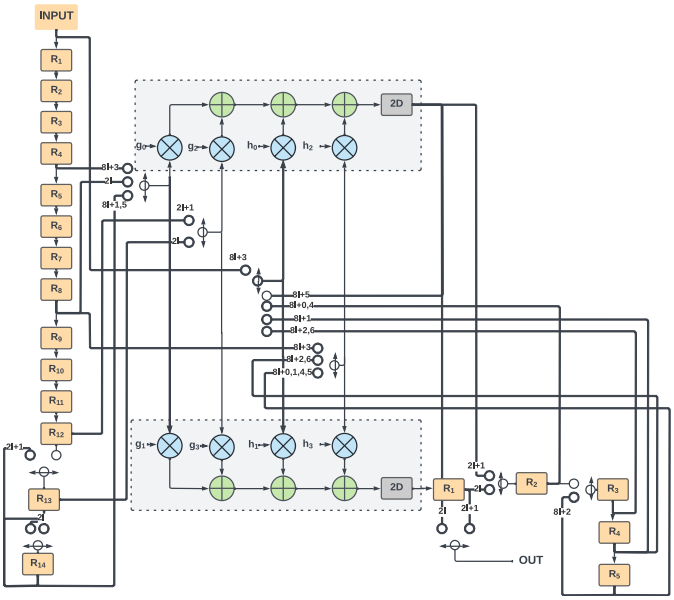


Fig. 4. Folded three-level wavelet decomposition architecture. "R_i" and "D" both represent word-level registers. "i" denotes the time allocation of the register "R_i" [27].

adders, multipliers, and registers, resulting in higher resource utilization. In the proposed work, instead of employing conventional decomposition architectures, a new architecture is depicted employing the integer Haar wavelet as the mother wavelet. This architecture is optimized using approximate and detailed coefficients of the integer Haar wavelet illustrated in Table I. Equations (6), (8), and (9) showcase the simplified 4th- and 5th-level wavelet coefficients

$$\mathrm{CD}_4[j] = \left(\sum_{k=16j}^{k=16j+7} x[k] - \sum_{k=16j+8}^{k=16j+15} x[k] \right) \gg 3 \tag{6}$$

$$\mathrm{CA}_4[j] = \left(\sum_{k=16j}^{k=16j+15} x[k] \right) \gg 4 \tag{7}$$

$$\mathrm{CD}_5[j] = \left(\sum_{k=32j}^{k=32j+15} x[k] - \sum_{k=32j+16}^{k=32j+31} x[k] \right) \gg 4 \tag{8}$$

$$\mathrm{CA}_5[j] = \left(\sum_{k=32j}^{k=32j+31} x[k] \right) \gg 5. \tag{9}$$

Utilizing the approximate and detailed coefficients shown in Table I, the proposed wavelet decomposition architecture is simplified by incorporating add, subtract, and shift operations. This eliminates the need for a clock divider network for down-sampling and the multipliers to extract the wavelet coefficients. Fig. 5 exhibits the proposed discrete wavelet decomposition implementation, in which coefficients at a specific level are estimated without utilizing the previous outputs. It employs three adders, two subtractors, three registers, three shifters, and five switches. It is to mention that the switches are realized as multiplexers on hardware, and the shifters are used to simplify the division operation. This architecture accepts raw data and provides the approximate and detailed coefficients, including CD_4, CD_5, and CA_5.

The design shown in Fig. 4 implements level-three DWT while the proposed implementation in Fig. 5 realizes level-five DWT with reduced hardware utilization. The implementation

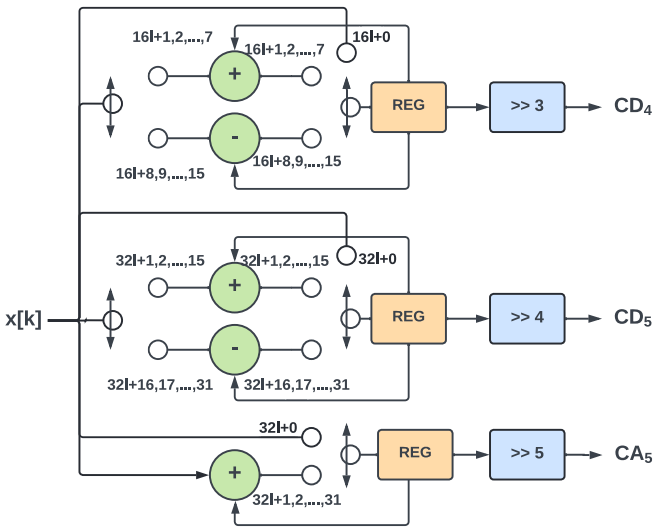


Fig. 5. Proposed wavelet decomposition architecture. "REG" represents word-level register and "$\gg n$" represents a right shift by n operation.

depicted in Fig. 4 requires several multipliers, while the proposed design employs only shifting operations, thus reducing complexity. Estimating wavelet coefficients in this manner utilizes lesser registers without using downsampling circuit and multipliers. In our previous work [15], DWT using integer Haar wavelet is implemented as per Mallat's decomposition algorithm, which includes downsampling by two at every stage. This downsampling logic requires the realization of clock division. The present work further optimizes DWT implementation to eliminate the need for complex filters and clock division for downsampling. Table I showcases the detailed and approximate coefficients at every stage using the integer Haar wavelet.

Loh et al. [29] and Jobst et al. [30] have also used DWT for feature extraction block. Loh et al. [29] propose a digital signal processing accelerator for cardiac arrhythmia detection. It employs discrete wavelet transform with CNN to classify cardiac arrhythmia. The DWT-based feature extraction is performed using Daubechies (db) wavelets and is implemented as per Mallat's decomposition algorithm. In this design, each filter requires four multipliers. The proposed work does not utilize multipliers because the integer Haar wavelet uses division by 2, which can easily be implemented using a shift operation. The work presented in [29] also requires a downsampling logic at each stage. For subsampling in the DWT stages, Loh et al. [29] employ duty cycling of the clock signal, i.e., corresponding filter taps are gated every second or third cycle relative to their previous components. While our proposed work eliminates the need for the clock division and realizes (6), (8), and (9), representing the detailed coefficients at level 4, detailed coefficients at level 5, and approximate coefficients at level 5, respectively. Note that [29] requires level-4 wavelet coefficients for CA detection. The level-4 wavelet coefficients are extracted using five 4-tap filters and three downsampling units. This can easily be reduced to (6) and (7), which can be realized using add, subtract, and shift operations only.

A flexible hardware for ECG classification with configurable time-domain filters is proposed in [30]. The filters are used to extract features, which are then fed into a recurrent neural network for classification. The feature extraction module is a filter bank that has multiple stages with several filters each. Each stage performs a subsampling by a factor of 2 for data reduction. The implementation presented in [30] can be reused for low-frequency analysis, but it requires offline retraining of the model and updating the hardware parameters. With this, the filters are implemented as per Mallat's decomposition algorithm, which includes downsampling by two at each stage. This downsampling logic requires the implementation of clock division. On the contrary, in our proposed implementation, the complex filter and downsampling logic for Mallat's decomposition reduces to a counter-based shift, add, and subtract operations, eliminating clock division for downsampling. Our proposed design also requires less memory as approximate and detailed coefficients of each stage are directly a function of the input samples. Thus, using the proposed realization of DWT provides a threefold improvement. First, it reduces the overall latency of the system. Second, it reduces hardware complexity, and third, it also simplifies the synthesis of the clock tree network.

B. Implementation of DNN-Based AF Classifier

The proposed architecture of the DNN-based AF classifier is presented in Fig. 2. During the feature extraction, using CD_4, CD_5, and CA_5, 64 wavelet coefficients are obtained, which are fed as an 1-D input feature vector to a DNN to classify AF. These 64 wavelet coefficients cover frequency components of the ECG signal below 12 Hz. Note that for the ECG analysis, the samples need to be passed through a bandpass filter of 0.5–50 Hz [33] to filter out noise. The 4th- and 5th-level DWT coefficients used in the proposed work contain low-frequency information. Since the high-frequency wavelet coefficients are not utilized, the high-frequency noise does not propagate to the DNN classifier. In the hidden layers of the DNN, "ReLU" is utilized as an activation function, which not only reduces hardware complexity to a simple multiplexer but also addresses the gradient vanishing problem faced by other activation functions.

As we know, each layer of a DNN comprises neurons, which mainly consist of multiply and accumulate (MAC) units. Mathematically, a neuron is modeled using (10), where x_i, w_i, and b_i are the input to the neuron, the weight of each branch, and node bias, respectively,

$$y = \sum_{i=1}^{N} w_i x_i + b_i. \tag{10}$$

The complexity and hardware resource utilization of a DNN primarily depends on the number of network layers and nodes in each layer. Therefore, we utilize a minimal input feature vector of size 64 as per the requirement, which helps in optimizing the size of a DNN to $64 \times 45 \times 30 \times 15 \times 2$ after several epochs of training. After optimizing and training the network, this pre-trained network is implemented on hardware using Verilog HDL. Note that the signed floating-point arithmetic has a high resource utilization and power consumption

and also impacts the performance of a system. Therefore, the floating-point weights and biases obtained using a pre-trained network are converted to 18-bit fixed-point arithmetic in the proposed method. As we know, each neuron consists of a MAC unit, and multiplication is the most computationally intensive operation in any hardware design; therefore, to optimize the MAC unit of a neuron, an approximate multiplier (AM) is proposed in this article, which is described in Section III-C.

C. Proposed Optimized AM

Several designs of accurate and AMs are proposed in the literature. However, AMs have better power, area, and speed efficiency than accurate multipliers. Moreover, Mitchell's algorithm [17] is the most simplified approach for integer and fixed point multiplication among all the available methods of approximate multiplications.

Mitchell proposes an efficient AM, which utilizes a log multiplication property. It computes approximate multiplication by linearly approximating log and antilog values. In this method, first, the characteristic part of the approximate log is obtained by finding the position of the leading "1," i.e., the leftmost one in the binary sequence. Next, the remaining values are used as an approximate fractional part of the log. Later, these two operands are added, and the approximate antilog operation is performed to this summation, which generates an approximate product.

Consider an N bit integer B with bits $B_{n-1}, B_{n-2}, \ldots, B_0$, which is represented as $B = \sum_{i=0}^{N-1} 2^i b_i$. Assuming that leading one occurs at position k, where $(N-1) \geqslant k \geqslant 0$, B can be written as (11) without any loss of the accuracy

$$B = 2^k\left(1 + \sum_{i=0}^{k-1} 2^{i-k} b_i\right). \tag{11}$$

Equation (11) can further be represented as $B = 2^k(1+x)$, where $x = \sum_{i=0}^{k-1} 2^{i-k} b_i$. Hence, log of B can be described as (12), where k is essentially an integer representing characteristic value of log, and $\log_2(1+x)$ is the fractional part

$$\log_2 B = k + \log_2(1+x). \tag{12}$$

Since, $0 \leq x < 1$, the linear approximation of $\log_2 B$ can be depicted as (13), where function $\widetilde{\log}$ represents approximate binary log

$$\widetilde{\log_2} B = k + x. \tag{13}$$

Further, assuming there are two N-bits binary numbers B_1 and B_2 with leading ones at k_1 and k_2, they can be represented as $B_1 = 2^{k_1}(1+x_1)$ and $B_2 = 2^{k_2}(1+x_2)$. Thus, the approximate product can be represented as mentioned in the following:

$$\widetilde{\log_2}(\widetilde{P}) = k_1 + k_2 + x_1 + x_2 \tag{14}$$

where $\widetilde{P}$ represents approximate product. Later, to estimate approximate antilog of (14), "1" is added to the fractional part x and is scaled with respect to the characteristic part. It is to be noted that the fractional part should be in the range [0, 1). Since $0 \leq (x_1, x_2) < 1$, thus, in this case $0 \leq x_1 + x_2 < 2$.

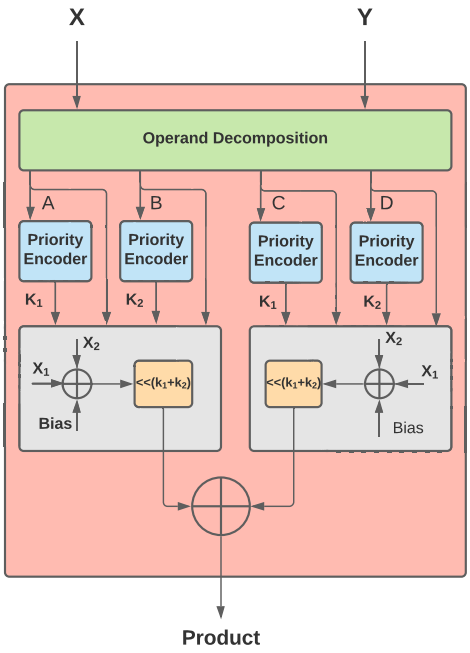


Fig. 6. Proposed architecture of AM.

Therefore, the approximate product is decomposed into two cases. First, $0 \leq x_1 + x_2 < 1$, which has no carry generating from the fractional part to the characteristic and second, $1 \leq x_1 + x_2 < 2$, which has a carry generating to the characteristic part. This can be expressed as follows:

$$\widetilde{P} = \begin{cases} 2^{k_1+k_2}(x_1 + x_2 + 1), & x_1 + x_2 < 1 \\ 2^{k_1+k_2+1}(x_1 + x_2), & x_1 + x_2 \geq 1. \end{cases} \tag{15}$$

Although the above-mentioned approximation reduces the entire multiplication to addition and shift operations, it introduces a significant error in the product obtained. Therefore, to obtain an efficient multiplier architecture for the MAC unit of a neuron, we propose an efficient error reduction technique and an optimal hardware implementation of (15). This obtains a more accurate result compared to the conventional Mitchell's algorithm. The multiplier architecture proposed in this article is described in Fig. 6. The error reduction technique utilized in the proposed method is described below.

1) Error Reduction Scheme: Here, we propose a two-step method to reduce error in the approximated product. First, a bias is calculated by averaging errors across the entire range of the fractional part x. Further, this average error is added to the approximate product improving its accuracy [25]. Error E in the approximate product is estimated by (16), where P is the logarithmic product of B_1 and B_2 and $\widetilde{P}$ is the approximate product obtained by Mitchell's algorithm

$$\begin{aligned} E &= \widetilde{P} - P \\ &= \begin{cases} -2^{k_1+k_2}(x_1 x_2), & x_1 + x_2 < 1 \\ -2^{k_1+k_2}(1 + x_1 x_2 - x_1 - x_2), & x_1 + x_2 \geq 1. \end{cases} \end{aligned} \tag{16}$$

Later, the average error (bias) is calculated using the following:

$$\begin{aligned} E_{\text{avg}} &= \frac{1}{(1-0)(1-0)} \int_0^1 \int_0^1 E dx_2 dx_1 \\ &= \int_0^1 \int_0^{1-x_1} -2^{k_1+k_2}(x_1 x_2) dx_2 dx_1 \\ &\quad + \int_0^1 \int_{1-x_1}^1 -2^{k_1+k_2}(1 + x_1 x_2 - x_1 - x_2) dx_2 dx_1 \\ &= -2^{k_1+k_2}(0.083333). \end{aligned} \tag{17}$$

It can be inferred that the average error (bias) is always negative and depends on k_1 and k_2. Bias needs to be shifted and added as per the position of the leading "1" in B_1 and B_2. It is to mention that the error got introduced in the product is due to the approximation of the fractional part. Thus, in the second step, a decomposition strategy similar to [18] is utilized in the proposed multiplier. To reduce the error, the fractional part x is decomposed into two small operands. The proposed decomposition scheme minimizes the fractional part and the error in the logarithm approximation. This decomposition methodology further decreases switching activity [18], reducing power. The proposed decomposition scheme is represented by (18), where $X = X_{n-1}, X_{n-2}, \ldots, X_0$ and $Y = Y_{n-1}, Y_{n-2}, \ldots, Y_0$ are two N-bit binary numbers, and A, B, C, and D are the decomposed operands depicted as follows:

$$\begin{aligned} A[i] &= X[i] \mid Y[i] \\ B[i] &= X[i] \;\&\; Y[i] \\ C[i] &= \overline{X[i]} \;\&\; Y[i] \\ D[i] &= X[i] \;\&\; \overline{Y[i]} \\ XY &= A * B + C * D. \end{aligned} \tag{18}$$

The implementation of the proposed multiplier is depicted in Fig. 6. To implement the AM, leading "1" finding module can be optimized and implemented in hardware as a simple priority encoder to reckon k_1 and k_2. Next, the binary numbers can be shifted and rearranged in the fixed point representation to get respective fractional parts x_1 and x_2. Finally, the approximate product is obtained by adding and shifting x_1, x_2, k_1, and k_2 according to (15).

2) Error Analysis of the Proposed Multiplier: In this section, the estimated error of the outputs of Mitchell's AM and the proposed AM is compared. The multipliers are numerically analyzed using Python and the error is calculated analyzing their outputs when same binary inputs are provided to both the multipliers. In this method, many 18-bit numbers are multiplied, and their products are represented by 36-bit numbers. Each product is stored in a 32 Byte number by default. The entire process generates $2^{18} \times 2^{18} = 68, 719, 476, 736$ multiplication products. Thus, the total memory used to store these products is $68, 719, 476, 736 \times 32 = 2048$ GB. We use the same space to save the error calculated. Additionally, extra memory is required for 3D surface plot rendering. We employ a computer having 256 GB RAM for the analysis of the proposed method. However, it also cannot plot the errors produced by Mitchell's algorithm, operand decomposition, and Operand Decomposition with bias addition for 18-bit multiplications due to memory requirement more than physically available. Therefore, we choose to plot errors produced by the multiplication of 10-bit numbers to study the behavior of the error produced, consuming 87% of the available memory.

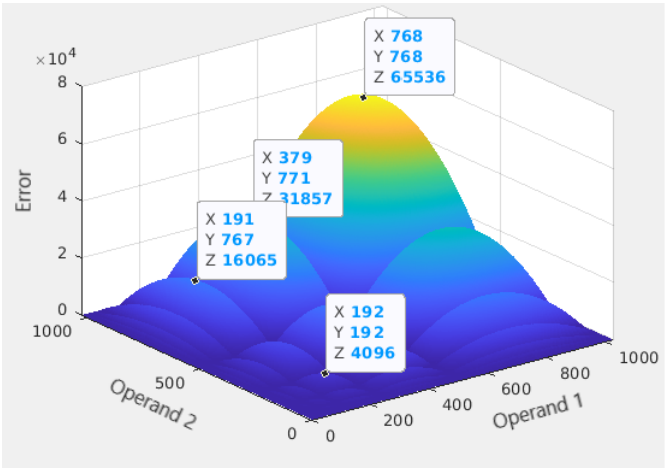


Fig. 7. Error in Mitchell's multiplication algorithm.

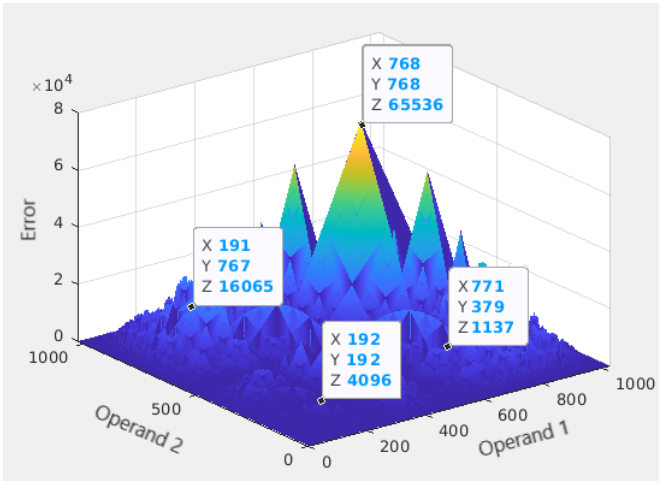


Fig. 8. Error correction using operand decomposition.

Here, the subsequent figures illustrate the error estimation, in which the x- and y-axes denote the inputs, and the z-axis represents the multiplier error. Fig. 7 showcases the error produced by Mitchell's algorithm. Fig. 8 depicts the error of Mitchell's algorithm when operand decomposition is used along with it. Fig. 9 illustrates the error of Mitchell's algorithm when it incorporates both the bias addition and operand decomposition. The average error for 18-bit multiplication using Mitchell's algorithm is 3.88%. Bias addition reduces the average error to 2.19%. Furthermore, by incorporating operand decomposition, the average error scales down to 1.79%. However, for certain inputs, the error might increase using the proposed multiplier, but the average error and max error reduce substantially. Thus, the proposed multiplier with an average error of 1.79% can be utilized in biomedical signal processing applications. The details of the realization of the proposed AF classifier are explained in the next section.

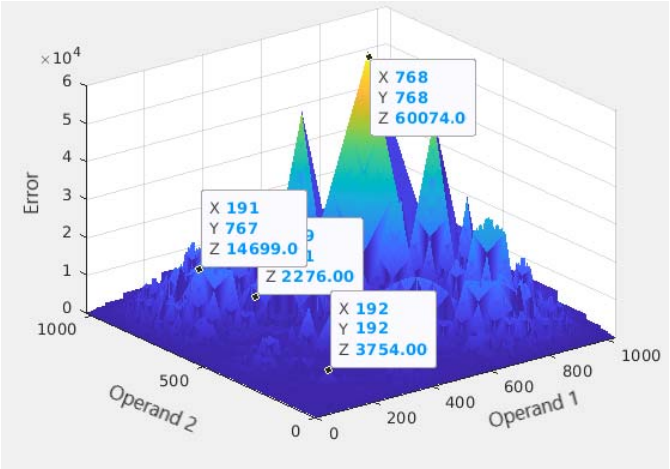


Fig. 9. Error correction using operand decomposition and bias addition.

IV. Results and Discussion

In this section, a detailed analysis of the implementation of the proposed work is presented.

A. Hardware Implementation

In this section, a detailed discussion is presented on the evaluation of the proposed design. In this study, ECG signals are acquired from the Physionet database. The AF and non-AF signals are extracted from Physionet's MIT-BIH AF database [19]. This dataset contains 25 ECG recordings sampled at 250 Hz with an approximate duration of 10 h. However, in this study, four recordings are omitted because two recordings ("00735" and "03665") are not available, and the other two ("04936" and "05091") have incorrect reference annotations as specified in the dataset. The dataset provides annotated data labeled normal and AF. The annotated normal and AF segments are used to create test and training data. Each segment is divided into subsegments using a sliding window of 500 samples. These subsegments are then partitioned into training and test sets as per two schemes (Experiment-1 and Experiment-2) to evaluate the performance metrics of the proposed classifier.

The class-oriented approach is utilized in Experiment-1. This approach divides the entire subsegments with AF and normal episodes into 20:80. Here, 20% of segments are used for testing, and the remaining 80% are utilized for training. Experiment-1 using integer Haar wavelet yields an average accuracy of 92.37%, an F1 score of 91.63, and sensitivity and specificity of 91.84 and 92.87. Further, the subject-oriented scheme is incorporated in Experiment-2, where training and test sets are entirely different. Out of the 21 subjects, subsegments from five subjects are used as testing data and the rest for training the model. Experiment-2 exhibits a more realistic estimate of the classifier's ability in practical scenarios. Further, test data always remain blind to the model in Experiment-2. It yields an average accuracy of 81.60% and an F1 score of 82.40, along with sensitivity and specificity of 82.06 and 81.14 using the integer Haar wavelet. Finally, the model generated using Experiment-2 is implemented further due to its realistic behavior.

TABLE II
FPGA Implementation of AF Classifier Architecture (Xilinx Virtex-7 FPGA)

Resource	Available	Utilization	Percentage Utilization(%)
LUT	303600	6332	2.08
SLICE-REG	607200	3162	0.52
F7MUX	151800	646	0.42
F8Mux	75900	179	0.23
IO's	600	23	3.83
BUFGCTRL	32	1	3.12

The proposed AF classifier is first implemented and tested using Python to validate its correctness. Later, the complete design is realized using Verilog HDL and is synthesized on a Xilinx Virtex–7 FPGA board [20] to verify its performance by testing it on multiple testcases. The resource utilization of the proposed architecture on the Xilinx Virtex–7 FPGA board is exhibited in Table II. It is observed that the proposed design utilizes 1.7% of the total available resources. It is the first end-to-end ML-based AF classifier on hardware. Since our goal is to realize an end-to-end solution for AF detection as an ASIC, no FPGA-specific power optimization techniques are employed while implementing the proposed design on FPGA. Therefore, the area and power of the proposed design are calculated using Synopsys IC Compiler and SCL 180-nm bulk CMOS PDKs, as reported in Table III

The complete design utilizes 1.18 mm^2 area and consumes 11.098-μW power at 1.98 V and 25 kHz. Since we utilize ECG excerpts having a sampling frequency of 250 Hz, it is observed that an operating frequency of 25 kHz is suitable for the proposed design to process ECG beats in real time. Placement and routing are performed using Synopsys IC compiler, and Fig. 10 presents the chip layout of the proposed classifier. A detailed comparative analysis of the proposed method with other implementations is given below.

B. Comparison With State-of-the-Art Methods

Table III presents the comparison of the proposed work with state-of-the-art methods. It is inferred from Table III that the proposed classifier yields higher average accuracy, F1 score, sensitivity, and specificity than the methods reported in [8], [10], and [12]. It is worth mentioning that Zihlmann et al. [8] use a complex CRNN with spectrogram, making both feature extraction and classification computationally expensive. While Lake and Moorman [10] and Alcaraz et al. [12] use complex features, including sample entropy, relative harmonic energy, and dominant atrial frequency, yield comparable performance. Couceiro et al. [6] use various features, including P-wave absence, HRV, Kullback Leibler divergence, and wavelet transform with 12 beats, to classify AF. However, this approach yields sensitivity and specificity of 93.80% and 96.09%, respectively. It will demand higher hardware resources for computational and memory complexity of the feature extraction block and analysis of 12 ECG complexes. This makes it suitable only for software platforms. Work proposed in [13]

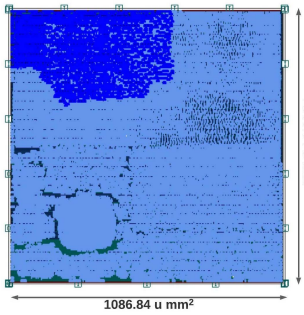
TABLE III
COMPARISION OF THE PROPOSED WORK WITH STATE-OF-THE-ART METHODS

Work / Parameter	[11]	[8]	[12]	[13]	[10]	[6]	[21]	[14]	[31]	[32]	[22]	Proposed Work	
Features Used	Stationary Wavelet Transform, PSD, log energy entropy	Spectogram	Sub band Sample entropy, Dominant Atrial Frequency, Relative Harmonic Energy	Wavelet Transform, Peak to average power Ratio, Log-energy entropy	Sample Entropy Estimate and mean heart beat interval	P wave absence, HRV, Kullback Leibler divergence, Wavelet Transform	Raw ECG Data	37 features including time domain, linear, nonlinear and frequency domain features	Turning Point Ratio (TPR), Root Mean Square of Successive Difference (RMSSD), and Shannon Entropy (SE)	63 time domain Features	RR interval series - turning point ratio, RMS of successive differences, and Shannon entropy	Wavelet Coefficients	
Classifier	ANN	CRNN	Min Error Rate Classifier	SVM	NA	NA	CNN	Random Forest	Threshold Detector	ANN	Threshold Detector	Deep Neural Network	
ECG Sample Window	10 sec	NA	1250	30 sec	12 Beats	12 Beats	2700	NA	128 RR interval	30 sec	128 RR interval	500	
Evaluation Scheme	COS	COS	COS	COS	COS	COS	COS	SOS	COS	COS	COS	COS	SOS
Implementation Platform	Hardware Software Co-design	Software	Software	Software	Software	Software	Software	Software	ASIC	ASIC	ASIC	ASIC	ASIC
Sensitivity	NA	NA	92.00	97.00	91.00	93.80	99.91	NA	94.90	89.00	90.60	91.84	82.06
Specificity	NA	NA	80.00	97.10	94.00	96.09	99.99	NA	95.80	100.00	97.60	92.87	81.14
Accuracy	95.30	82.30	NA	NA	NA	NA	99.98	82.70	NA	98.80	NA	92.37	81.60
F1 Score	NA	79.20	NA	NA	NA	NA	99.95	79.00	NA	NA	NA	91.63	82.40
Technology Node	NA	NA	NA	NA	NA	NA	NA	NA	65 nm	65 nm	65 nm	180 nm	
VDD	NA	NA	NA	NA	NA	NA	NA	NA	0.4 V	1.1 V	290 mV	1.98V	
Operating Frequency	NA	NA	NA	NA	NA	NA	NA	NA	1 kHz	1 kHz	1 kHz	25 kHz	
Area (mm^2)	NA	NA	NA	NA	NA	NA	NA	NA	0.105	1.67	0.10	1.18	
Power (μW)	NA	NA	NA	NA	NA	NA	NA	NA	0.56 pJ/operation	12.9 nJ/inference	NA	11.098	

NA: Not Applicable. COS: Class Oriented Scheme (Experiment-1). SOS: Subject Oriented Scheme (Experiment-2).

also uses a set of complex features like log energy entropy, peak-to-average power ratio, and wavelet transform with a 30-s ECG sample window, making it unsuitable for wearable devices. It is important to note that most researches focus on software implementation, while only Andersson et al. [22] have implemented the AF classifier on ASIC to achieve sensitivity and specificity of 90.60% and 97.60%, respectively.

It is worth mentioning that the work proposed in [22] accepts a 128 RR-interval time series as input and includes only an AF detector. The detector in [22] depends on an external processing unit to analyze and extract the RR-interval time series from the ECG data. Implementation of the RR-interval extraction block substantially increases area and power due to its computation and memory requirements. Further, the design in [22] employs hard-coded threshold values, which are not adaptive to the ECG variations in practical scenarios.



CMOS Process	SCL 180nm
Area	1.18 mm^2 (1086.84 $\mu m \times 1086.84$ μm)
Voltage	1.98 V
Frequency	25 kHz
Dynamic Power	10.2407 μW
Static Power	0.8571746 μW

Fig. 10. Layout photograph of co-processor and its specifications.

Andersson and Rodrigues [31] use a similar approach as stated in [22]. The model proposed in [31] takes the RR-interval series as input to extract the turning point ratio (TPR), root mean square of successive difference (RMSSD),

and Shannon entropy (SE), Later, it utilizes a threshold detector to classify a beat as normal or AF. Similar to the model proposed in [22], this model does not extract the RR-interval from ECG signal, and an external processing unit is required for RR-interval extraction. Further, the AF detector [31] also uses nontunable hard-coded thresholds.

The methods proposed in [22] and [31] consume less area than ours because they implement only classifier on hardware using 65-nm technology and does not realize RR-interval extraction block. Further, their power consumptions are less than our proposed work because of using very low supply voltage (VDD) in the subthreshold (subV_T) region and low operating frequency. The operating frequency of the classifier block is considered while estimating the power in the methods proposed in [22] and [31]. Since power is proportional to VDD and the operating frequency, the power consumptions of the work reported in [22] and [31] are less. As stated above, the RR-interval series extraction is not realized in hardware in [22] and [31]. Due to this, the actual operating frequency of the proposed designs would vary for real-time ECG analysis. Our proposed design implements a complete end-to-end solution for AF detection, including feature extraction and classification of ECG signal. Therefore, it can be considered as a better choice for real-time AF detection and can be employed in any wearable and portable devices.

Additionally, Lim et al. [11] explore hardware–software co-design to achieve an accuracy of 95.30% but employ complex features including power spectral density, log energy entropy, wavelet transform with ANN as a classifier, and a 10-s input ECG sample window. Lim et al. [11] have prototyped the design on Intel's DE2-115 FPGA board, which features a Cyclone IV FPGA and an onboard Nios II processor. The ECG-processing blocks along with classifier are executed on the Nios II processor software, while only fast Fourier transform (FFT) is hardware accelerated and is implemented on FPGA. Thus, the hardware–software co-design methodology is utilized to implement ECG feature processing along with the classifier on the Nios II processor (software) and FFT on Cyclone IV FPGA (hardware). Our proposed design, including the complete ECG processing and the classifier, is realized on FPGA and then synthesized for an ASIC implementation. Sadasivuni et al. [32] have proposed an analog machine learning classifier IC to detect sepsis and AF from ECG signal. The classifier achieves an average accuracy of 98.2% for AF detection and 90.7% for predicting sepsis. Using a 30-s ECG signal, a set of 63 time domain features are computed for AF detection. It is important to note that only the ANN classifier is implemented on ASIC, while feature extraction is implemented OFF-chip. Extracting 63 features for AF detection will drastically increase the feature extraction block's computation complexity. The on-chip ANN classifier [32] utilizes an area of 1.67 mm^2 at 65 nm, while our classifier, with its simplified DWT implementation and ML model, utilizes an area of 1.18 mm^2 at 180 nm.

The proposed classifier has 8% less accuracy than the algorithm presented in [21]. The classifier proposed in [21] utilizes a feature vector of 2700 samples with a 13-layer convolution neural network, making it compute-intensive and consuming more power and hardware resources. Such complex classifiers are not suitable for wearable devices and can only be employed in a medical setup using a software platform.

Moreover, the method reported in [14] uses an approach to evaluate performance on the blind data and reports an accuracy of 82.7% and an F1 score of 79. This method is similar to the approach adopted in Experiment-2 and yields comparable performance with our implementation. However, it employs 37 time and frequency domain features, making it inefficient to be used in wearable applications. Our proposed classifier is optimized not only at the algorithmic level but also at the architectural level to be utilized in low-power wearable devices.

It is observed from Table III that our proposed classifier has better or comparable performance than other state-of-the-art methods. However, multilead ECG data and more physiological variables should be employed to obtain a medically acceptable device. As we know, the primary motive of the wearable device is to alert an individual for any anomaly and not to provide any clinical suggestions. Therefore, the area and power optimal proposed AF classifier can be utilized to realize wearable devices.

V. Conclusion

This article proposes hardware realization of an end-to-end area and power-efficient AF classifier for wearable health care devices. Since abnormal atrial activity is confined in the low-frequency range (<12 Hz), it is to mention that for the first time, this frequency band is directly analyzed for AF detection. Further, using the integer Haar wavelet and an efficient realization of the multilevel decomposition technique, the computational complexity of the proposed classifier is reduced significantly. This classifier is implemented on 180-nm CMOS technology. It utilizes an AM with an optimized DNN to classify AF consuming 11.098 μW power at 25 kHz with an accuracy of 92.37% for the class-oriented classification and 81.60% for the subject-oriented classification in real-time. This makes our design a highly suitable candidate for wearable health care devices.

References

[1] V. Fuster et al., "ACC/AHA/ESC 2006 guidelines for the management of patients with atrial fibrillation-executive summary: A report of the American College of Cardiology/American Heart Association task force on practice guidelines and the European Society of Cardiology Committee for practice guidelines (writing committee to revise the 2001 guidelines for the management of patients with atrial fibrillation)," *Eur. Heart J.*, vol. 27, no. 16, pp. 1979–2030, 2006.

[2] P. Kirchhof et al., "2016 ESC guidelines for the management of atrial fibrillation developed in collaboration with EACTS," *Kardiologia Polska, Polish Heart J.*, vol. 74, no. 12, pp. 1359–1469, 2016.

[3] S. Nattel, "New ideas about atrial fibrillation 50 years on," *Nature*, vol. 415, no. 6868, pp. 219–226, Jan. 2002.

[4] J. Oldgren et al., "Variations in cause and management of atrial fibrillation in a prospective registry of 15 400 emergency department patients in 46 countries: The RE-LY atrial fibrillation registry," *Circulation*, vol. 129, no. 15, pp. 1568–1576, Apr. 2014.

[5] P. Langley, J. P. Bourke, and A. Murray, "Frequency analysis of atrial fibrillation," in *Proc. Comput. Cardiol.*, vol. 27, 2000, pp. 65–68.

[6] R. Couceiro, P. Carvalho, J. Henriques, M. Antunes, M. Harris, and J. Habetha, "Detection of atrial fibrillation using model-based ECG analysis," in *Proc. 19th Int. Conf. Pattern Recognit.*, Dec. 2008, pp. 1–5.

[7] N. Sasaki et al., "Frequency analysis of atrial fibrillation from the specific ECG leads V7–V9: A lower DF in lead V9 is a marker of potential atrial remodeling," *J. Cardiol.*, vol. 66, no. 5, pp. 388–394, 2015.

[8] M. Zihlmann, D. Perekrestenko, and M. Tschannen, "Convolutional recurrent neural networks for electrocardiogram classification," in *Proc. Comput. Cardiol. Conf. (CinC)*, Sep. 2017, pp. 1–4.

[9] S. Ladavich and B. Ghoraani, "Rate-independent detection of atrial fibrillation by statistical modeling of atrial activity," *Biomed. Signal Process. Control*, vol. 18, pp. 274–281, Apr. 2015.

[10] D. E. Lake and J. R. Moorman, "Accurate estimation of entropy in very short physiological time series: The problem of atrial fibrillation detection in implanted ventricular devices," *Amer. J. Physiol.-Heart Circulatory Physiol.*, vol. 300, no. 1, pp. H319–H325, Jan. 2011.

[11] H. W. Lim, Y. W. Hau, M. A. Othman, and C. W. Lim, "Embedded system-on-chip design of atrial fibrillation classifier," in *Proc. Int. SoC Design Conf. (ISOCC)*, Nov. 2017, pp. 90–91.

[12] R. Alcaraz, F. Sandberg, L. Sörnmo, and J. J. Rieta, "Classification of paroxysmal and persistent atrial fibrillation in ambulatory ECG recordings," *IEEE Trans. Biomed. Eng.*, vol. 58, no. 5, pp. 1441–1449, May 2011.

[13] S. Asgari, A. Mehrnia, and M. Moussavi, "Automatic detection of atrial fibrillation using stationary wavelet transform and support vector machine," *Comput. Biol. Med.*, vol. 60, pp. 132–142, May 2015.

[14] R. Mahajan, R. Kamaleswaran, J. A. Howe, and O. Akbilgic, "Cardiac rhythm classification from a short single lead ECG recording via random forest," in *Proc. Comput. Cardiol. Conf. (CinC)*, Sep. 2017, pp. 1–4.

[15] M. Janveja and G. Trivedi, "An area and power efficient VLSI architecture for ECG feature extraction for wearable IoT healthcare applications," *Integration*, vol. 82, pp. 96–103, Jan. 2022.

[16] S. G. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," in *Fundamental Papers in Wavelet Theory*. Princeton, NJ, USA: Princeton Univ. Press, 2009, pp. 494–513.

[17] J. Y. L. Low and C. C. Jong, "Unified Mitchell-based approximation for efficient logarithmic conversion circuit," *IEEE Trans. Comput.*, vol. 64, no. 6, pp. 1783–1797, Jun. 2015.

[18] M. Ito, D. Chinnery, and K. Keutzer, "Low power multiplication algorithm for switching activity reduction through operand decomposition," in *Proc. 21st Int. Conf. Comput. Design*, 2003, pp. 21–26.

[19] G. Moody, "A new method for detecting atrial fibrillation using RR intervals," in *Proc. Comput. Cardiol.*, 1983, pp. 227–230.

[20] *Virtex-7 FPGA Family*. Accessed: Sep. 15, 2022. [Online]. Available: https://www.xilinx.com/products/silicon-devices/fpga/virtex-7.html

[21] S. Nurmaini et al., "Robust detection of atrial fibrillation from short-term electrocardiogram using convolutional neural networks," *Future Gener. Comput. Syst.*, vol. 113, pp. 304–317, Dec. 2020.

[22] O. Andersson, K. H. Chon, L. Sörnmo, and J. N. Rodrigues, "A 290 mV sub-V_T ASIC for real-time atrial fibrillation detection," *IEEE Trans. Biomed. Circuits Syst.*, vol. 9, no. 3, pp. 377–386, Jun. 2015.

[23] G. B. Moody and R. G. Mark, "The impact of the MIT-BIH arrhythmia database," *IEEE Eng. Med. Biol. Mag.*, vol. 20, no. 3, pp. 45–50, May/Jun. 2001.

[24] M. Janveja, R. Parmar, M. Tantuway, and G. Trivedi, "A DNN-based low power ECG co-processor architecture to classify cardiac arrhythmia for wearable devices," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 69, no. 4, pp. 2281–2285, Apr. 2022.

[25] H. Saadat, H. Bokhari, and S. Parameswaran, "Minimally biased multipliers for approximate integer and floating-point multiplication," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 37, no. 11, pp. 2623–2635, Nov. 2018.

[26] J. Lee, B. A. Reyes, D. D. McManus, O. Mathias, and K. H. Chon, "Atrial fibrillation detection using an iPhone 4S," *IEEE Trans. Biomed. Eng.*, vol. 60, no. 1, pp. 203–206, Jan. 2013.

[27] K. K. Parhi and T. Nishitani, "VLSI architectures for discrete wavelet transforms," in *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 1, no. 2, pp. 191–202, Jun. 1993, doi: 10.1109/92.238416.

[28] M. Vishwanath, R. M. Owens, and M. J. Irwin, "VLSI architectures for the discrete wavelet transform," in *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 42, no. 5, pp. 305–316, May 1995, doi: 10.1109/82.386170.

[29] J. Loh, J. Wen, and T. Gemmeke, "Low-cost DNN hardware accelerator for wearable, high-quality cardiac arrythmia detection," in *Proc. IEEE 31st Int. Conf. Appl.-Specific Syst., Architectures Processors (ASAP)*, Jul. 2020, pp. 213–216, doi: 10.1109/ASAP49362.2020.00042.

[30] M. Jobst et al., "ZEN: A flexible energy-efficient hardware classifier exploiting temporal sparsity in ECG data," in *Proc. IEEE 4th Int. Conf. Artif. Intell. Circuits Syst. (AICAS)*, Jun. 2022, pp. 214–217, doi: 10.1109/AICAS54282.2022.9869958.

[31] O. Andersson and J. N. Rodrigues, "A 400 mV atrial fibrillation detector with 0.56 pJ/operation in 65 nm CMOS," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2015, pp. 2628–2631, doi: 10.1109/ISCAS.2015.7169225.

[32] S. Sadasivuni, S. P. Bhanushali, S. S. Singamsetti, I. Banerjee, and A. Sanyal, "Multi-task learning mixed-signal classifier for in-situ detection of atrial fibrillation and sepsis," in *Proc. IEEE Biomed. Circuits Syst. Conf. (BioCAS)*, Oct. 2021, pp. 1–4, doi: 10.1109/BioCAS49922.2021.9644994.

[33] W. J. Tompkins, *Biomedical Digital Signal Processing*. Upper Saddle River, NJ, USA: Prentice-Hall, 1993.

Rushik Parmar received the B.E. degree in electronics and telecommunication from the Institute of Engineering and Technology, DAVV, Indore, India, in 2019.

He worked as a Software Engineer with Accenture Digital, Mumbai, Maharashtra, India, from 2019 to 2020. He is currently a Research Scholar with IIT Guwahati, Guwahati, Assam, India. His research areas include VLSI system design, machine learning, and VLSI architectures for biomedical applications.

Meenali Janveja (Student Member, IEEE) received the B.Tech. degree in electronics and communication engineering from the Government Women Engineering College, Ajmer, India, in 2013, and the M.Tech. degree in VLSI design from Indira Gandhi Delhi Technical University for Women, India, in 2016.

She worked as an Assistant Professor with the Department of Electronics and Communication Engineering, G.L. Bajaj Institute of Technology and Management, India, from 2016 to 2017. She is currently a Research Scholar with the Department of Electronics and Electrical Engineering, IIT Guwahati, Guwahati, India. Her research areas include digital VLSI design, computer architecture, machine learning, and VLSI for biomedical signal processing.

Jan Pidanic (Senior Member, IEEE) was born in 1979. He received the M.Sc. and Ph.D. degrees from the University of Pardubice, Pardubice, Czechia, in 2005 and 2012, respectively.

His research interests include signal processing in passive radar systems, bistatic radars, clutter modeling, and optimization of signal processing algorithms with parallel processing techniques.

Gaurav Trivedi (Member, IEEE) received the Ph.D. degree in electrical engineering from IIT Bombay, Mumbai, Maharashtra, India, in 2007.

He is currently an Associate Professor with the Department of Electronics and Electrical Engineering, IIT Guwahati, Guwahati, India. He worked as a Senior Member of Technical Staff with Cadence Design System India Pvt. Ltd., Noida, UP, India and Mentor-Siemens (Earlier Berkeley Design Automation India), Bengaluru, India, for three years, and as a Post-Doctoral Fellow at IIT Bombay, for two years. His research interests include VLSI CAD, semiconductor devices, digital and analog circuit design, high-performance computing, computer architecture and algorithms, embedded and IoT, and quantum computing.