Biochip Architecture for Cardiac Pathologies Detection

Pedro Brito de Sá

Abstract—Off-the-person electrocardiography is gaining momentum as superior quality single-lead ECG signals can now be acquired from a person’s hands or arms (Lead 1). Although several heart disease algorithms have been described in recent years, all of them make use of more than one ECG lead and, by being tested only against standard databases, are sensitive to the acquisition conditions. The purpose of this work was to design a robust Lead 1-only real-time heart disease detection system and implement it on an FPGA-based development board. The proposed system is based on a signal processing pipeline composed of: ECG signal denoising; heartbeat detection and segmentation; extraction of dynamic morphological features; and heartbeat classification (normal or abnormal and corresponding condition). Resorting to the only database from MIT's Physiobank with Lead 1 annotated recordings, InCarTDb, software implementation of this pipeline resulted in a 4-class model with a classification accuracy up to 96.5%. A biochip architecture for cardiac pathologies identification was then designed and simulated in VHDL, and mapped to a virtual FPGA evaluation board. The resulting architecture verified accuracies from 84 to 100% for different patients of the database, less than 30% of the evaluation board resources, a working clock frequency of 35 MHz and a total power consumption of 192 mW.

Keywords—Single-Lead Electrocardiogram, Heart Disease Identification, Signal Denoising, Feature Extraction, Signal Processing Architecture, FPGA

I. INTRODUCTION

Cardiovascular diseases account for the death of more than 17.3 million people per year around the world. By 2030, this number is expected to have increased to 23.6 million [1]. Since its development, electrocardiography has been used as a diagnostic tool for multiple heart conditions, being most effective in the detection of arrhythmias and myocardial infarctions [2].

Several electrocardiography devices are currently available on the market, ranging from high-end hospital equipment to portable domestic appliances. The acquisition methods vary with the device, although the most common practice is the 12-lead ECG, which relies on an array of ten adhesive sensors on the subject’s torso and limbs [3]. Although there is redundancy in this setup, it prevents inconsistencies and provides a full spatial overview of the heart’s electrical behavior. However, it makes prolonged monitoring impractical for a lay person such as a bus driver. Nonetheless, it is possible to measure one view of the ECG by taking the electrical potential with a contact based sensor on each of the subject’s hands [4], [5]. This single-lead view, although not as complete as the 12-lead view, still provides relevant information on the subject’s biometrics and pathologies detectable through the ECG’s Lead I [4].

Aside from the non-intrusiveness and easiness of use, contact based single-lead acquisitions also imply a smaller amount of data [4]; therefore, the required computational and processing power of the system decreases as well. Such a setting allows for the current embedded computational platforms to be regarded as a possibility for ECG processing in real-time environments. Although there are authors already implementing ECG processing algorithms, for example, on FPGA [6], [7], [8], [9], [10], [11], [12], there is still work to be done towards a reliable automatic real-time pathology detection embedded platform.

A. Contributions

This dissertation follows the tendency of off-the-person electrocardiography and is the first end-to-end architecture specially designed for Lead-1 real-time processing, with computational and energetic efficiencies which enable the usage in wearable applications. In this setting, it lays the ground work for real-time Lead 1-only ECG heart disease detection, in particular with the InCarTDb. A whole system was implemented taking the 75 Lead 1 annotated recordings of this database as a starting point, although it can be applied to any other set of signals. The system developed in software considered the following steps:

- The signal denoising and artifact removal methods were assessed and the pre-processing considered baseline removal by subtracting a moving average window and low-pass filtering with an IIR filter;
- The filtered signal then has its amplitude normalized by dividing the samples of each consecutive window by the a normalization parameter calculated from the maximum of the previous window;
- From the annotated peaks location, a window of 250 ms before each peak and 375 ms afterwards is taken;
- The dynamic features of each peak are derived from the RR interval, and the segmented heartbeat is downsampled if the heart rate is below 60 bpm and upsampled if it is above 133 bpm;
- The morphological features of the heartbeat are obtained from the analysis coefficients of the DWT and PCA is applied only to these features for dimensionality reduction, being then concatenated with the dynamic features;
- The recordings are randomly separated between a training and a test set and the features are clustered, having the training cluster being fed into a PNN classifier, and the testing cluster verifying its performance.
Classification of 4, 5 and 6 different classes were attempted, but due to the lack of occurrences of the 5th and 6th classes and the poor results they provided, the 4-class classifier was set as final. Verifying an overall accuracy of over 96.5% in a mostly inter-patient paradigm, it is on par with multiple-lead 4-class classifier described by Jannah et al [13].

These stages were then projected onto a multiple-cycle hardware architecture, considering them independent but interconnected blocks. All these blocks have their own Control Unit state-machine, and most of them have also a Datapath sub-unit to aid in the arithmetic operations. However, instead of having the annotated peak location, two heartbeat detection stages were implemented. One considered a fixed amplitude threshold for the normalized signal, above which a peak occurrence would be considered. The other took a cross-correlation of a fixed size window of the normalized signal with a template heartbeat. In the particular case of the InCarTDb, this was necessary to identify a pathology class that had no positive peak. The architecture is also scalable and programmable, which allows it to be tailored to: different acquisition setups; different parameters in most of the filtering, segmentation, feature extraction and classification stages; different abnormalities to be classified; different leads and even more than one lead simultaneously.

The final design of the architecture was then mapped to the virtual resources of a FPGA development board verified very satisfactory results. The signal classification verified accuracies of the correctly segmented heartbeats between 84 to 100% in some patients. The resources utilized does not reach 30% of the available resources of the board, the clock frequency can go up to 35 MHz and the power consumption can be as low as 122 mW.

Although being a prototype and having its limitations, this architecture establishes important ground work for the future development or integration of heart disease monitoring solutions into low-power and computationally simple environments.

B. Outline

An overview of the current developments in ECG processing systems is presented in Section II. Section III provides more detailed view of the methodologies tested in software, having the chosen hardware implementation being described in IV. The discussion of the experimental results for a practical case is elaborated in Section V, while

II. RELATED WORK

For multiple-classification systems, the works of Martis et al. [14], Wang et al. [15] and Ye et al. [16] verify some of the best results possible. Their works start with the application of pre-processing stages like baseline or bandpass filtering. Afterwards, all of these authors take a DWT-based denoising approach. A signal reconstructed from some detail coefficients has the QRS complexes segmented and each sample is considered to be a feature. For the purpose of dimensionality reduction all three approaches make use of PCA. Then Wang et al. [15] uses Fisher’s Linear Discriminant Analysis (LDA), while Ye et al. [16] resorts to Independent Component Analysis (ICA), having Martis et al. [14] combining both of the methods. These three approaches are different methods to project the data into other dimensions of higher variance and correlation, having a reduced set of features that corresponds to a good approximation of the data. Ye et al. [16] then use a Support Vector Machine (SVM) to classify each heartbeat into one of 16 different classes. Both Martis et al. [14] and Wang et al. [15] try different settings of Neural Networks (NN) and SVM’s. Their results unanimously state that the use of a PNN classifier after dimensionality reduction with PCA+LDA (+ICA in Martis et al. [14]) provides the best result possible, all around 99.5% of accuracy.

The current state-of-the-art hardware systems works verify a set of different applications, having simple arrhythmia detectors based on the heart rate or other straightforward features [8], [12], [17], more complex algorithms to detect P and T waves in real-time [7] or ST segment elevation/depression [10], and, at the top, neural networks [9], [11] and fuzzy clustering [18] based algorithms, which make up for a bigger challenge.

However, most authors either do not fully implement a classification system or, when they do, the architectures they describe either have incomplete information or used microprocessors to aid the processing. Jewajinda et al [9] and Chatterjee et al [7] developed dedicated VHDL architectures, however no data on the hardware resources, timing or power was provided. Both Mimouni et al [10] and Junior et al [18] implemented the architecture in FPGA’s MicroBlaze processing unit, also with no indications of the performance aside from the signal classification quality.

The work of Sun et al [11] offers a comparison between the use of 200 Hz smartphone processor programmed through software and a 500 MHz ASIC. The ASIC, which uses 45 nm CMOS technology, was first prototyped in a Xilinx Virtex-5 FPGA with 65 nm scale. This yields faster execution time and lower power consumption for the ASIC, but due to the high costs and quantities required to produce a batch of ASIC’s, the circuit should be designed, verified and then tested on an FPGA to avoid wasting money and production resources. On another hand, an FPGA based feature extraction system for prenatal ECGs was developed by Abburi et al [19] verified a maximum clock frequency of 22.4 MHz and a total power consumption of 68.25 mW.

III. ECG HEART DISEASE DETECTION ALGORITHM

The ECG processing workflow is composed by four main stages: pre-processing, heartbeat segmentation, feature extraction and classification. Figure 1 contains block diagram of the dataflow of the whole process.

The current software system implements a three-part pre-processing stage. First, the baseline is removed by subtracting the result of a moving average filter to the signal. A high-pass IIR filtering is then applied to remove the high-frequency noise. The pre-processing ends after the amplitude of the signal is normalized to have its peaks around 1 mV.

Since the databases used for model development must have annotations of the heartbeats, the heartbeat identification (the
Feature extraction begins with the dynamic features of the signal being calculated from the time intervals between the R peaks of each heartbeat. Once extracted, the heartbeat segment goes into the resampling block where, if needed, it is stretched or compressed in order to have the full length of the P-QRS-T waves one only one heartbeat inside the segment window. This step is followed by the extraction of the morphological features of the segment through the analysis coefficients of the DWT. Dimensionality reduction ensues, and with a reduced set of morphological features plus the dynamic features, the system is ready for classification.

The final set of heartbeat features is divided into a training set and a test set. The former is used to generate the parameters of a PNN classifier, which is then applied to the test set. This process renders a set of labels for each of the test set heartbeats, which are the final product of the ECG heart disease detection system.

A. Pre-Processing

There are several filtering and noise removal methods that allow for a better feature extraction and, consequently, beat classification.

Baseline drift is a dynamic low-frequency offset which can be removed with a high-pass filter. However, this baseline drift can be approximated by the mean of the signal for well-calibrated window. Since our goal is to have the signal oscillate around zero, a moving-average filter was applied to the signal and the result subtracted to it. This method is an extremely efficient high-pass filter which bears almost no computational load. Different window sizes were tested: 125 ms, 250 ms, 500 ms, 1000 ms and 2000 ms. For each sample of the signal, the average of the amplitudes of all the samples inside the window centered on it would be subtracted. Testing revealed 500 ms as the most suitable window size.

After the baseline filtering, the most concerning noise is of high frequency. Thus, a low-pass filter was dimensioned in as to remove this noise. Given that the purpose of filtering for this work is to have well-defined P, Q, R, S and T waves with little to no noise, different filters were designed. Since FIR filters have a low stop-band attenuation for lower orders and is computationally more complex than an equivalent IIR filter, the aim was set to design an IIR filter with the lowest order possible, since the phase shifts do not affect the signal greatly. Different quantized IIR filters were tested, with a Chebyshev Type II filter with 45 Hz of cutoff frequency and 80 dB of stopband attenuation of orders 8 providing the best results.

ECGs have tremendous amplitude variability between patients, sometimes even for the same patient, and also between acquisition setups. Given that the aim of this work is to provide a one-fits-all processing system and morphological features of the signal are to be analyzed, signals have to be normalized in order to guarantee a more reliable classification. The current work’s real-time aims at standardizing the signal in order to to have its peaks around $1 \text{ mV}$. For this, the following steps were adopted: take a window of the signal with no overlap; detect the window’s maximum and consider it the standardization parameter; take the closest power of two of the normalization parameter and divide the values of the next window by it; check if the maximum of the window is within a desirable band and, if not, updates it. The presented algorithm comprises an initialization stage, which considers that the first window of the signal is not normalized, only being used to generate the first normalization parameter. The normalization parameter update law takes an average between the previous value and the new one as to perform a smoother transition between the amplitudes.

B. Feature Extraction

Once the signal is filtered and normalized, the next step is to extract the features to be used in classification. For software analysis, the annotations of the database were used to get
the segments of signal 250 ms before and 375 ms after each peak, as is standard procedure [20]. However, the hardware implementation will contain a peak thresholding and a template matching strategies for beat detection, as described in Section IV.

Dynamic features were considered in this work, namely the ones used by Ye et al [16]. These features are derived from each annotated beat by considering

\[ RR_{prev}(n) = \frac{\text{peakPosition}_n - \text{peakPosition}_{n-1}}{F_s}, \]

where \( F_s \) is the sampling frequency, \( ann \) is an array with the location of the peak of each beat in samples, and by storing these values in an array. Each of the entries of this array is the first dynamic feature for each beat, and the other two are computed from them as in

\[ RR_{local}(n) = \frac{\sum_{i=j}^{n} RR_{prev}(i)}{n-j}, \] (2)

\[ RR_{avg}(n) = \frac{\sum_{i=k}^{n} RR_{prev}(i)}{n-k}, \] (3)

where \( j \) and \( k \) are the positions of the first RR interval that is at a time stamp of less than 10 seconds and 5 minutes, respectively.

Similarly to the amplitude variability, ECG signals also present considerable dynamic variations. Since heart rates can range from 30 to 180 bpm or more, the P-QRS-T waves can be seen in windows that range from just 333 ms to 2000 ms. However, it is common among authors [3][20][16] to use a window of 600 to 850 ms (always with 33.3% of the samples before the peak, and the remaining 66.7% after) to consider the segment. This means that for higher heart rates there may appear waves from the adjacent beats inside the window, or for lower heart rates that some of the waves are not fully captured. Therefore, and considering that classification requires some time stability in terms of positioning of the features, resampling of some segments becomes necessary. In software, this process was implemented through the Matlab inbuilt resampling functions, and the best results were given after adding symmetric mirroring of the first and last eight coefficients of the input of each filter. This is necessary due to a property of FIR filters, which do not provide a reliable output until the input buffer is filled with a number of samples equal to half of the order of the filter. For the first or second stage of the DWT this is not a serious issue, but for the third and fourth stages the amount of features is significantly lower the features become almost irrelevant at that point.

These features are then put through PCA. An analysis of the most suitable number of principal components that can be used for the classifier is performed in Section V.

C. Classification

From the results of the work of Martis et al [14] and Wang et al [15], and with the knowledge that neural networks can usually be implemented with standard arithmetic blocks, thus being fit for hardware implementation, it was decided that a PNN would be used as a classifier for the system. PNNs are usually composed by three layers: Input Layer, Radial Basis Layer and Competitive Layer [21]. A functional representation of the general structure is shown in Figure 2.

The Input Layer consists of the signal or feature set to be classified. For the current method, we will consider it as a one-dimensional array of \( R \) features. The Radial Basis Layer implements

\[ A_i = \text{radbas}(||IW_i - F||b_i), \] (4)

where \( A_i \) is the \( i \)-th element of the Radial Basis Layer \( A \), \( IW_i \) is the \( i \)-th row of the matrix input weights, and \( b_i \) the \( i \)-th element of the biases vector. Each row of the \( IW \) matrix is a feature vector of the training set, which excludes the necessity of a backpropagation algorithm for obtaining a network model, unlike other neural networks. The radial basis function can be any gaussian function of the format

\[ \text{radbas}(x) = ae^{-\frac{(x-b)^2}{2c^2}}, \] (5)

where \( a, b \) and \( c \) are arbitrary constants. The PNN classifier used in the tests was the inbuilt Matlab PNN, which uses the
This layer takes the input features, computes the distance to each of the training feature vectors, multiplies it with the bias and computes an absolute similarity metric, in this case with a squared negative exponential function. This metric is then fed to the Competitive Layer, where the similarity values corresponding to each class are summed according to the values of the $LW$ matrix. For each of its columns, only one row is 1, whereas the others are 0, and that is the corresponding class of that neuron. Once all values have been summed, the class with the total maximum probability is outputted in a vector format [21].

IV. HARDWARE IMPLEMENTATION

The sampling frequency of an ECG is rarely above the 500 Hz [20]. Since both ASIC and FPGA technologies usually operate in the MHz and GHz range, a multi-cycle architecture was preferred to a pipelined one as to reduce the resource utilization. This architecture is structured in the following way: a main circuit contains the Control Unit, the sub-blocks of the Datapath Unit and the memory blocks to input data and store the circuit output. BRAMs were used for the signal input and output since the interface with signal acquisition and post-processing structures was not established. In a physical context, the architecture requires storage units between the acquisition setup and itself, as will be described ahead in this section. The dataflow of the architecture is illustrated in Figure 3. For this figure, every single-lined box represents a processing block that encompasses control and support logic, storage units and Datapath sub-units, which will be described in the coming sections. The double-lined boxes represent the storage BRAMs and the cell-vectors represent shared storage arrays that allow the transition from one block to the next. Throughout the remaining figures of this chapter, the double-line boxes in schematics always represent storage units, and for the cases where the type of unit is not specified it should be considered a register. The overall processing flow goes as follows:

1) The samples are processed in sets of variable size. Each new set of samples is streamed from the input memory block and stored in a FIFO array of 500 ms that may have previously read samples. This array is henceforth called a window.

2) The samples of this window are put sequentially through the Moving Average baseline filter to be high-pass filtered (see also Figure 1) and stored in a new FIFO window of the same size.

3) When this step is done, the samples are low-pass filtered sequentially through the IIR filter and stored in the same array during the intermediate stages.

4) During the final stage of the IIR filtering, each new sample is directly put through the amplitude normalization process and stored in a third FIFO window with a fixed size for processing. Every normalized sample is also immediately written to memory for usage outside the architecture, if needed.

5) For each new normalized sample, this last window is put through the template matching process, where the cross-correlation of the window with a heart disease template loaded from memory is calculated.

6) If the cross-correlation is above a certain threshold, it is considered a match and the processing moves to the feature extraction.

7) If not, the sample is analyzed by the peak detection block. Here, when a peak is detected, the block will ensure that more new samples are stored after the peak in the normalized samples window. In all cases, this requires that the set of samples finishes pre-processing and that new samples are read from memory and pre-processed. The number of new samples required is calculated as to place the peak sample in a specific position of the window, more precisely at the point corresponding to 250 ms. When no peak is detected, a default number of new samples is read from memory and pre-processed.

8) The feature extraction starts either when a peak is in the right place of the normalized window, or there was a template match. In both cases the processing is the same and it disables the reading of new samples from the input memory. The dynamic features are calculated and stored in an array which will contain the final features for the classifier. The features are also outputted to a BRAM, since they contain relevant information on the heart rate.

9) Depending on the value of one of the features (the instantaneous RR interval), the normalized window may be up-sampled, down-sampled or left as-is.

10) After the resampling block, the DWT analysis coefficients are extracted as morphological features and stored in an array.

11) This array is fed into the Principal Component Analysis for dimensionality reduction and the outputted features are joined with with the dynamic features in the final features array.

12) The PNN classifier is fed the final features and outputs the classification of the processed heartbeat, which is then written to an output BRAM. When the classification is finished, the processing of new samples is resumed, either from reading the signal from the initial BRAM or from the point where it stopped after the template match.

A. Control Unit and Datapath

The Control Unit commands all the data transfers and processes. It works as a sequential state-machine, and it is composed of independent blocks of smaller state-machines. These parts interact with each other by using flag signals. These enable and disable its functioning and access to the shared storage units, both inside and outside of the Control Unit. With the exception of the Enable Read block, all single-
lined boxes of Figure 3 are controlled by an independent state-machine inside the Control Unit.

For each processing stage, the Control Unit interacts with the storage and the Datapath units of the circuit, here represented by Figure 4. The input of raw signal and the architecture’s outputs are received and sent by the Control Unit.

For all stages of the processing except the Amplitude Normalization and the Resampling, there is a corresponding Datapath sub-unit. These sub-units contain mostly arithmetic functions which are dedicated to process both the inputs from the Control Unit and from the necessary memory blocks. In particular, the PNN is decomposed in several blocks, all similar to each other, for parallel processing of different inputs. In this architecture, the number of PNN blocks is equal to the number of classes of the model.

### B. Time Constraints

The processing of the signal is not done through a continuous stream but through batches of data. For different systems, there can be both power and performance limitations, and the frequency of the main clock greatly impacts both parameters. Therefore, a general guideline for the choice of the clock frequency is presented.

The clock frequency takes several parameters into consideration: the execution time of the baseline ($t_{MA}$) and low-pass ($t_{IIR}$) filtering, template matching ($t_{tpMat}$) and peak detection ($t_{pDet}$) for each sample, the processing and classification of a detected peak/match, the sampling frequency and the heart rate. When no peak/match is detected and the samples are being processed, every sample should be processed in $1/Fs$ seconds. This translates as
However, when a detected peak/match is being processed, no new sample is being read from memory, thus having a lag. This lag depends on the parameters of each of feature extraction and classification block. This creates a dependency on the heart rate. If no sample is read during the peak/match processing, then after the processing finishes the system must ensure that all stored samples are read before the next peak. In case this condition is not verified, the lag of the processing will consistently increase. Therefore, between two RR peaks at a certain heart rate the condition

$$t_{pProc} + t_{RR}F_s(t_{MA} + t_{IIR} + t_{tpMat} + t_{pDet}) < t_{RR},$$

with

$$t_{RR} = \frac{60}{\text{heartRate}},$$

should be verified. However, except for $t_{pRR}$, all times in Equation IV-B depend on the system clock frequency. Therefore, the equation can be written in function of clock frequency as:

$$S_{pProc} + t_{RR}F_s(S_{MA} + S_{IIR} + S_{tpMat} + S_{pDet}) < t_{RR},$$

where all $S$ parameters are the number of samples of execution of each block and $F_c$ is the clock frequency. Therefore, the clock frequency should verify

$$F_c > \frac{1}{t_{RR}}S_{pProc} + F_s(S_{MA} + S_{IIR} + S_{tpMat} + S_{pDet}).$$

On another hand, the functioning of the architecture when implemented is not instantaneous, having both logic and path delays. This limits the speed at which the operations are performed, thus setting a top boundary to the clock frequency. These delays are dependent on the implementation specificities, therefore an analysis of the critical path should also be performed in order to choose the most suitable clock frequency. The clock frequency then verifies a top boundary as in

$$F_c < \frac{1}{t_{critical Path}}.$$  

V. EXPERIMENTAL RESULTS

Having the software and hardware methodologies been described in the previous chapter, a particular case of the their implementation is here described. Firstly, an overview of the standard databases and classifier evaluation parameters are described in Subsection V-A. That section contains also an analysis of the database that was used throughout this work. From the standard databases, this was the only one that had annotated Lead 1 recordings, hence being the only option to validate that the architecture is fit for Lead 1-only processing, as initially proposed. However, both the software and hardware methodologies are lead-agnostic, as nothing binds them to be used only with Lead 1, any other lead could be used. The software results are described in Subsection V-B, with the most important test results at each stage and their repercussions on the final classification. Subsection V-C contains an analysis of the performance of the data I/O format, the specificities of its structure, its time constraints, power consumption, hardware utilization and signal classification quality for the chosen database.

### A. Databases and Evaluation Standards

In order to provide a standardized data workspace and methodology for a fair comparison of ECG processing methods, the Association for the Advancement in Medical Instrumentation (AAMI) developed the ANSI/AAMI EC57:1998(R)2008 protocol [20]. This protocol provides a list of recommended databases, as well as classes of heartbeats to be detected, which are explicit in Table I.

<table>
<thead>
<tr>
<th>Group</th>
<th>Symbol</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>NP</td>
<td>N</td>
<td>Normal Beat</td>
</tr>
<tr>
<td>PA</td>
<td>P</td>
<td>Fusion beat</td>
</tr>
<tr>
<td>Q</td>
<td>Q</td>
<td>Fusion of paced and normal beat</td>
</tr>
<tr>
<td>SQ</td>
<td>S</td>
<td>Supraventricular premature beat</td>
</tr>
<tr>
<td>VEB</td>
<td>V</td>
<td>Ventricular escape beat</td>
</tr>
<tr>
<td>B</td>
<td>B</td>
<td>Bundle beat</td>
</tr>
<tr>
<td>I</td>
<td>I</td>
<td>Implantable</td>
</tr>
</tbody>
</table>

In order to evaluate the quality of segmentation methods, the following measurements should be taken: Sensitivity, Positive Predictivity Rate (PPR) and False Predictivity Rate (FPR). These metrics are calculated for each class as

$$\text{Sensitivity} = \frac{TP_{seg}}{\sum_{class} TP_{class}},$$

$$\text{PPR} = \frac{TP_{seg}}{TP_{seg} + FP_{seg}},$$

$$\text{FPR} = \frac{FP_{seg}}{TP_{seg} + FP_{seg}},$$

where $TP$, $FP$ and $FN$ represent, respectively, the number of heartbeats correctly segmented, the number of segmentations that do not correspond to annotated heartbeats, and the number of annotated heartbeats that were not detected.

Likewise, to evaluate the quality of classification methods, the previous three metrics are calculated for each class as

$$\text{Sensitivity} = \frac{TP_{class}}{\sum_{class} TP_{class}},$$

$$\text{PPR} = \frac{TP_{class}}{TP_{class} + FP_{class}}.$$
was verified, where only a drop of 2% occurred without the amplitude normalization, a drop of almost 20% in accuracy being this the order of the final filter. By not applying the order of orders superior to the 8th was less than 0.5%, hence the choice of the IIR filter cutoff frequency confirmed the choice of having a 500 ms window plus an IIR filter with cutoff at 45 Hz. The order of the IIR was then tested and the improvements for the 4, 5, 6, 8 and 16 classes from InCarTDb were selected, since the remaining classes did not have enough occurrences. Average templates of the heartbeats of these classes were generated from the 75 recordings and are shown in Figure 5.

B. Evaluation of the heart disease identification procedure

The effect of several parameters of the different stages of the system on the accuracy of the final classification system were tested. All these tests considered a 100 times cross-validation inter-patient paradigm for 4-class classifier (Healthy, APC, VPC and RBBB) for which the average accuracy was 94.25%. Models with 5 classes (previous 4 classes + VFN) and 6 classes (previous 5 classes + NEB) were tested, but their performance did not prove to be satisfactory. Confusion matrices of some of the best models generated for 4, 5 and 6 classes are presented in Figure 6.

However, for the present work neither of the AAMI standard databases can be used, as none contains recordings of Lead I. Thus, another database had to be used. From PhysioNet, the largest open source project on data collection of complex physiological signals through standard methods, InCarTDb was selected as it was the only one with Lead 1 recordings. Therefore, in the context of this work all of the seventy-five 30 minutes recordings at 257 Hz from 32 different patients were used. For the classification models in this work, only six classes from InCarTDb were selected, since the remaining classes did not have enough occurrences. Average templates of the heartbeats of these classes were generated from the 75 recordings and are shown in Figure 5.

The final analysis of the classification tested the accuracy of the method for each patient. Figure 7 contains a trend line of the mean accuracy of each patient.

C. Hardware Implementation

Throughout the development of the architecture in this work, the data was inputted to the processing stages in Q4.12 or Q1.15 fixed-point format. It was then stored in hexadecimal format in .coe files and used to program distributed memory BRAMs before synthesis. At the end of the processing stages, the filtered signal, the peaks locations and respective classifications are outputted to three separate .txt files to be analyzed. Due to high simulation times, the input signal files contained 130,000 samples of data at each time, which correspond to files of around 8 minutes and 45 seconds. The architecture was synthesized and implemented in Vivado, and the hardware mapped to a Xilinx Zynq-7 ZC702 Evaluation Board.

The different tests for the resource utilization only took into consideration the assessment of the scalability of the number of classes. The post-implementation resources utilized in the architecture for 4, 5, 6, 8 and 16 classes were tested. The resource evolution with the class is represented in Table III.
Fig. 6. Confusion Matrix for some of the best models generated for an inter-patient classification process of 4, 5 and 6 classes.

Fig. 7. Accuracy average of the classifier for each patient every time it is included in the testing set.

TABLE III. FPGA RESOURCES UTILIZATION TABLE FOR THE ARCHITECTURE WITH CLASSIFIERS OF 4, 5, 6, 8 AND 16 CLASSES.

<table>
<thead>
<tr>
<th>Resource</th>
<th>4 Classes</th>
<th>5 Classes</th>
<th>6 Classes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LUT</strong></td>
<td>53200</td>
<td>53200</td>
<td>53200</td>
</tr>
<tr>
<td><strong>LUTRAM</strong></td>
<td>17400</td>
<td>17400</td>
<td>17400</td>
</tr>
<tr>
<td><strong>FF</strong></td>
<td>104400</td>
<td>104400</td>
<td>104400</td>
</tr>
<tr>
<td><strong>BRAM</strong></td>
<td>140</td>
<td>140</td>
<td>140</td>
</tr>
<tr>
<td><strong>DSP</strong></td>
<td>220</td>
<td>220</td>
<td>220</td>
</tr>
</tbody>
</table>

TABLE IV. FPGA STATIC AND DYNAMIC POWER CONSUMPTION TABLE FOR THE ARCHITECTURE WITH CLASSIFIERS OF 4, 5, 6, 8 AND 16 CLASSES.

<table>
<thead>
<tr>
<th>Target Class</th>
<th>Clock Frequency (MHz)</th>
<th>Static Power (mW)</th>
<th>Dynamic Power (mW)</th>
<th>Total On-Chip Power (mW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.07143</td>
<td>122</td>
<td>122</td>
<td>122</td>
</tr>
<tr>
<td>2</td>
<td>1.0</td>
<td>122</td>
<td>122</td>
<td>122</td>
</tr>
<tr>
<td>3</td>
<td>2.1</td>
<td>122</td>
<td>122</td>
<td>122</td>
</tr>
<tr>
<td>4</td>
<td>3.1</td>
<td>122</td>
<td>122</td>
<td>122</td>
</tr>
</tbody>
</table>

TABLE V. FPGA STATIC AND DYNAMIC POWER CONSUMPTION TABLE FOR THE ARCHITECTURE WITH DIFFERENT CLOCK FREQUENCIES.

<table>
<thead>
<tr>
<th>Target Class</th>
<th>Clock Frequency (MHz)</th>
<th>Static Power (mW)</th>
<th>Dynamic Power (mW)</th>
<th>Total On-Chip Power (mW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.07143</td>
<td>122</td>
<td>122</td>
<td>122</td>
</tr>
<tr>
<td>2</td>
<td>1.0</td>
<td>122</td>
<td>122</td>
<td>122</td>
</tr>
<tr>
<td>3</td>
<td>2.1</td>
<td>122</td>
<td>122</td>
<td>122</td>
</tr>
<tr>
<td>4</td>
<td>3.1</td>
<td>122</td>
<td>122</td>
<td>122</td>
</tr>
</tbody>
</table>

can be chosen to suit performance and power needs accordingly, as long as the condition

\[ F_c \in [0.071343; 35.714] \text{MHz} \quad \text{(19)} \]

is verified.

This maximum clock frequency is close to the values that other authors obtained [18], [19], between 22.4 and 50 MHz, and is 500 times higher than the minimum clock frequency.

As for the power consumption of the architecture, Vivado’s post-implementation power reports were used as to describe the evolution of power both for different classes and different clock frequencies. For the class analysis, Table IV contains the assessment of the power consumption for the 4, 5, 6, 8 and 16 classes. The static power remains the same, and a fairly linear increase of 2% per class of the dynamic power is verified.

For the dependency of power to the clock frequency, different values were tested: 71.343 kHz, and 1, 10, 20 and 33.3 MHz (Table V). As for the class tests, the static power remained the same as for all values of clock. However, the dynamic frequency went from less than 1 mW, at the minimum clock frequency, to 70 mW close to maximum frequency.

The results of segmentation and classification for some selected recordings are shown in Table VI and VII, respectively.

VI. CONCLUSION

This work aimed for the development of a customizable, scalable, low-power and low area architecture for real-time ECG processing. An emphasis was made on Lead 1 systems,
because Lead 1-only off-the-person electrocardiography is taking its first steps [4], [5], and a specific hardware architecture for disease detection in these conditions has not been described in literature yet.

The ECG heart disease detection system that was proposed was developed and a 4-class classifier with accuracies up to 96.5% in a mostly inter-patient paradigm was obtained. Independent blocks for each stage of the processing were developed and interconnected in a multiple-cycle architecture. Having their own Control Unit state-machine, most of them also a Datapath sub-unit to aid in the arithmetic operations. The architecture is customizable through generic parameters defined prior to synthesis. Being mapped to a low-end FPGA evaluation board, the minimum power consumption of the architecture is of 122 mW, which is sufficiently good for integration in portable devices. The clock frequency can be any value between 71.343 kHz and 35.714 MHz, which is within the standard working frequencies of different authors [18].

In terms of resource utilization, this system did not require more than 20% of any available resources of the evaluation board except for the LUTs, where 38% of them were required. For the signal classification quality, accuracies from 88.3% to 100% were verified for different patients.

REFERENCES


