8-1-2007

Investigating the effects of an on-chip pre-classifier on wireless ECG monitoring

Alexandru Samachisa

Investigating the Effects of an On-Chip Pre-Classifier on Wireless ECG Monitoring

by

Alexandru Samachisa

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Computer Engineering

Supervised by

Assistant Professor Dr. Marcin Lukowiak
Department of Computer Engineering
Kate Gleason College of Engineering
Rochester Institute of Technology
Rochester, New York
August 2007

Approved By:

Dr. Marcin Lukowiak
Assistant Professor
Primary Adviser

Dr. Daniel B. Phillips
Associate Professor, Department of Electrical Engineering

Dr. Fei Hu
Assistant Professor, Department of Computer Engineering

Thesis Release Permission Form

Rochester Institute of Technology

Kate Gleason College of Engineering

Title: Investigating the Effects of an On-Chip Pre-Classifier on Wireless ECG Monitoring

I, Alexandru Samachisa, hereby grant permission to the Wallace Memorial Library reporduce my thesis in whole or part.

Alexandru Samachisa

Date

Dedication

To my mom, for her continuous support and for always having faith in me.

Acknowledgments

I would like to thank my advisers: Dr. Marcin Lukowiak for his continuous guidance in the completion of this work, Dr. Daniel Phillips for his help in the biomedical engineering areas of this work, and Dr. Fei Hu for serving on my committee. I would also like to thank Dr. Juan Cockburn for his advice in parts of the work.

I am grateful to Daniel Fava and Justin Hnatow for their help in proofreading this work and to Meng Jiang and Inan Omer for their availability to answer questions about their research.

Abstract

In past years, heart disease has been the leading cause of death in most developed countries. Timely detection of a heart condition is necessary in order to prevent life threatening situations. Even when the problem is not a heart condition, the activity of the heart can supply vital information, which makes its monitoring extremely important.

A new approach to patient monitoring was taken recently by introducing wireless sensor networks into medical care. The capability of monitoring multiple patients at once makes such a system ideal for pre-hospital and in-hospital emergency care. The main problems associated with wireless sensor networks are power consumption and scaling. The power consumption is a problem due to the need for increased mobility of such a system, while scaling is of concern because a large number of nodes is desired in order to monitor more patients.

This thesis addresses the power and bandwidth problems associated with monitoring patients using wireless networks by introducing another level of signal processing at each node. The goal is to design a digital circuit that would detect any abnormality in the ECG signal and enable the data transmission only if such has occurred. Reducing the amount of data being transmitted reduces the necessary bandwidth for each node and with the introduction of the proposed chip, the power consumption of each node is affected as well.

Contents

List of Figures

List of Tables

Glossary

APC Atrial Premature Contraction, 53

CVDs Cardiovascular Diseases, 3, 4

ECG electrocardiogram, iv, 5

FIR Finite Impulse Response, 20–23, 34

PVC Premature Ventricular Contraction, 27, 28, 53

QRS the most noticeable shape in a heart beat formed by the three waves names Q, R, and S, 5, 6, 8, 15, 16, 20, 23–26, 28–30, 32, 36, 37, 39, 40, 44, 46, 47, 51, 52, 55

RF Radio Frequency, 18, 40

VHDL VHSIC Hardware Description Language, iv, 20, 29, 47–49, 51, 56

VLSI Very Large Scale Integrated Circuits, 19, 30, 41

WSN Wireless Sensor Network, 3, 5, 16–18

ambulatory ECG monitoring moitoring of the heart activities while the patient performs his usual everyday activities outside the hospital, 15

bundle branch block refers to an irregularity in the heart's electrical conduction system caused by an injury to the left or the right bundle branch in the heart, 53

motes sensor nodes, usually part of a wireless sensor network, 18, 58–61

myocardial ischemia pathological loss of or reduction in blood flow (ischemia) to a part of the muscular tissue of the heart (myocardium), 15

paced beat beat resulted from the action of an artificial pacemaker, a medical device used to regulate the beating of the heart, 53

Chapter 1

Introduction

Cardiovascular disease (CVD) is the number one underlying cause of death in most advanced countries in the world. In 2004, CVD accounted for 36.3% of the total number of deaths in the United States, but they were mentioned as a primary or secondary cause of death in 58.7% of the total number of deaths. Despite the fact that the death rates from CVD decreased by 25% from 1994 to 2004, CVD still accounted for more deaths than any other single cause every year since 1900, except for 1918. The direct and indirect cost of CVD for 2007 is estimated at $431.8 billion [1].

These alarming statistics justify the need for better heart monitoring in order to save lives. Timely detection of CVD can prevent later complications. Unfortunately, about two thirds of deaths due to CVD occur without any prior recognition of cardiac disease and/or outside the reach of appropriate medical attention [1]. This means constant heart monitoring needs to be extended to more than just hospitalized patients in critical condition, or patients that are known to suffer from CVD. This kind of monitoring is mostly intended to monitor patients at home, without influencing the patient's day-to-day life, while still allowing a fast response time in the event medical intervention is needed.

An important step in this direction was taken with the introduction of Wireless Sensor Networks (WSN) technology in medical applications. Leading research in this area is represented by the CodeBlue project at Harvard University [2]. The portable ECG sensors utilized can transmit the patient's heart data through a wireless connection for recording

and/or further analysis. The recording of the data allows medical professionals to diagnose patients based on their heart activity history, while real time processing can provide a way to detect if immediate medical intervention is needed. Besides the functionality provided, such a system needs to be scalable to meet the need for monitoring a large number of patients. Another goal is to reduce the power consumption at the nodes, in order to increase their portability by extending their battery life. The general approach to increase the bandwidth and reduce power in WSN is to reduce the amount of data being transmitted by preprocessing or compression.

The purpose of this thesis is to study the effects of removing normal data from the ECG signal before transmission over the wireless connection, data which can be considered irrelevant to the diagnosis of the patient. This approach would introduce another level of processing at each node, but reduce the amount of data traveling through the network. The necessary bandwidth will be reduced, making the system more scalable. The power consumption of the wireless transmitter will be decreased at the cost of adding the power requirements of the new processing stage. This modification will be most beneficial if the monitoring is extended to patients with no signs of CVD, since these will have regular heartbeats most of the time and, thus, will ideally have no need to transmit data over the network.

Chapter 2

Background

This chapter provides a brief introduction to the subjects of ECG, ECG signal processing, wavelets and artificial neural networks, which are used for ECG signal processing in this work, and WSN used for ECG monitoring. It is by no means a thorough tutorial for any of the above topics, but merely an attempt to provide some basic knowledge necessary to understand the underlying research work that went into preparing for this thesis.

2.1 ECG

The entire section is based on Grauer's practical guide to ECG interpretation [3]

The standard for analyzing heart activity over time is the electrocardiogram (ECG or EKG). The ECG is a graphic representation of the electric heart activity, which is obtained by measuring the electrical potential differences between points of the body. In order to perform an ECG, self-adhesive electrodes are attached to fixed predefined locations on clear skin. Each measurement between two points is called a lead. Each lead is defined by end points of the measurement. For example, Lead I measures the electric potential difference between the left leg (+) and the right arm (-). Based on these locations, each lead records the heart's electrical potential from a particular vantage point and can contain unique information relative to the other leads. The optimal compromise between time-efficiency and completeness of the ECG is provided by a set to 12 leads, which is considered the standard ECG. The 12 leads are: I, II, III, aVR, aVL, aVF, V1, V2, V3, V4, V5 and V6.

A typical 12-lead ECG can be seen in Figure 2.1 taken from [4], where each row contains 3 leads. The first row contains leads I, aVR, V1, and V4, the second contains leads II, aVL, V2, and V5, and the third contains leads III, aVR, V3, and V6, all separated by small vertical lines.

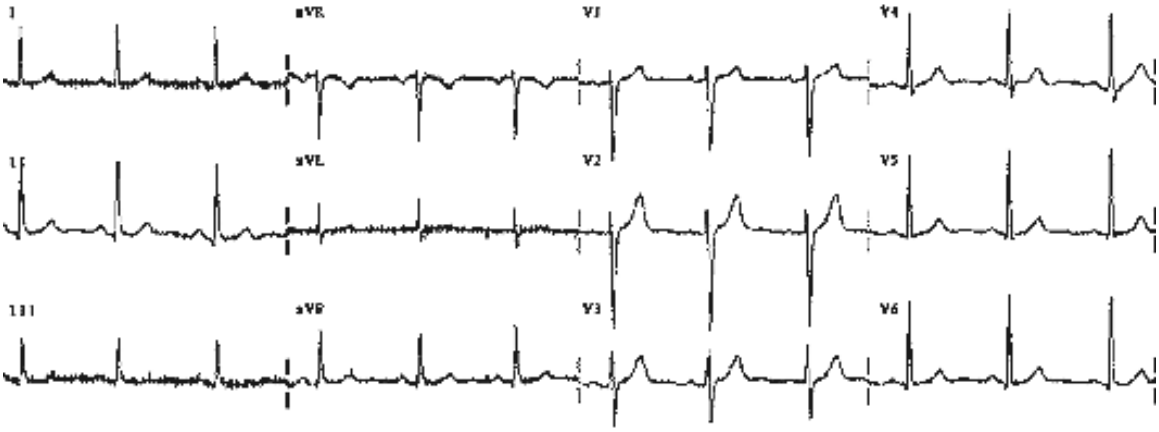

Figure 2.1: A normal adult 12-lead ECG

Figure 2.2 shows a typical ECG beat. A beat is composed of a P wave, a QRS complex and a T wave, and sometimes a U wave following the T wave. The shape of the waveforms depicted by the ECG depend on the orientation of the individual lead, with respect to the direction of the polarization wave associated with the synchronized contraction of the heart muscle or myocardium.

The waves are a pure reflection of the heart's electrical activity, but from a medical perspective, the actual mechanical events are important in diagnosis as well. These events are defined as segments on the ECG. Some of the useful segments from the analysis point of view are the PR, ST and QT segments, which are also shown in Figure 2.2. The PR interval is measured from the onset of the P wave to the beginning of the QRS complex. The ST segment is defined as the duration from the end of the QRS until the onset of the T wave, while the QT is the ST segment plus the QRS complex.

All the waves and segment lengths contain information about the state of the heart and can be used to detect any abnormalities in the heart activity of the patient being observed.

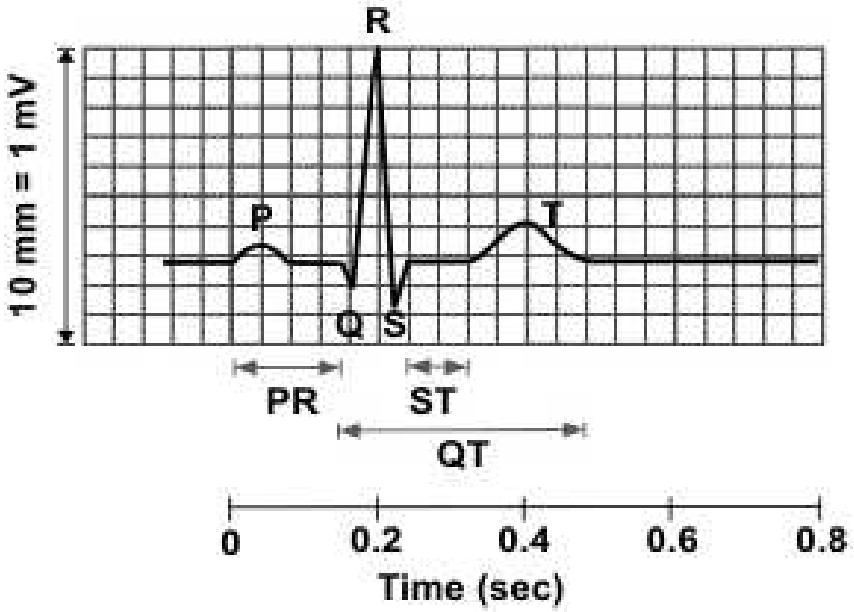

Figure 2.2: Typical ECG heart beat

For example, the P wave rate is can indicate the heart rate, which is considered normal only when it lies between 60 and 100 beats per minute with less than 10% variation.

2.2 ECG Signal Analysis

In order to continuously monitor a patient, one or more ECG signals similar to the one shown in Figure 2.2 would have to be observed at all times. Unfortunately, it is impractical to have each patient constantly observed by a doctor, even if the data is centralized and a doctor could monitor multiple patients at once.

Computer based automated signal analysis can easily handle the large amounts of ECG data and provide an alert only if human intervention is indicated. Even expert electrocardiographers agree that computers can, at the very least, serve as backup in processing the large amounts of data, where due to the limited amount of time, important findings can be overlooked by the electrocardiographers [3].

Section 2.2.1 presents the main goals of ECG signal analysis together with possible

solutions, while sections 2.2.2 and 2.2.3 give a short background on two of the most commonly used techniques used in ECG signal processing: Wavelets and Artificial Neural Networks.

2.2.1 Main Goals of ECG Analysis

One of the most important problems in ECG analysis is automatic beat identification. This is needed in many cases ranging from simple heart rate computations to serving as the first stage of complex automatic diagnosis. Beat identification techniques have to start by identifying features in the ECG signal that can be constantly detected in each heart beat.

Simply by looking at an ECG plot, it is noticed that the QRS complex is the predominant feature. The other features of the ECG signal, like the P wave and T wave, are sometimes too small to be detected, or have characteristics too close to those of noise. This makes the QRS complex the feature that can yield the best detection accuracy. As an example, Cuiwei obtained an accuracy of over 99.8% in detecting the QRS complexes over the entire MIT-BIH in [5], through the use of wavelets. The high reliability of its detection made the QRS detection the basis for all the automated ECG signal analysis from simple heart rate determination to complex classification schemes of the cardiac cycles. Most beat identification techniques use the QRS as a staring point. For example, one possible approach is selecting a fixed interval before and after the detected QRS for further analysis [6].

The next step in ECG analysis is to further process the extracted beats to determine the state of the patient. The beat classification problem is defined as successfully detecting what type of beat is being analyzed upon its extraction from the ECG signal. This problem can be solved through one of the many classification algorithms available. A large number of existing signal processing methods have been successfully used in ECG classification. These include Artificial Neural Networks [6], Block-Based Neural Networks [7], Hidden Markov models [8], and Support Vector Machines [9]. Besides the algorithm, the features chosen for the classifiers are also of great importance. In the case of ECG, the features

can be directly obtained from samples of the ECG signal, morphological descriptors, time-frequency descriptors, or a combination of the types [10]. The morphological features are usually features extracted from the segmentation of the ECG beat, for example the ST length, while the time-frequency descriptors can be coefficients of the Discrete Wavelet Transform (DWT) of the ECG waveform.

2.2.2 Wavelets

The theory in this section is based on *Ten lectures on wavelets* [11] by Daubechies, and on *Elements of wavelets for engineers and scientists* [12] by Mix and Olejniczak.

The first step in performing signal analysis is choosing the domain to be used. The frequency domain would be useful because of the possibility of removing noise that has specific frequency bands. For example baseline drift, a low frequency type of noise, and muscular contractions, a high frequency type of noise, are very common in ECG signals [13]. Unfortunately, this approach would lose all the time information, which sometimes is the only sign of a problem (e.g. accelerated heart rhythm). The time domain would keep the location in time of the points of interest, but it could make the feature detection harder, due to noise and ECG variance. The ECG varies across different patients and even across different beats from the same patient. An example of the latter is shown in Figure 2.3, where the amplitude and shapes are different from one beat to the next, even though all the beats are normal activity beats.

A middle ground is provided by wavelet analysis, which allow evaluation of a signal in both time and frequency domains. Wavelets can decompose the signal into different frequency bands, but at the same time keep the time information, thus making them a perfect tool for ECG analysis. Representative studies of ECG analysis is provided by [14, 5, 6, 13].

A wavelet is a waveform of effectively limited duration that has an average value of zero. A sample wavelet is presented in Figure 2.4. The wavelet chosen for the wavelet transform is also known as the mother wavelet. The use of wavelets is similar to Fourier's

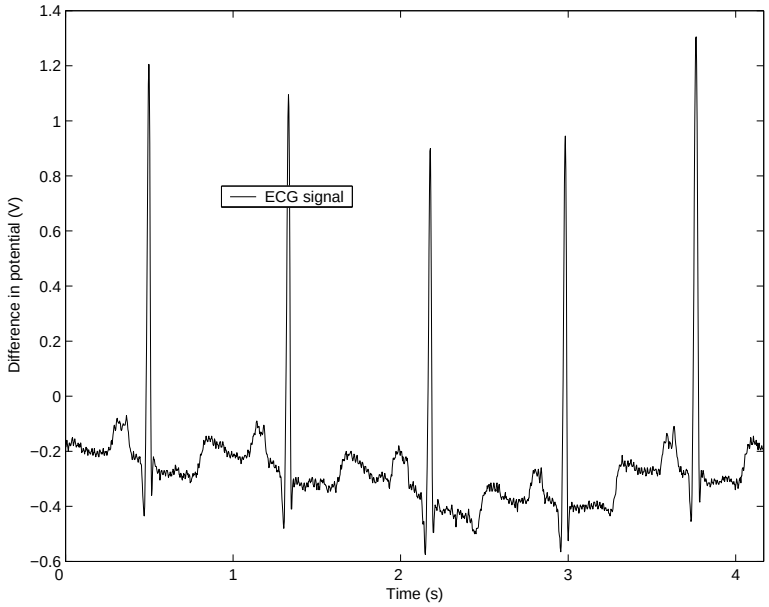

Figure 2.3: Portion of ECG signal from MIT-BIH database - Patient 100

idea of representing functions by superposing sines and cosines. The Fourier transform contains the frequency information of the signal, independent of the time at which the frequencies occur. Wavelets moved from Fourier's frequency analysis to scale analysis. This means the signal is divided in scales of resolution rather than different frequencies. This is achieved by superposing shifted and scaled versions of the mother wavelet to approximate the signal. The wavelet transform can be continuous (CWT) or discrete (DWT), just like the Fourrier transform. The general formula for the CWT is presented in equation (2.1).

$$CWT_f(b,a) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} f(t)\psi\left(\frac{t-b}{a}\right) dt \tag{2.1}$$

where the two parameters a and b describe the scaling and shifting respectively of the wavelet $\psi(t)$. As the scaling factor (a) changes, different frequency ranges are covered, and as the shifting factor (b) changes, the time localization of the waveform.

The DWT formula is presented in equation (2.2), which is obtained from equation (2.1) by simply restricting a and b to discrete values.

Figure 2.4: Mother Wavelet Example

$$DWT_f(m,n) = \frac{1}{\sqrt{|a_0^m|}} \int_{-\infty}^{\infty} f(t)\psi\left(\frac{t - nb_0a_0^m}{a_0^m}\right) dt \tag{2.2}$$

In equation (2.2) a_0^m represents the scaling factor and $nb_0a_0^m$ represents the time shift, where $a_0 > 1$, $b_0 > 0$, and m and n are integers.

Both equations (2.1) and (2.2) assume that $\psi(t)$ satisfies:

$$\int \psi(t)dt = 0 \tag{2.3}$$

Based on equation (2.1) and equation (2.2), the general steps in finding the wavelet transform of a given signal can be described as:

1. Start at the beginning of the signal and compare it with the mother wavelet by correlation
2. Shift the wavelet to the right and go back to step one. Continue until the entire signal is covered
3. Scale (stretch) the mother wavelet and repeat steps 1 and 2
4. Repeat steps 1 through 3 for as many scales as needed

The 4 steps described above can be applied in continuous fashion or in discrete steps.

The CWT following the steps above would produce a two dimensional continuous domain of coefficients indexed by scaling and shifting (a and b), while the DWT will have discrete scaling and shifting (m and n), thus producing a discrete two dimensional domain.

However, in practice, a different method for obtaining the DWT is used. The method involves using a low pass and a high pass filters at each level to obtain the coefficients at the next scale. This can be explained because small values of $|a|$ in equation (2.1) correspond to high frequencies or very fine scale, while large values correspond to low frequencies. The equations for these transformations are (2.4) and (2.5), where j is the scale, H is the low pass filter and G is the high pass filter.

$$c_{j+1}(k) = \sum H(m - 2k)c_j(m) \tag{2.4}$$

$$d_{j+1}(k) = \sum G(m - 2k)c_j(m) \tag{2.5}$$

The choice of scales chosen for this approach is the dyadic scale. In order to obtain the next dyadic scale according to the equations (2.4) and (2.5), the signals have to be down sampled and then passed through the same filters over and over as shown in Figure 2.5, where c_0 is the signal to be processed.

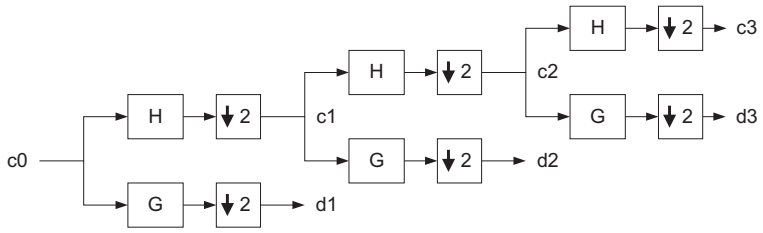

Figure 2.5: General schematic of a practical implementation of the DWT

Based on the use of the low pass and high pass filters, the d_i coefficients can be interpreted as the details of the signal in a specific frequency band, defined by the chain of filters leading to d_i.

2.2.3 Artificial Neural Networks (ANN)

A common approach to classification problems, which include the beat classification problem described in section 2.2.1, involves the use of Artificial Neural Networks (ANN). The entire background on ANN given in this section is based on [15].

An ANN is a processing system that attempts to emulate a biological neural network, in an attempt to gain its performance characteristics. The ANN is composed of nodes (neurons) and weighted links among them, which are used to multiply the signal transmitted through that link and thus simulate a simplified neuron interactions model. An example of an ANN is presented in Figure 2.6.

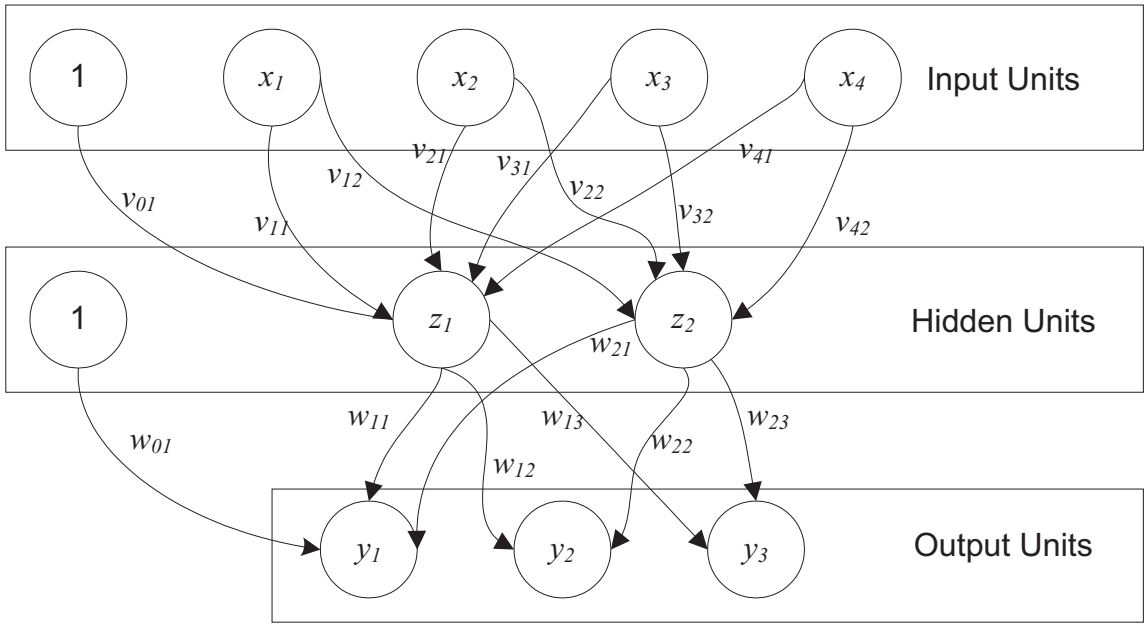

Figure 2.6: Small single hidden layer neural network

The input units are the chosen features for the classification. These features are multiplied by the weights associated with all the links and act as inputs to the next layer. Let the input feature vector be $x = (x_1, ..., x_i, ..., x_n)$ and let the weight of the link connecting x_i to z_j be v_{ij}. This means the input to the hidden node z_j is given by:

$$z\_in_j = v_{0j} + \sum x_i v_{ij} \tag{2.6}$$

Each neuron layer has an activation function associated with it. A typical activation function is a sigmoid like the one defined in equation (2.7). The sigmoid shape is showed in Figure 2.7.

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2.7}$$

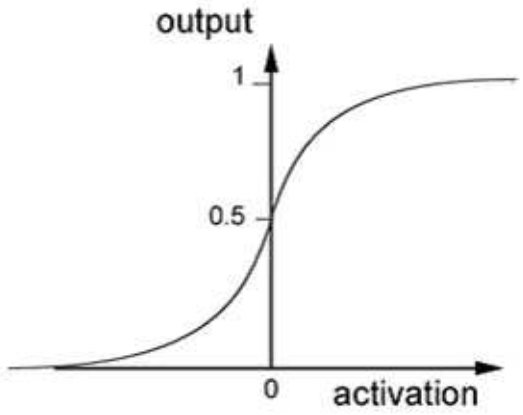

Figure 2.7: Typical activation function.

Based on the chosen activation function the output value form each node is computed as shown in equation (2.8), where f is the chosen activation function.

$$z_j = f(z\_in_j) \tag{2.8}$$

The next steps are computing the value at the inputs of the output nodes y_in_k as shown in equation (2.9), and the output values based on the activation functions of these output nodes y_k as shown in equation (2.10), where w_{ij} is the weight of the link connecting hidden node z_i to output node y_j. The output values y_k form the response of the ANN to the input vector.

$$y\_in_k = w_{0k} + \sum z_j w_{jk} \tag{2.9}$$

$$y_k = f(y\_in_j) \tag{2.10}$$

There is no rule for deciding the number of hidden layers, the number of hidden units in each hidden layer, or the activation function. These features are usually determined empirically.

Determining the appropriate weight values for each of the connections in an ANN based on the problem requirements is called the training of the network. The training of a multi-layer ANN is usually performed by using the backpropagation training method. This method was the main reason ANN became popular again after a quiet period in the 1970. The lost interest was due to the failure of the single layer network to solve even simple problems like the XOR function, and the lack of a method to train a multi-layer network. As the name suggests, the backpropagation method traverses the ANN in reversed direction during the training phase.

In the beginning, the weights v_{ij} and w_{kl} are usually initialized to random small values. A small training set of representative input vectors is selected, for which the expected output is already known. These will be run through the ANN, and the weights will be modified according to the algorithm until a certain condition is met. This condition is usually reaching a given small error value obtained when comparing the ANN responses with the desired responses to the training vectors, or reaching a maximum given number of iterations in the training algorithm, also known as epochs. The latter is used as a backup in case the small error condition cannot be met. The error can also be computed using a set different than the training set. Using a separate test group to compute the error is preferred to avoid over-fitting, making the network respond better to new inputs, not present in the training set.

2.2.4 Standard ECG Databases

Since the beginning of computerized ECG analysis, research was difficult due to the lack of means to validate new algorithms and techniques. This created the need for ECG databases annotated by expert ECG interpreters. The differences between the databases are the leads chosen, the digitization sampling, and most importantly, the annotations present in the

databases. Three popular databases are the MIT-BIH Arrhythmia Database [16], the European ST-T Database [17] and the QT database [18].

The MIT-BIH Arrhythmia Database [16] was a major product of the arrhythmia analysis research performed at Boston's Beth Israel Hospital and MIT, which began its distribution in 1980. The database contains 48 half hour excerpts of two-channel ambulatory ECG recordings obtained from 47 subjects, digitized at 360 samples per second. The recordings are fully annotated by cardiologists. The annotations include the R wave peak locations and the type of the beats. The database became so popular that most subsequent databases adopted the MIT-BIH data format for storing the recordings. This allows for all the software created for the MIT-BIH database to be used to access the other databases as well.

The European ST-T Database [17] was created to originally assess the quality of ambulatory ECG monitoring. Thirteen research groups from 8 countries contributed to the creation of the database. The recorded data is composed of 2 hour records of continuous two-channel ECG records, digitized at a rate of 250 samples per second. Each record contains at least one episode related to myocardial ischemia. In each case, the two leads most likely to reveal ST-T changes were recorded. The annotations present in the ST-T database include QRS, beat types, rhythm, signal quality changes, T wave onset, peak, and offset and ST segment. The focus of the ST-T database was to detect ST-T changes as the name suggests.

The QT database [18] is aimed at ECG interval based research. The motivation for creating this database was that the research based on ECG segment lengths was lacking a database with manually made measurements of the wave boundaries. The records were chosen mainly from the existing MIT-BH Arrhythmia Database, the ST - T Database and several others. From each record, a subset of 30-100 beats was chosen and annotated with the beginning, peak and end of the P wave, beginning and end of the QRS complex, the peak and end of the T wave and if present the peak and end of the U wave.

All the mentioned ECG databases can be downloaded freely from PhysioNet [19], together with software necessary to access the databases.

2.3 Existing Wireless ECG Monitoring System

One of the main concerns in patient monitoring is reducing the number of medical personnel needed for everyday monitoring. One solution is to centralize the data from patients in one location so that one doctor can monitor multiple patients at once. This can be done by connecting all the monitoring devices to a base station. One solution to centralize ECG data from multiple patients at once is to use a WSN. This removes the need for wires and also allows the patient to move around freely. Such a system is proposed and implemented in [20].

The sensor node of the system implemented in [20] is shown in Figure 2.8. This device is carried around by the patient being monitored. This approach allows the possibility of monitoring patients while conducting normal everyday activities. Another advantage is the ability to setup a small area network in a very short period of time. This could be useful in a situation where, for example, an accident occurred which resulted in many victims that need to be monitored immediately, before they reach a hospital.

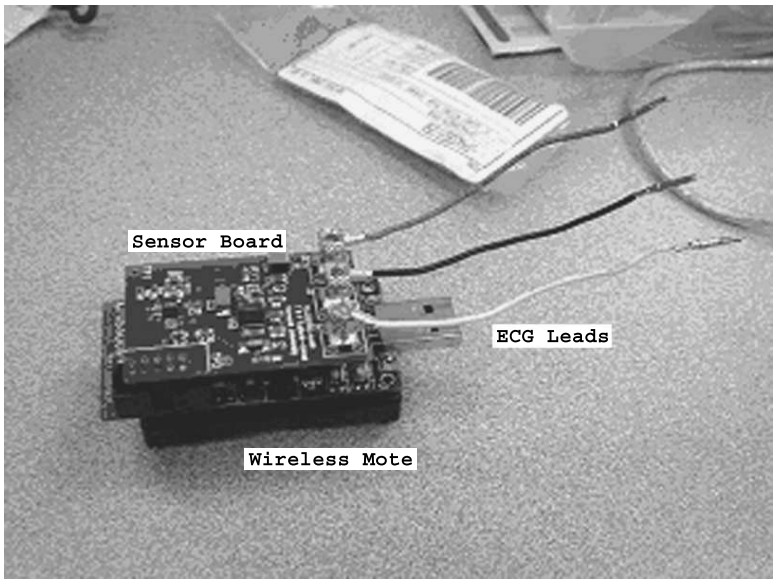

Figure 2.8: ECG sensor node in a WSN

Two of the main concerns in such a system are power consumption at the nodes and scalability of the network, i.e. the numbers of nodes it can accommodate. The power

consumption directly affects the useful lifetime of the system; the smaller the power consumption, the longer the system can operate. Scaling is a problem associated with WSN in general because multiple wireless nodes in the same physical area have to share the same bandwidth, and thus reduce the bandwidth available at each node.

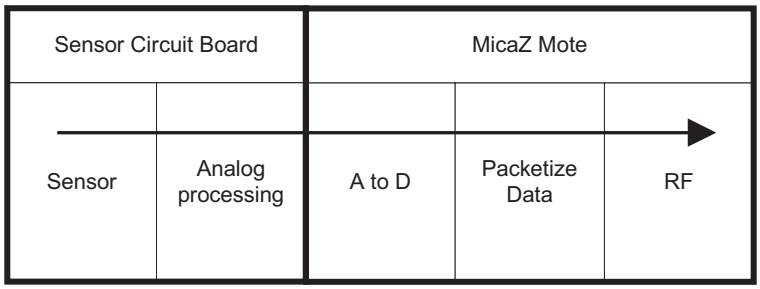

Figure 2.9: Block diagram of the node in Figure 2.8

One way to improve bandwidth would be to compress the ECG data before transmission. This would require modifications to the data packetizing block presented in Figure 2.9. Due to the importance of the details in the signal, an algorithm that attempts to compress it at any other level than lossless becomes very complex. The complexity is due to the difficulty in successfully deciding which data can be discarded. One approach to ECG compression is presented in [21]. The algorithm takes into account the time critical nature of the data, and it manages to perform all the encoding and decoding in under 0.5 seconds.

2.4 Thesis Objectives

The motivation for this work was based on the general statistics of the MIT-BH database, where even among patients with problems, over 70% of the beats are normal [16], and thus of little interest in diagnosis. Added to this, the power specifications of the most common RF transmitters used in WSN today presented in Table 2.1 show an over 35 mW active power consumption for transmitting data (Tx Power), which is generally more than the active power consumed by the microcontrollers used on the respective motes (MC Power). The largest difference can be seen in the last column of Table 2.1, which shows that the

power requirements of the microcontrollers have decreased at a faster rate than the power requirements of the radios used for the wireless transmittion. The values in Table 2.1 were taken from [22].

Mote Type Year	WeC 1998	Rene 1999	Rene2 2000	Dot 2000	Mica 2001	Mica2Dot 2002	Mica2 2002	Telos 2004
Rx Power (mW)	9	9	9	9	12	29	29	38
Tx Power (mW)	36	36	36	36	36	42	42	35
MC Power (mW)	15	15	15	15	8	8	33	3
Data Rate (kbps)	10	10	10	10	40	38.4	38.4	250

Table 2.1: Power specification summary for the motes communication component

This thesis investigated the effects on bandwidth and power consumption resulting from the introduction of another level of processing between the analog to digital converter and the data packetization block in the nodes of an existing system like the one proposed in [20]. The placement of the new block is illustrated in Figure 2.10. The processing block performs signal analysis and attempts to remove as much of the normal data as possible, so that less data travels across the network. The pre-classifier takes the incoming digital ECG signal, isolates the individual beats, detects as many of the normal beats as possible and only enables transmission of the abnormal beats that would require further analysis. This can be combined with another more complex software classifier at the receiving end to process the signal further and decide whether human attention is required or not.

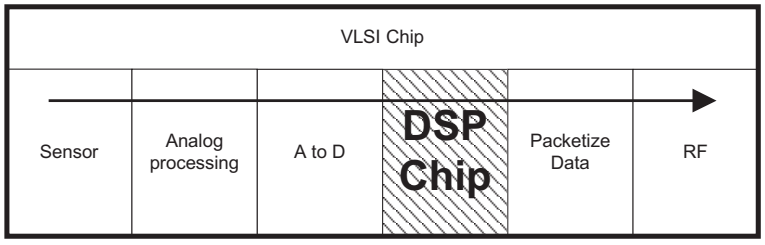

Figure 2.10: Location of the pre-classifier in the node 2.10

In order to achieve the highest power savings and integration, the entire system could be implemented on a single VLSI chip as a very low power solution. A VLSI architecture of the first three blocks before the pre-classifier (the sensor, the analog processing and the

A to D converter) is presented in [23]. This can be used as the front end to the pre-classifier proposed in this work, which can be combined later with a VLSI design of the last two blocks to obtain a complete on chip solution to a low power ECG Wireless Sensor Network Node.

The proposed solution was analyzed from a bandwidth usage point of view as well as its power requirements. The bandwidth improvement is certain, but the exact impact is entirely based on the accuracy of the classifier. The power analysis compares the estimated total power savings of the wireless transmission to the estimated power requirements of the newly proposed chip. The power savings come from transmitting less data and thus having the radio transmitter turned off for the amount of time data is not transmitted, as opposed to the continuous transmission currently in place.

Chapter 3

Methods and Supporting Work

This chapter describes the methodology used to investigate the performance of the proposed circuit. The chapter is divided into 3 parts. The first part, section 3.1, describes the algorithms chosen to process the ECG signal, and the modifications needed to better fit the purposes of the current work. MATLAB was chosen to implement the modified algorithms and to asses the performance of the implemented algorithms with the MIT-BIH database. Section 3.2, describes the digital design of the algorithms chosen in section 3.1. ModelSim was used to write the VHDL model of the digital design. The third part, section 3.4, describes the tools used for the synthesis of the VHDL model in order to obtain power consumption, delay, and area estimates for the proposed circuit.

3.1 Processing of the ECG Signal

Wavelets were chosen to start the processing of the signal due to promising results in the processing of the ECG [14, 5, 6, 13]. The wavelets based algorithms chosen for processing the ECG signal are the QRS detection algorithm proposed in [14] followed by the feature extraction and classification methods proposed in [6]. The two algorithms were chosen because they both require the wavelet transforms of the ECG signal using the same mother wavelet, both have high accuracies over the MIT-BIH database, and most importantly, their combination leads to one possible solution to solve the larger problem of identifying abnormal beats from the raw ECG signal.

3.1.1 Computing the Discrete Wavelet Transform (DWT)

The DWT can be modeled using FIR filters, whose coefficients can be computed based on the mother wavelet used. The mother wavelet used in this thesis is a quadratic spline wavelet, successfully used for the first time to process ECG signals in [5]. The wavelet is defined by its Fourrier transform in equation (3.1) and shown in Figure 3.1, which was taken from [24].

$$\hat{\psi}(w) = jw \left(\frac{sin(w/4)}{w/4} \right)^4 \tag{3.1}$$

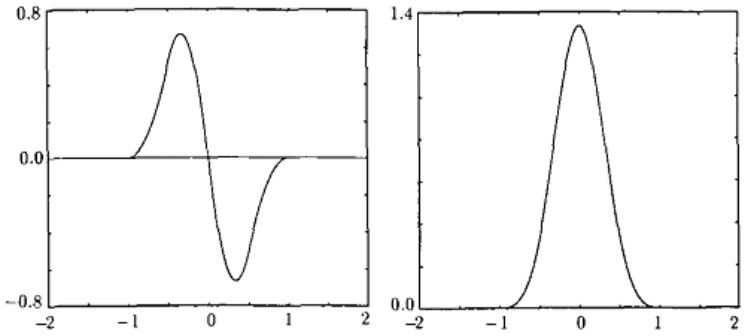

Figure 3.1: Spline wavelet with compact support and one vanishing moment (left) and its primitive (right)

The filters needed to implement the DWT based on the quadratic spline wavelet are $H(w)$ and $G(w)$, as defined by their frequency response in equations (3.2) and (3.3), which were taken from [5]. The two filter responses lead to the transfer functions defined in equations (3.4) and (3.5). These equations give the coefficients for the two FIR filters needed to be implemented, as well as the delays introduced by each filter.

$$H(w) = e^{jw/2} \left(\cos \frac{w}{2} \right)^3 \tag{3.2}$$

$$G(w) = 4ie^{jw/2} \left(\sin \frac{w}{2} \right) \tag{3.3}$$

$$H(z) = \frac{1}{8}z^2 + \frac{3}{8}z + \frac{3}{8} + \frac{1}{8}z^{-1} \tag{3.4}$$

$$G(z) = 2z - 2 \tag{3.5}$$

In order to obtain the different DWT scales needed, the filters H and G defined in equations (3.4) and (3.5) have to be cascaded as shown in Figure 3.2. The filter responses of the 5 FIR filters needed to obtain the first 5 DWT coefficients based on the quadratic spline wavelet are presented in Figure 3.3. Each filter is a band pass filter built from the convolution of the filters between c_0 and d_i as shown in Figure 2.5.

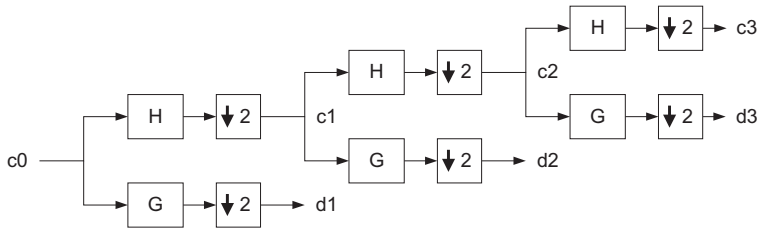

Figure 3.2: Using the H and G filters to obtain the DWT scales

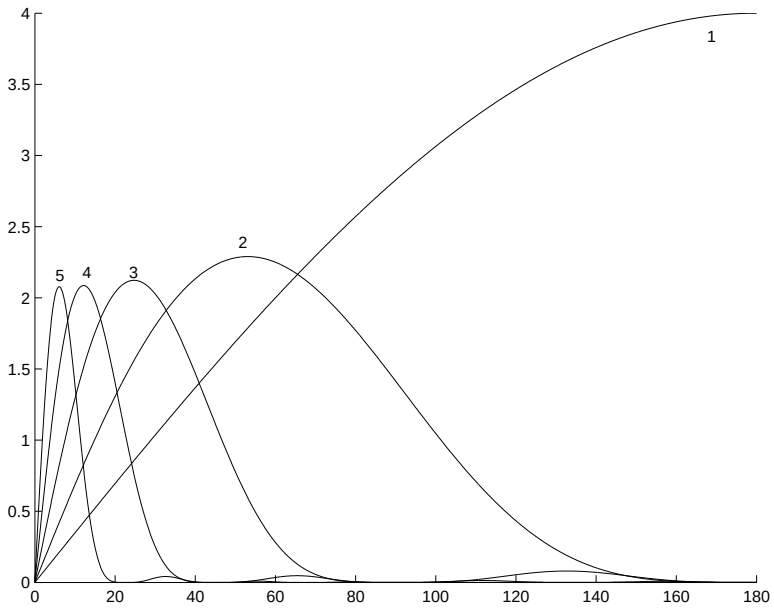

Figure 3.3: Frequency Response of the first 5 DWT scales used

The down sampling by a factor of two followed by filtering with a filter F is the same as filtering a filter F_1 obtained by introducing zeros between the coefficients of F and then

down sampling. Due to this the approach taken in [14] was to modify the filters to include the down sampling. The new cascaded design is shown in Figure 3.4, where H_p and G_p are filters obtained by introducing $2^{p-1} - 1$ zero coefficients between all the coefficients of H and G respectively. The difference is that the new outputs e_i and f_i have the same sampling rate as the original signal. The old outputs c_i and d_i can be obtained by simply down sampling the new ones by the appropriate factor.

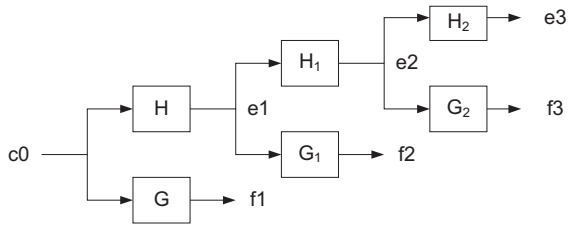

Figure 3.4: Modified DWT design

Figure 3.5 shows an example of applying the filters to an ECG signal from the MIT-BIH database. It can be noticed that the time component is kept on the horizontal axis, while the vertical axis is similar to the frequency domain due to the band pass characteristic of the filters.

3.1.2 QRS Detection

The algorithm proposed in [14] was chosen for QRS detection due to its accuracy of 99.7% over the MIT-BH database and due to the fact that it is geared towards real-time analysis unlike other algorithms [13, 5]. It uses the DWT coefficients computed in section 3.1.1, in equation (3.4). The algorithm reslies on the characteristics of QRS like shapes across the first 5 DWT scales, which are to generate zero crossings flanked by opposite sign peaks. An example can be seen in Figure 3.6, where the QRS-like shapes in the top signal generate the zero crossings shown in the bottom signal, which is the 5th DWT sub band of the signal shown on the top.

The algorithm first detects all the zero crossings flanked by a positive and a negative peak larger than a chosen threshold in the first five scales of the DWT. A different threshold

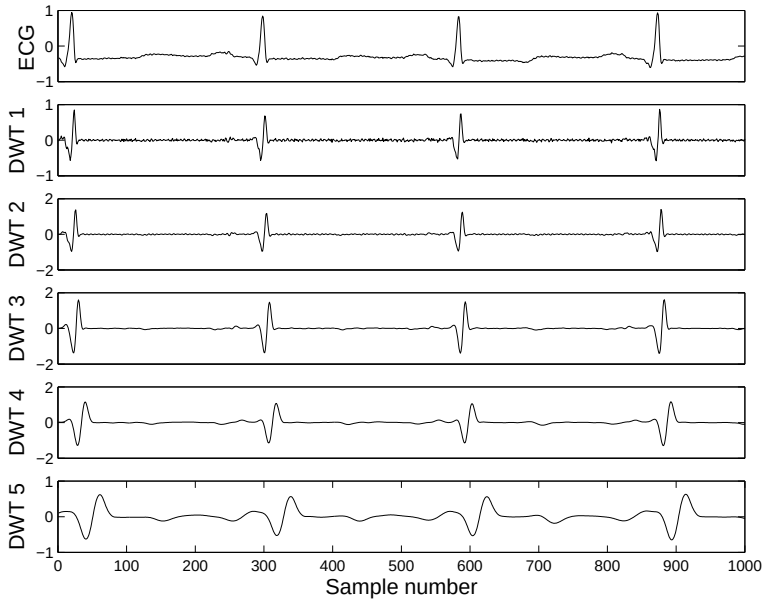

Figure 3.5: First five scales of the DWT

is chosen for each of the scales. If the location of a zero crossing is the same across all the scales, an R wave is assumed to be found, which is the highest peak of the QRS complex. The delay introduced by the DWT FIR filters has to be taken into account when determining the locations of the zero crossings in order to match their locations. It is shown in [5] that if a uniphase wave symmetric to its peak is present in the signal to be analyzed, the corresponding zero-crossing in the i^{th} DWT scale is delayed by $2^{i-1} - 1$ points. However, if the wave is asymmetric, the delay will vary. This is shown in Figure 3.6, where the first QRS shape is symmetric and the next two are asymmetric. The delay is shown with respect to the 5th DWT scale, and the delay difference is shown by the three black arrows. The error compared to the $2^{i-1} - 1$ delay is increasing with the scale i and the degree of asymmetry of the wave, but it is tolerant for the first 5 DWT scales needed.

The high level algorithm flow proposed in [14] is shown in Figure 3.7.

The main difference between the work in [5, 14] and the current work is the sampling rate of the input signal. In the current work, the sampling rate is kept at 360 Hz, which is the sampling rate of the MIT-BIH standard database, as opposed to the 250 Hz rate used in [5, 14]. This was decided in order to match the requirements of the work in [6], which is

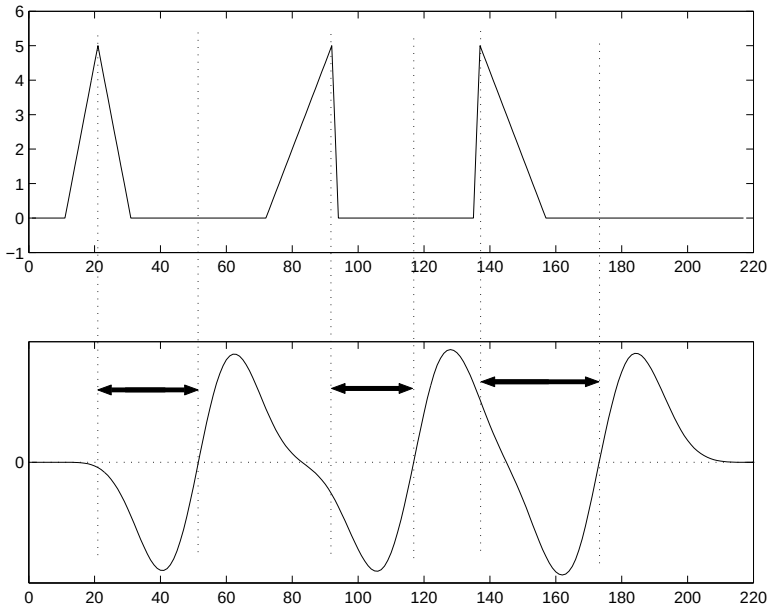

Figure 3.6: Symmetric and asymmetric QRS shapes (top), the 5th DWT scale of the signal (bottom)

used to classify the beats after they are isolated. This change will modify the bandwidths of the filters used by [5, 14] to the ones already shown in Figure 3.3. A comparison between the 3-dB bandwidths of the two sets of filters is shown in Table 3.1. It can be noticed that the bandwidth limits are different mostly for the high frequency limits. The decision was made to use the same filters as defined in equations (3.4) and (3.5), and modify the QRS detection algorithm based on the detection results to match the needs of the current work.

	3 db bandwidth (Hz)	
scale	sampling at 250 Hz	sampling at 360 Hz
$s = 2^1$	62.5 - 125.0	60.5 - 180.0
$s = 2^2$	18.0 - 58.5	17.6 - 97.0
$s = 2^3$	8.0 - 27.0	8.4 - 45.7
$s = 2^4$	4.0 - 13.5	4.2 - 22.2
$s = 2^5$	2.0 - 6.5	2.1 - 11.3

Table 3.1: 3-dB Bandwidth comparison for the filters for different sampling rates

The algorithm was implemented as described in MATLAB. All the thresholds and other

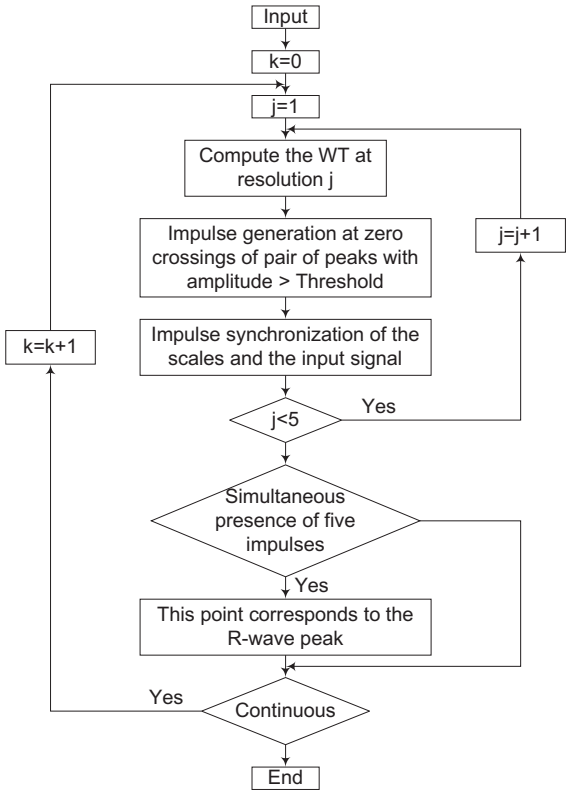

Figure 3.7: R waves detection algorithm

variables were obtained empirically with the goal of maximizing the QRS detection accuracy over the MIT-BIH database. These variables include the thresholds detecting the peaks at each scale, the allowed delay error values for non symmetric waves, and the distance from the annotation where a detection is considered successful. The distance from the actual QRS annotation has to be variable because the annotations are not always exactly on the sample at the peak of the R wave.

In the process of improving the QRS detection accuracy, it was observed that most of the misses are due to the fifth DWT scale, because some QRS complexes are hard to observe at this scale. The algorithm was ran without the fifth DWT scale altogether, and an improvement in the detection accuracy was seen, without an increase in false detections. Due to these results, the decision was made to only use the first four DWT scales in the current work, unlike the algorithm used in [14].

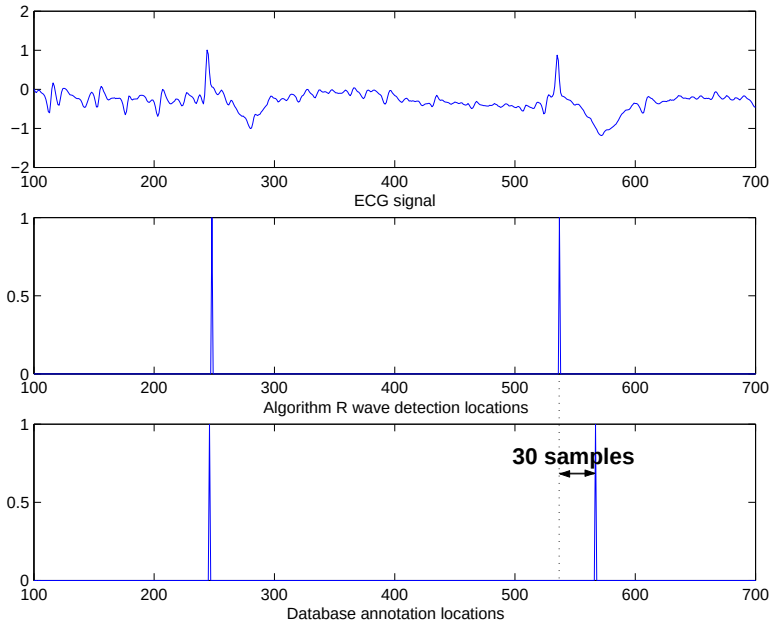

Figure 3.8: ECG signal (top) with the R wave detections (middle) and database annotations (bottom)

3.1.3 Beat Classification

The algorithm chosen for beat classification is the one presented in [6], which has an accuracy of 95.2% over the MIT-BIH database. The algorithm starts by extracting the features needed for the classification based on the QRS location of the beat under analysis. The next step is to use a neural network to classify the beat.

The features are chosen from a fixed size window of 700 ms, with 300 ms before the detected QRS complex and the 400 ms immediately following it, as done in [6]. Based on the sampling frequency of 360 Hz used, there are 252 samples in the fixed window. The best results obtained in [6] are based on 43 features. 42 of the features are the fixed size window samples of the 4^{th} scale of the DWT transform down sampled by a factor of 6, and the last feature is the RR-interval ratio, defined in equation (3.6), where T_i is the time at which the R-wave for beat i occurs.

$$IR_i = \frac{T_i - T_{i-1}}{T_{i+1} - T_i} \tag{3.6}$$

The neural network implemented for classification in [6] is a feed-forward multi-layer perceptron with a single hidden layer. This configuration was chosen empirically over other configurations including radial basis neural networks and multiple hidden layers. The number of neurons in the hidden layer was chosen to be 43, equal to the number of inputs.

The classifier proposed in [6] is classifying the beats into three categories: normal beats, premature ventricular contractions (PVC), and the rest. The current work only needs two categories: normal and abnormal. The difference is that two of the bins in [6] become one, which should improve the accuracy of the classifier. The need for a hardware implementation on the other hand imposes constraints over the classifier that could reduce the accuracy of the classifier. One such example is the fact that the classifier presented in [6] normalizes the features extracted from the 4th scale of the DWT to a mean of zero and a standard deviation of unity. Due to the complexity of the normalization operation from the hardware perspective and the need for a low power circuit, the decision was made to skip the normalization step.

The second difference is that the RR-interval ratio defined in equation (3.6) was replaced in the current work with just $T_i - T_{i-1}$. The main reason for this change is that in the current real-time implementation, only one beat is available for processing at a time, and thus T_{i+1} is not available at when processing beat i. If the feature defined in equation (3.6) is used, an extra delay of one beat would be added to the receiving end and the design also need to implement the division operation, which is a very complex operation from the hardware perspective. On the plus side, the feature defined in equation (3.6) was chosen in [6] mainly to improve the separation between PVC beats and other type of normal beats, categories that are merged together in this work.

The classifier proposed in [6] is using samples from the 4^{th} scale of the DWT to classify the beats. The QRS detection problem was ignored and the samples were chosen with respect to the annotations present in the MIT-BIH database. This approach cannot be taken in the current work because the QRS detection can be as far as 30 samples away from the database annotation as shown in Figure 3.8. Clearly the features extracted with and offset

as large as 30 samples will be very different, but figure 3.9 shows that even in a case where the QRS detection is done only one sample away from the annotation, the error between the features extracted is significant. This leads to the conclusion that the training set should be built from beats extracted with the QRS detection algorithm implemented in section 3.1.2, rather than extracting beats based on their annotations. This will ensure that the beats used for training will be a better match for the beats being extracted and classified in real time, and thus provide a better classification accuracy for the combined algorithms.

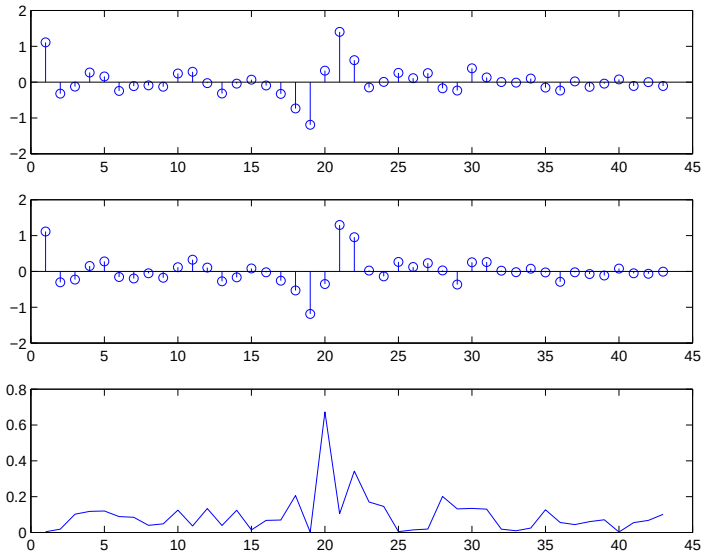

Figure 3.9: Features of a beat extracted with respect to database annotation (top), with respect to R-wave detection (middle), and the difference between the two (bottom)

3.2 Digital Implementation of the ECG Processing Algorithms

After implementing the QRS detection and beat classification algorithms in MATLAB and testing their accuracy over the MIT-BIH database, a VHDL implementation to match the MATLAB model is needed. The VHDL model needs to be synthesizable, so that the power consumption, delay and area estimates can be extracted from the synthesized design.

The first section in this chapter, section 3.2.1, presents general considerations for the chip design, which include low power design considerations and number representation. Sections 3.2.2, 3.2.3, and 3.2.5 describe the digital implementations of the algorithms chosen in 3.1.1, 3.1.2, and 3.1.3 respectively. Section 3.2.4 describes extra circuitry designed to handle the cases where the beats go undetected, while the last section presents a detailed analysis of the delay introduced by the proposed design.

3.2.1 Top Level Design Considerations

Designing for Low Power

The proposed chip's power requirements will add to the total power consumption at each node. One of the main goals of this work is to reduce the power consumption at the node, and thus the chip should be designed to consume as little power as possible. The power consumption of a VLSI chip can be divided into dynamic power consumption and static power consumption.

The dynamic power is defined in [25] as:

$$P_{avg} = P_{switching} + P_{shortcircuit} + P_{leakage} = \alpha C_L V_{dd}^2 f_{clk} + I_{sc} * V_{dd} + I_{leak} * V_{dd} \quad (3.7)$$

where α is the activity factor, which can take a value between 0 and 1. The activity factor describes the probability of switching in a clock period. As an example, the clock has an activity factor of 1. C_L is the capacitance that needs to be charged/discharged in the event of a switch. Sometimes another term is defined as $C_{effective} = \alpha * C_L$, which is the effective capacitance to be charged/discharged per clock period. I_{sc} is the short circuit current, which appears in the middle of the transition, when neither of the pull up and pull down networks are completely closed. I_{leak} is the leakage current that appears across a transistor in its off state.

There are many techniques that can be used to reduce any of the components of P_{avg}. Many techniques of reducing the power consumption are presented in [25]. Out of these,

a few were implemented in the current work, namely clock gating, resource sharing and multiplication with constants.

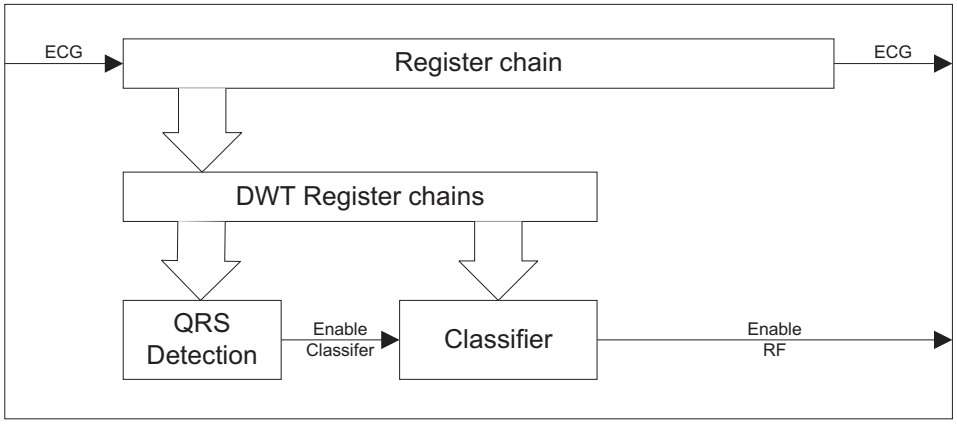

Figure 3.10: Top level block diagram of the chip

The clock gating technique allows reducing the dynamic power by disabling the clock for the structures that are not in use. This reduces the activity factor (α) for that part of the circuit. In the current design, clock gating is used for the classifier, since this is only needed when a successful QRS detection is made. A successful QRS detection is used as an enable signal for the classifier as shown in Figure 3.10, which will load the new features and compute the result of the classification. When in standby, the classifiers inputs are held at the last known value and thus no switching will occur inside the classifier.

The resource sharing technique is a technique to use the same hardware and performing different computations in different cycles instead of multiplying the hardware and executing everything in parallel. In the current design, if the ANN were to be designed to compute the result in one cycle, 43^2 multipliers would be required. Due to the fact that speed is not as important as power in the current design, only 43 multipliers were shared at the cost of computing the classification result in 43 cycles as described in more detail in section 3.2.5.

The multiplication with constants is a circuit level technique to reduce both dynamic and static power. The method consists in replacing multiplications by shifts and adds whenever possible. This approach leads to less hardware. This reduces the dynamic power because the switching capacitance is less and also the static power simply due to the need

for fewer devices, and thus fewer sources for leakage. Due to the values of the coefficients of the filters used for computing the DWT transforms, the multiplications with these coefficients can be successfully replaced with shift and addition operations. For example, the coefficient $\frac{3}{8}$ can be written as shown in equation (3.8). The design for multiplications with $\frac{3}{8}$ is shown in Figure 3.11.

$$\frac{3}{8} = \frac{1}{8} + \frac{1}{4} = 2^{-3} + 2^{-2} \tag{3.8}$$

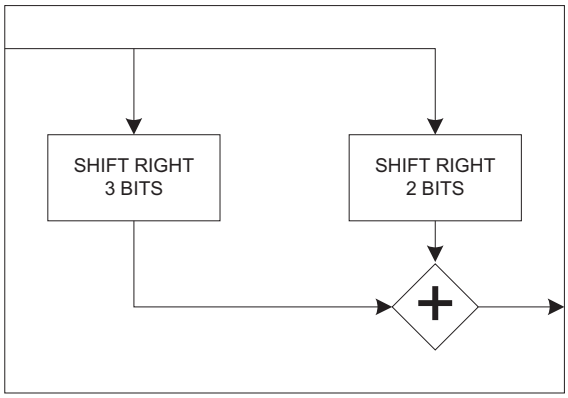

Figure 3.11: Design of the multiplication with $\frac{3}{8}$ based on the representation in equation (3.8)

Following this technique, the representation for all the coefficients needed is shown in Table 3.2, which can be designed without the use of multipliers. The symbols $<<$ and $>>$ are used to denote shifting to the left and right by the number of bits immediately following the symbols, respectively.

Operation needed	Design to replace multipliers
$x * \frac{1}{8}$	$x >> 3$
$x * \frac{3}{8}$	$x >> 3 + x >> 2$
$x * 2$	$x << 1$
$x * (-2)$	$(not(x) + 1) << 1$

Table 3.2: Replacing the multiplication operations

Even with these techniques, the main technique to reduce the dynamic power consumption ($P_{switching}$) is to design the chip to work at a very low frequency f_{clk}. This frequency

was chosen to be 360 Hz, which is also the sampling frequency of the ECG signal. This would allow for the entire chip to work on the same clock as the A to D converter that samples the ECG signal. For comparison, the DSP chip used to implement the QRS detection in [14] (SPROC-1400) operates at a frequency of 50 MHz. If the current chip would operate at such a frequency, the dynamic power consumed would increase significantly due to the fact that dynamic power is directly dependent on frequency of operation of the circuit, as shown in equation (3.7).

The static power reduction techniques are mostly circuit level techniques, and none were applied due to the lack of special libraries needed for these techniques. One method that will certainly reduce the static power consumption is to use high threshold devices across the entire chip. The introduction of high threshold devices slows down the circuit at the cost of reduced static power consumption, but due to the extremely low frequency of the chip (360 Hz), speed is not a concern.

Number Representation

The first decision to be made about the number representation is between the fixed-point and the floating point representation. Figure 3.12 shows the main components of the two representation. One advantage of fixed-point arithmetic is that it requires significantly less hardware than the floating-point arithmetic [26]. This advantage reduces the area and thus the power consumption of the chip, making the fixed-point representation a better candidate for the current work.

Due to the extra bits used to represent the mantissa, the fixed point representation can also be capable of higher accuracies. This advantage is useful if all the numbers are represented using the same common magnitude, or they can be normalized and scaled to meet this criteria. However the accuracy advantage is lost quickly if numbers of different magnitudes have to be manipulated [26]. This means a common magnitude in which all the values can be represented needs to be found.

The ECG samples coming inside the chip are voltage levels with values between -5 and

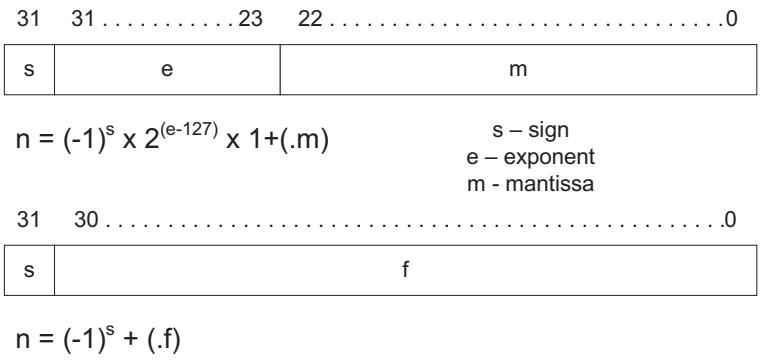

Figure 3.12: Floating point (top) and fixed point (bottom) representations

5 Volts. These values get processed to obtain the DWT coefficients. All the DWT values for all the scales that were obtained using the MIT-BIH database are values that can be fit into the [-5,5] interval as well. The only large value used in the processing is the $RRinterval$ feature, which counts the number of samples between two consecutive R waves. This value is located in the [100,500] interval, which corresponds to a length of 0.36 to 1.5 seconds. In order to accommodate all the numbers in a smaller interval, the decision was made to use $RRinterval/256$ as the feature, instead of the number of samples, $RRinterval$. The 256 value was chosen because it is a power of two and thus can be implemented in hardware as a simple shift to the left by 8 bits, and it reduces the $RRinterval$ value to something that can fit into the [-5,5] interval like the rest of the values in the chip.

The final representation of the numbers was chosen as shown in figure 3.13, where 1 bit is used for the sign, 5 bits for the integer part to match the interval [-31:31], and 18 bits for the fractional part. Despite the fact that the MIT-BIT database only uses an accuracy of 12 bits per sample, this representation was chosen to have a very good accuracy in computations and to make sure that no value used in the processing is outside the interval that can be represented.

Due to multiplication operations inside the classifier the number representation in some parts of the classifier will change to the representation seen in Figure 3.14, in order to accommodate the double size results of the multiplications. However, this representation will not be visible from anywhere outside the chip. The only representation visible from

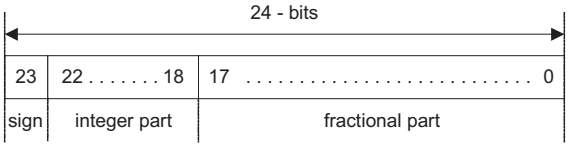

Figure 3.13: Chosen number representation: 24-bit fixed point

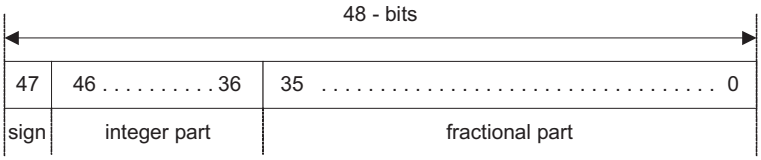

Figure 3.14: Alternative representation: 48-bit fixed point

outside the chip is the one presented in Figure 3.13, which is the format in which the ECG signal will enter and exit the chip. The other inputs and outputs are single bit signals like result of the classification, clock, reset, etc.

With these considerations in mind, each of the algorithms chosen in the previous chapter is implemented in hardware as described in the following sections.

3.2.2 Computing the DWT

The computation of the DWT is equivalent with the design of a series of FIR filters already described in section 3.1.1. Table 3.3 shows the filters that are required for each DWT scale needed, while Table 3.4 shows the coefficients of the filters present in Table 3.3.

DWT scale	Filters
2^1	G
2^2	$H\ G_1$
2^3	$H\ H_1\ G_2$
2^4	$H\ H_1\ H_2\ G_3$

Table 3.3: Filters needed for each DWT scale

The computation hardware for each of the DWT scales needed can be designed using tables 3.3 and 3.4. Figure 3.15 shows the implementation of the second scale of the DWT. The top signal is the ECG signal passing through the pipelined registers. The second row

Filter	Coefficients
H	$\frac{1}{8}\ \frac{3}{8}\ \frac{3}{8}\ \frac{1}{8}$
H_1	$\frac{1}{8}\ 0\ \frac{3}{8}\ 0\ \frac{3}{8}\ 0\ \frac{1}{8}$
H_2	$\frac{1}{8}\ 0\ 0\ 0\ \frac{3}{8}\ 0\ 0\ 0\ \frac{3}{8}\ 0\ 0\ 0\ \frac{1}{8}$
G	$2\quad -2$
G_1	$2\quad 0\quad -2$
G_2	$2\quad 0\quad 0\quad 0\quad -2$
G_3	$2\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ -2$

Table 3.4: Coefficients of the required filters

is the implementation of the H filter with its corresponding coefficients. Each of the multiplications with a coefficient, symbolized here by a block with the coefficient value inside, is realized using only shift and add operations as already described in section 3.2.1. After the first filter, the resulting signal is pipelined again and then the second filter is applied, in this case G_1. The zero coefficient is simply designed by not connecting the corresponding sample. The result is the second scale of the DWT of the ECG signal, which is pipelined for further processing.

The fact that the higher DWT scales use intermediate results from the computations of the lower scales leads to the idea of sharing parts of the hardware when computing all the scales, in order to reduce the total amount of hardware needed. For example, the third DWT scale passes the signal through H, H_1 and G_2, while the third scale can take the intermediate signal from the output of H_1 and pass it through H_2 and G_3 instead of replicating the first two filters. The design of the computation of all the first 4 scales of the DWT is shown in Figure 3.16. The inset shows the output filters hardware that need to be connected on each row to obtain the DWT at each scale. The two shaded registers on each row correspond to the two registers in the inset.

The outputs from the G_i filters shown in the inset are the DWT coefficients, which are pipelined to be used further for QRS detection and beat classification.

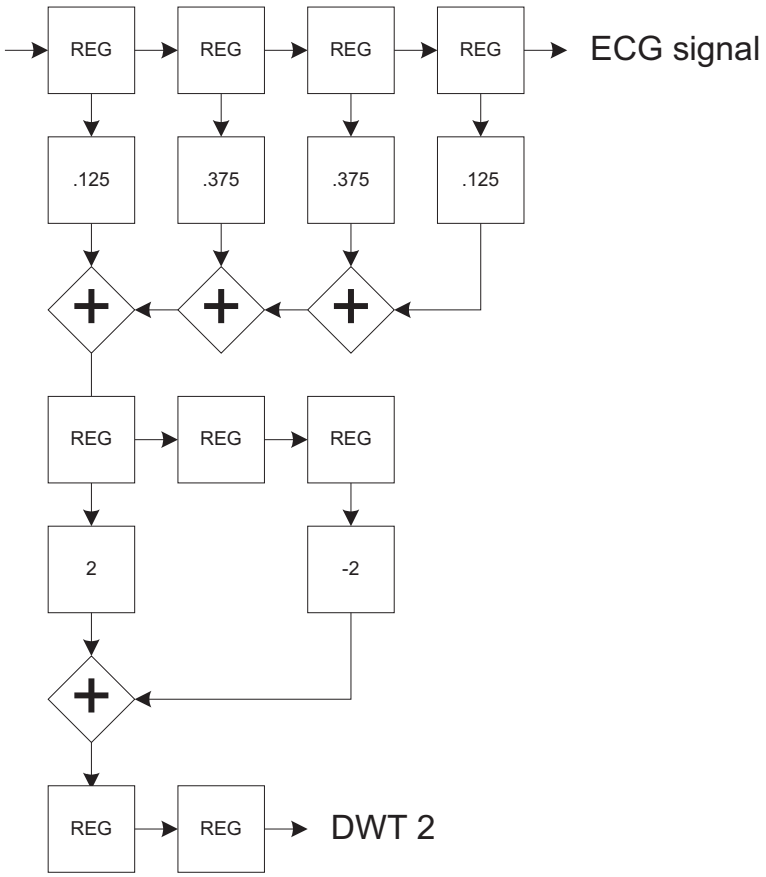

Figure 3.15: Design of the scale 2^2 of the DWT

3.2.3 QRS Detection

The QRS detection algorithm needed to be implemented in hardware consists of detecting matching zero crossings that are flanked by opposite sign peaks across the first 4 DWT scales of the signals, as described in section 3.1.2.

The first step is to detect the zero crossings at each scale. Figure 3.17 shows the circuitry that needs to be added to each of the DWT scales pipelines. The figure shows two comparators in the middle that are used to localize a zero crossing and 2 comparators on the sides used to check for peaks. The symmetric hardware on top and bottom of Figure 3.17 accounts for the two cases: 1)positive peak followed by negative peak or 2)negative peak followed by positive peak. The check for peaks is shown in Figure 3.17 to be 2 samples

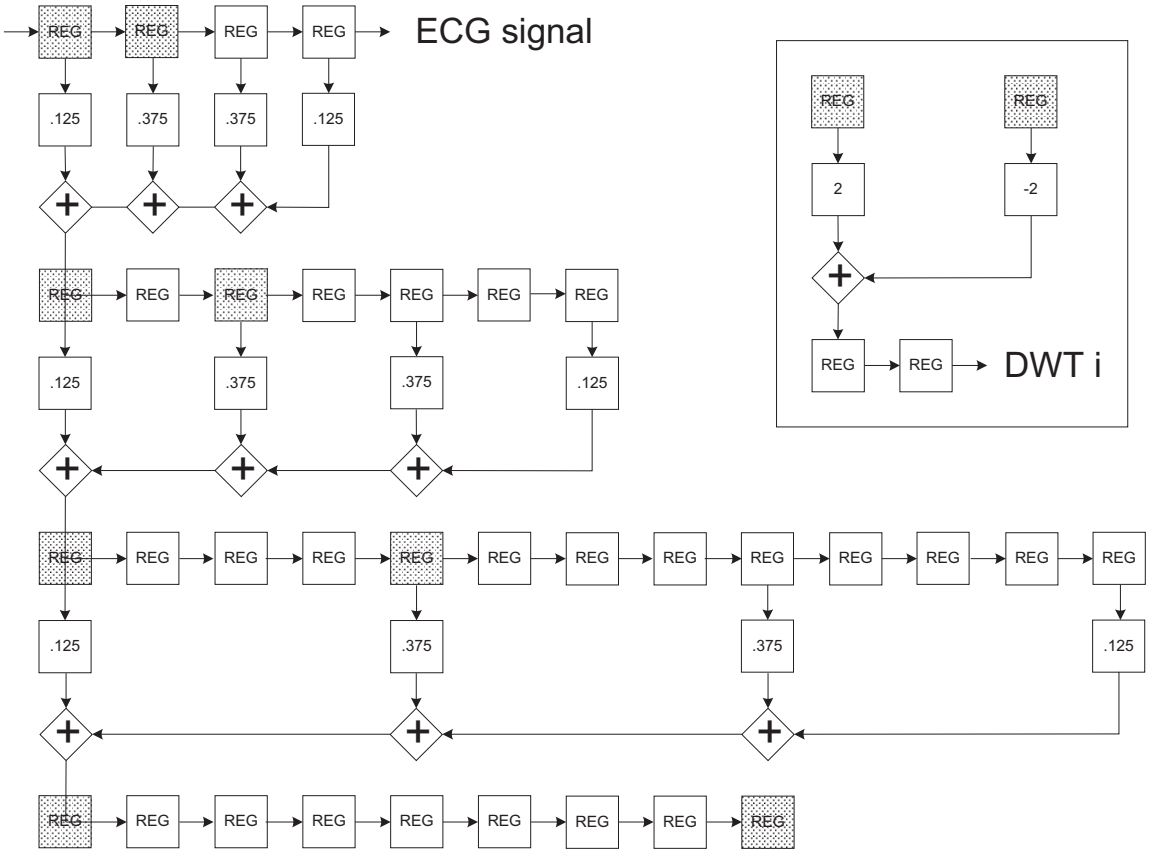

Figure 3.16: Combined design for the first 4 scales of the DWT

away from the zero crossing, but the distance is different with every scale. This distance also needs to include a few neighbors in the check in order to account for asymmetric QRS shapes. If either the hardware above the DWT chain or the one below evaluates as true, a zero crossing is assumed to be detected and pipelined for further processing.

After the 4 zero crossings pipelines are generated, the hardware in Figure 3.18 is used to detect the R waves. The delays introduced by the hardware designed to perform the DWT computations and the zero crossings detection has to be taken into account in each scale, so that the connection can be made accordingly. Table 3.5 shows the delays needed for each of the DWT scales with respect to the ECG signal. The first column shows the delay introduced by the theoretical computation of the DWT for a symmetric R wave as calculated in [5], the second column the added delay of the particular design used in the

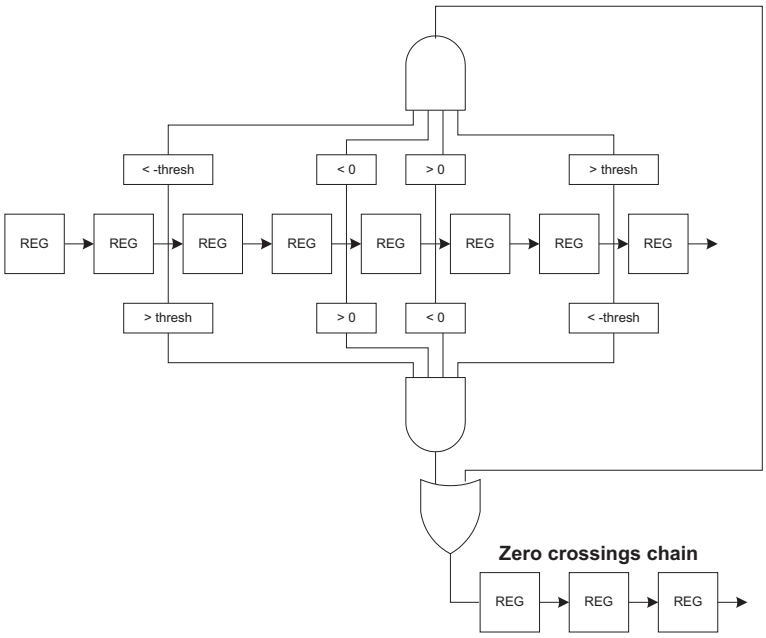

Figure 3.17: Design of obtaining the zero crossing at each DWT scale

work, the third column shows the delay accuracy, while the last column shows the total delay and accuracy.

DWT scale	Theoretical delay	Delay added by design	Delay accuracy	Total
2^1	1	2	± 5	3 ± 5
2^2	3	3	± 5	6 ± 5
2^3	7	4	± 5	11 ± 5
2^4	15	5	± 0	20

Table 3.5: Delays for each DWT scale in samples

In order to implement the delay in the last column, the connection is made to the zero crossings pipelines as many stages later with respect to the ECG point analyzed as the delay indicates. This delay is represented in Figure 3.18 by the long vertical dashed lines. The accuracy is needed because if the wave is not symmetric, the actual delay will be different from the delay present here, and thus this error needs to be taken into account in order to detect non symmetric R waves. The values for the accuracy were chosen empirically. It was decided to fix the detection on the fourth scale of the DWT because it is less affected by noise and set a delay a margin error of ± 5 on the other scales. Figure 3.18 shows how

the ± 5 error margin is taken into account by making connections to all the neighboring 5 stages before and after the zero crossing. Note that only 4 of these neighbors are shown at each scale, the rest are hidden by the short vertical dashed lines.

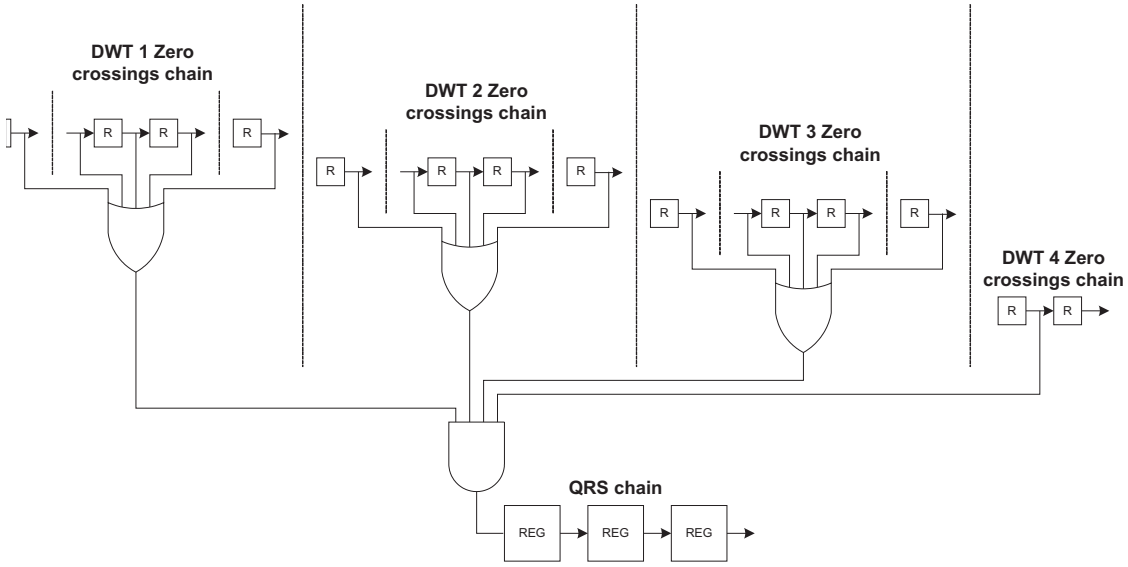

Figure 3.18: Design of the QRS detection from the zero crossings across the scales

If the result of the logic in Figure 3.18 evaluates as true, the QRS is assumed to be 20 samples before the zero crossing detected on the 4th scale of the DWT. The detection is then pipelined and will be used to trigger the classification of the beat when the entire beat is inside the chip.

3.2.4 Design of Control Circuitry for Missed Beats

The proposed chip needs to ensure that if no QRS complex is detected over a certain period of time the transmission is enabled, because something abnormal might be happening, and by definition all abnormal activity needs transmitted to the base station. This can be ensured by measuring the time since the last R wave detection. If a wave is not detected after the maximum RR interval length allowed, the RF will be turned on so that the signal can be transmitted. Figure 3.19 shows the implementation of this part of the chip.

The desired behavior in the case of abnormal beats is to ensure their transmission.

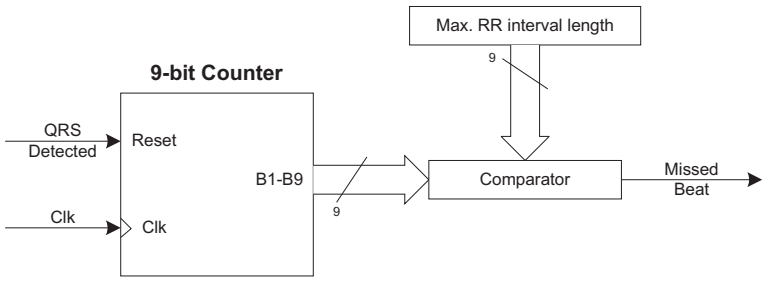

Figure 3.19: Logic to ensure transmission of missed beats

Assuming the the QRS detection algorithm fails to detect the R peak of an abnormal beat, the beat will still be sent due to the missed beat logic, which is also the desired behavior in this case. This means that the QRS algorithm can be relaxed because the accuracy of detecting the QRS in the case of abnormal beats has no importance for the current work due to the missed beat detection circuitry.

3.2.5 Beat Classification

The classifier chosen is a neural network with 43 input nodes, 1 hidden layer composed of 43 nodes and one output nodes, as described in section 3.1.3. The training of the classifier is performed off line, using MATLAB, and the computed weights are loaded into the designed circuit, which has no self training capabilities. From the digital perspective, the classifier is thus composed of the feed forward network, the memory to store the weights, and the control logic for the two units.

The main problems associated with the digital VLSI implementation of a Neural Network are the non linearity of the activation function, the large amount of memory needed for the weights and the large number of interconnects and multipliers in the case of large networks [27].

Digital Design of the Activation Function

One of the main problems in hardware implementation of a neural network is the non linearity of the activation function. This function is usually a sigmoid. One of the most common

sigmoid functions, which is also used in the current work, is defined in equation (3.9). The most common approaches for hardware implementation of the activation function are approximation functions that are more suitable for digital implementation.

$$f(x) = \frac{1}{1 + e^{-x}} \tag{3.9}$$

The most straight-forward approach is the use of a look-up table. The problems with using the look-up table are its size, which is directly proportional to the accuracy of the function approximation, and the fact that it is not feasible to have a look-up table at each node that uses the activation function in the case of large networks. The problem with sharing one common large look-up table is that the parallelism advantage of a hardware neural network is lost.

A better alternative from a hardware perspective is the approximation of the sigmoid by a piecewise linear function. The most common approximation from this category is the one proposed by Alippi and Storti-Gajani in [27], which is also the approach chosen in this work. The idea for this approximation is to split the Y domain into an integer number of intervals, and define a linear function over each interval. The attractiveness of this proposed approach is that the entire function can be designed using only shifters and a counter. This was made possible by selecting the interval limits as powers of two (e.g. 0.5, 0.25). The function is defined in equation (3.10), where $\hat{x}$ represents the signed fractional part of x, and (x) represents the integer part of x.

$$f_{ap}(x) = \frac{\hat{x} + 2}{2^{|(x)|+2}} = \frac{\frac{\hat{x}}{4} + \frac{1}{2}}{2^{|(x)|}} \tag{3.10}$$

This function is a very good approximation of the sigmoid defined in equation (3.9). This can be seen in Figure 3.20, where the top figure shows the sigmoid defined in equation (3.9) with a dotted line overlapping almost perfectly the approximation proposed in [27], represented by a solid line. The bottom figure shows the approximation error between the two functions, error that is always less than 2%.

In order to increase hardware classification accuracy, the function defined in equation

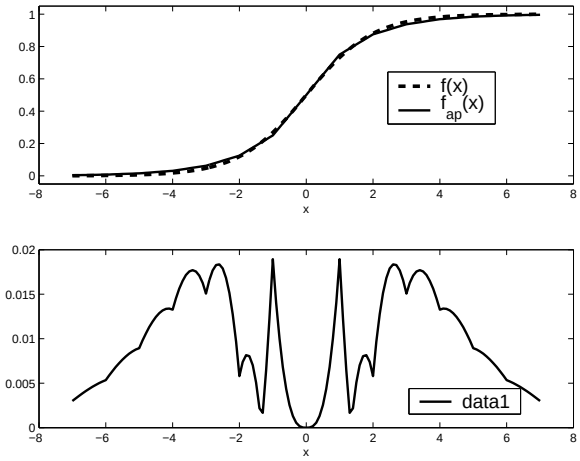

Figure 3.20: Comparison between the sigmoid and its approximation function (top) and the approximation error (bottom)

equation (3.10) is used in the backpropagation training of the neural network, replacing the default functions defined in the MATLAB neural network toolbox. This will ensure that the hardware implementation will work exactly like the MATLAB model. Figure 3.21 shows the hardware design of the function proposed in [27].

Digital Design of the Feed Forward Network

The advantage of a hardware neural network is its speed. This is due to the fact that it can execute all the multiplications needed for each hidden layer in parallel. However, the cost of this speed is a large number of interconnects and multipliers required. For example, for the current work, the straight forward implementation of the feed forward network would require 43^2 24-bit multipliers, which is the number of connections from the input layer to the hidden layer.

The proposed design is shown in Figure 3.22. The main difference is that only 43 multipliers are used to compute all the values leading to one hidden layer node each cycle and keep accumulating the results as shown in Figure 3.22. This means that at least 43 cycles are needed to compute the result of one classification. The speed advantage of the neural network was thus traded for less power consumption due to significantly less

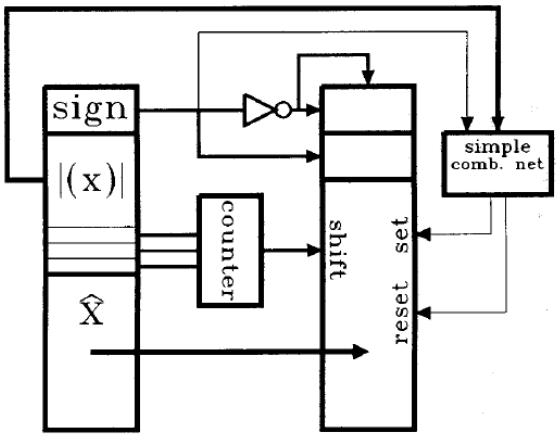

Figure 3.21: Possible hardware implementation of the activation function

hardware, which is the one of the main goals in the current work.

The 43 multipliers receive the 43 features as one of the inputs and each cycle the 43 weights corresponding to a hidden layer node as the second input. The results are added and passed through the activation function. This result is multiplied by the weight corresponding to the connection for the current node to the output and then accumulated into the result register.

The offset for the output layer also needs to be added to the accumulated result. This is done by setting all the weights in the 44^{th} cycle to zero, except for the last one. This one needs to be set to twice the value of the actual offset value because setting all the features to zero will return 0.5 at the output of the activation function, instead of a 1.

This means the memory has to be designed in such way that it will supply 44 weights each clock cycle for 44 cycles. The memory design is presented in Figure 3.23, memory that has 44 lines and 44 rows. Each weight is a 24-bit number, so the total memory needed to store the weights is 46464 bits.

The control logic needed for both the memory and the feed forward network is shown in Figure 3.24. The main clock is gated in order to save power as already described in section 3.2.1. The gated clock stops the memory from changing its output. The feed forward network is stopped from switching because the feature registers at the top of Figure 3.22

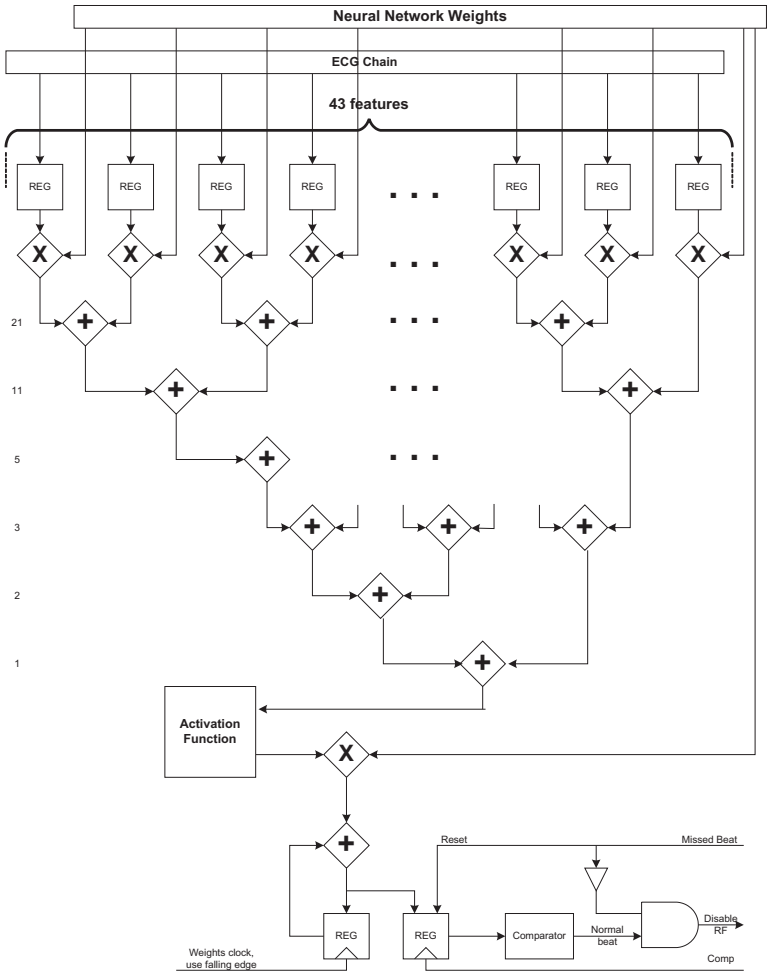

Figure 3.22: ANN Design

are loaded only on a successful QRS detection. The entire classifier is only active for 44 cycles after a successful QRS detection, after which no switching occurs in the classifier until the next QRS is detected.

3.2.6 Overall Delay Analysis

The proposed chip will add an extra delay to the transmission of the ECG signal. The chip analyzes the ECG signal beat by beat and no more than one beat needs to be available to the chip for processing. This allows for a delay of roughly one beat length. The delay is split into its main components in Figure 3.25.

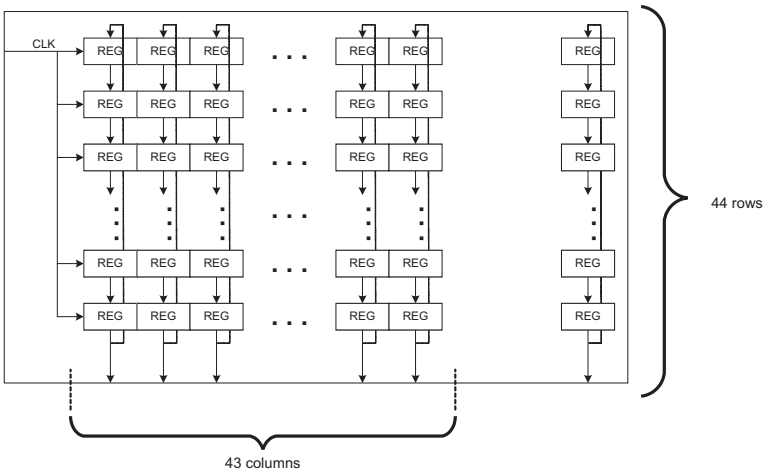

Figure 3.23: Memory design

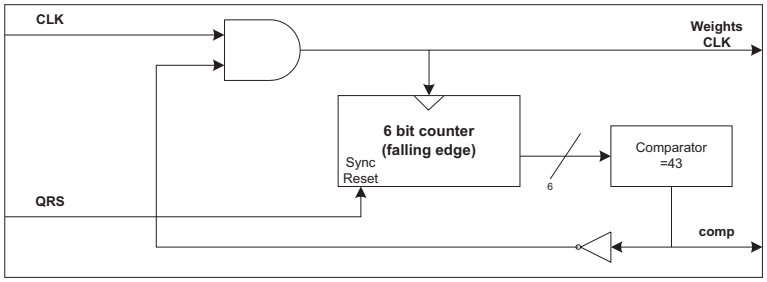

Figure 3.24: Control logic for the ANN

The main component of the delay is generated by the need to have a beat entirely inside the chip for analysis. As discussed in section 3.1.3, a beat is considered to be composed of the samples starting 300 ms before the R wave and 400 ms after the R wave. This means the main component of the delay is the 700 ms long, from T_0 to T_3. Once the entire beat is inside the chip, the classifier is turned on, which requires 44 cycles to compute the result as shown in section 3.2.5. An extra cycle is needed for the comparison of the result to check whether the beat is normal or not. Because the chip's clock frequency is 360 Hz, the 45 cycles translate into a delay of 125 ms, shown from T_3 to T_4. This adds up to a total delay of 825 ms introduced by the proposed chip.

If the design is left as it is, the missed beat circuitry will have an upper limit on maximum RR interval length given by the current design. This upper limit is given by the

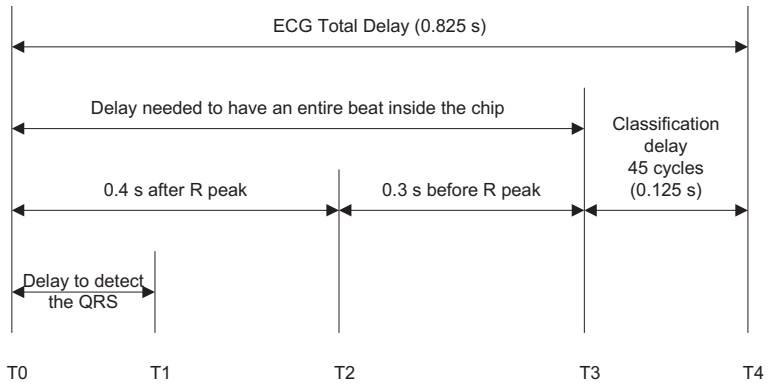

Figure 3.25: Total delay introduced by the chip

amount of data being buffered inside the chip. The count starts at T_1, when the last QRS was detected. Assuming the next beat will not be detected, its transmission has to begin when the missed beat is still inside the chip to avoid any possibly abnormal data loss. This maximum delay is equal to the time the previous beat needs to fully enter the chip after its QRS was detected plus the time it needs to exit the chip, which is composed by the time it needs to be classified and it's length so it can be sent out entirely.

$$RR_{max} = T_2 - T_1 + T_4 - T_3 + T_3 - T_0 = 275 + 125 + 700 = 1100(ms) \tag{3.11}$$

The value means that in the current design, any maximum RR length can be considered as long as it is less than 1100. If a larger value needs to be used, the ECG signal needs to be buffered further in order to avoid data loss. The extra delay introduced by the extra buffering would also add to the 825 ms delay of the chip. However, it is safe to assume that if no QRS was detected in 1100 ms, the next beat was either not detected or abnormal due to the extremely long RR-interval and the transmission should happen naturally.

3.3 VHDL Model Testing and Analysis

The MATLAB model was used as a reference for testing the VHDL model and ensure the functionality of the two models is identical. There are however differences between the two

models. The most important difference is that the VHDL model uses a 24 bit fixed point representation as described in section 3.2.1, while MATLAB uses double precision for all the computations, which is a floating point 64 bit representation. Section 3.3.1 presents how the VHDL model was tested, while the second section, 3.3.2, presents an analysis of the impact the accuracy has on the VHDL model.

3.3.1 Testing the VHDL Model

The VHDL model was tested by selecting random sequences from the ECG signals present in the MIT-BIH database and comparing the classification results to the classification results to the results obtained from running the same sequence through the MATLAB model. An example of a typical extracted ECG signal is shown in Figure 3.26.

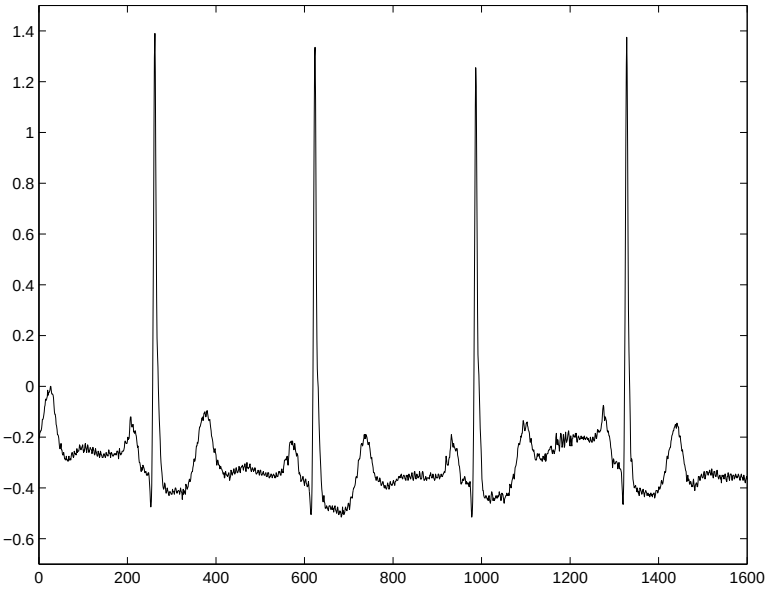

Figure 3.26: Sample of ECG portion used for testing the VHDL model

The waveform resulted from processing the ECG signal shown in Figure 3.26 is shown in Figure 3.27. Some of the functionality of the circuit described in this chapter can be identified in this figure. For example, it can be seen that many zero crossings are detected across the 4 DWT scales, but only 3 of them aligned with the appropriate delays to form a

QRS detection. It can also be seen how the classifier clock is activated only when a QRS is detected and lasts until the classification is completed. The delay seen from the point where the zero crossings were aligned to point of the classification activation is the delay introduced so that the entire beat is inside the chip. When the classification is complete, the wireless transmission is enabled or disabled based on the result of the classification and the previous state.

In the current scenario, the first three beats were successfully detected as normal beats and it can be seen that the wireless transmission is disabled for the duration of these three beats. In the case of the last beat, the QRS was missed, no classification was performed, but the missed beat circuitry described in section 3.2.4 detected this and enabled the transmission.

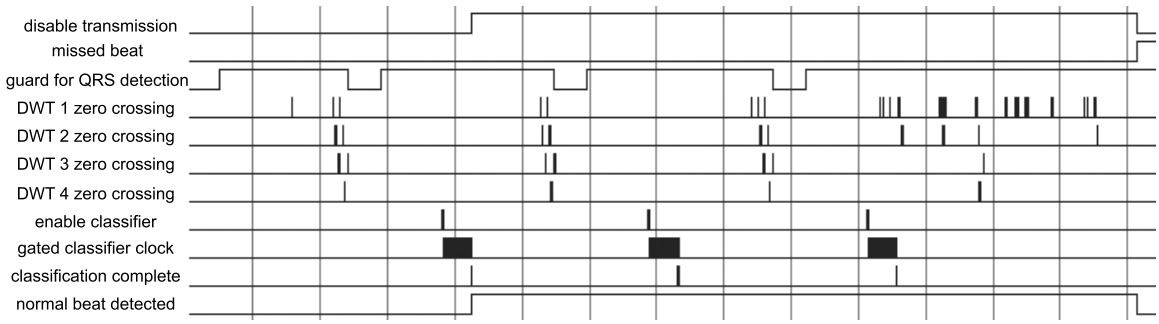

Figure 3.27: Waveform from simulating the VHDL model using the ECG from Figure 3.26

All the tests ran matched perfectly the MATLAB functionality, although accuracy differences exist, which are analyzed in more detail in the next section.

3.3.2 Accuracy Error Analysis

The accuracy of the VHDL model is reduced due to the fixed point representation chosen in section 3.2.1. The same example used section 3.3.1 is used here to show the accuracy difference. The first step is to compare were the features extracted for the classification between the two models. Table 3.6 shows the statistical results for all the $3x43$ features

extracted for the 3 beats detected in the example used in 3.26, which show that no more than a 1% error is present in any of the features extracted.

Min error	0
Max error	$2.73 * 10^{-5}$
Max relative error	0.903%

Table 3.6: Statistics for the features errors

The next step is to compare the exact value of classification results, not just the binary outcome of the classification. Table 3.7 shows a comparison of the exact classification result error between the MATLAB and the VHDL model. As the table shows, the differences are very small, and the average relative error is even smaller than the average relative error recorded at the input of the classifier. This shows that the MATLAB and the VHDL models are almost identical. The error shown in Table 3.7 is so small that a different outcome from the two models over the same beat is almost impossible. The outcome threshold for this work is set at 1.4, which means the only case where a beat is classified differently is when it has a value larger than 1.4 in one model and smaller than 1.4 on the second.

	beat 1	beat 2	beat 3
MATLAB result	1.308458	1.154395	1.326718
VHDL result	1.308404	1.154373	1.326710
Relative error	0.004146	0.001910	0.000577

Table 3.7: Classification results for the 3 identified beats in Figure 3.26

Overall, 10 randomly chosen sequences of beats like the one shown in 3.27 were tested on the two models. Out of the 39 beats, 2 were not detected, 31 were detected as normal beats, and 6 detected as abnormal beats. The classification behavior into normal and abnormal was identical on both models in all cases.

3.4 VHDL Synthesis

After building and verifying the synthesizable VHDL model, the next step is to synthesize it in order to obtain power and delay estimations. The tool used for this step is Design_compiler from Synopsis. The model was synthesized without using any restrictions on area or power. The only thing supplied to the tool besides the VHDL model was the desired clock frequency of 360Hz, in order to obtain a more accurate power estimation. The cell library used for the synthesis was *cb13fs120_tsmc*, which was the only library available for this project. While this restricts the final design, it is sufficient to obtain power and delay estimates. If a more accurate estimates is desired, more accurate tools could be used after the synthesis step. For example, NanoSim could be used for power estimations and Pathmill for critical path delay.

Chapter 4

Results and Analysis

This chapter presents and analyzes the results in order to assess the performance of the proposed design, and how well it meets its goals. The first two sections show the performance of the chosen algorithms for QRS detection and beat classification, respectively. The third section shows the results obtained from combining the algorithms, which are the results that show the performance of the entire design. The results in these three sections were gathered from the MATLAB model, which was shown to be almost identical to the VHDL model in section 3.3.2. This was because of the large size of the MIT-BIH database and the impossibility to simulate the entire database through VHDL. The last section, 4.4, presents the results of the synthesis of the chip.

4.1 QRS Detection

The QRS algorithm described in section 3.1.2 was implemented both in MATLAB and VHDL. The thresholds and other variables were obtained empirically, by selecting the values that provide the best results when running the algorithm over the MIT-BIH database. Due to the circuitry implemented to account for missed beats as described in section 3.2.4, the accuracy of the detection of the abnormal beats can be ignored.

The best results obtained for QRS detection are shown in detail in Table 4.1. The table shows the results over each individual file on which the test was ran on. The first column shows the file number corresponding to the MIT-BIH record, the second column shows the

number of normal beats for which the QRS was successfully detected, the third column shows the number of abnormal beats for which the QRS was successfully detected, the fourth column shows the number of normal beats for which the QRS could not be detected, the fifth column shows the number of abnormal beats for which the QRS could not be detected, and the last column shows the percentage of normal beats in the file.

The false beats are important to remove because in the event the classifier fails as well and classifies the false beat as a normal beat, the transmitter could be turned off, and potentially useful data could thus be lost. This was the trade-off that had to be made, because lower thresholds for example would increase the detection accuracy of the normal beats, but also increase the detection of false beats, which is usually noise that resembles a QRS complex. One such example of detecting false beats is shown in Figure 4.1, where a noisy portion of the ECG signal contains multiple QRS like shapes. The example is taken from file 105, which is shown in Table 4.1 to have the highest number of false beats detected.

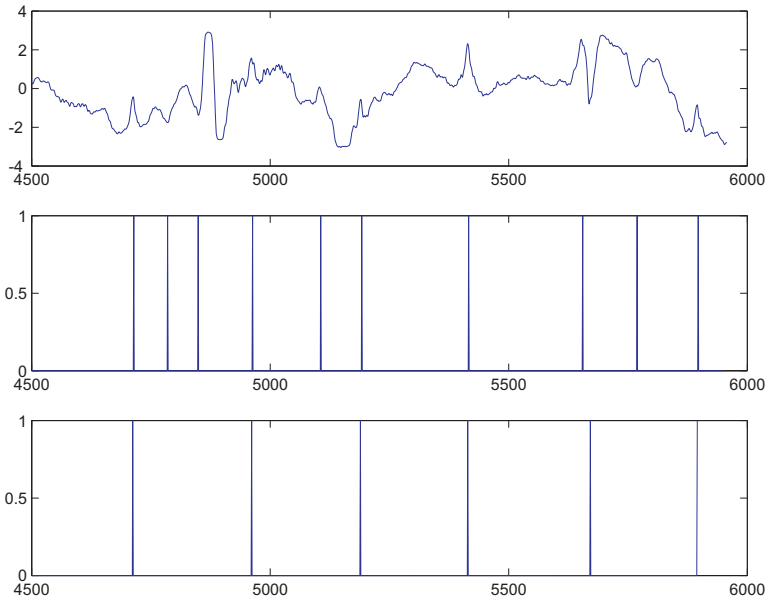

Figure 4.1: Portion of ECG signal from file 105 (top) with the algorithm QRS detections (middle) and the database annotations locations (bottom)

Due to the lack of computational power and the large size of each individual file, the algorithm was ran on smaller non overlapping segments extracted from each file. Added to

File Number	Detected Normal	Detected Abnormal	Missed Normal	Missed Abnormal	False Beats	Normal Beats (%)
100	2213	33	0	0	0	98.53
101	1834	4	1	1	1	99.67
103	2056	2	1	0	0	99.90
105	2471	45	26	1	71	95.52
106	1484	454	0	64	1	74.09
107	0	2088	0	24	0	0.00
108	797	21	922	2	0	98.68
109	0	2491	0	11	0	0.00
111	0	917	0	1182	0	0.00
112	2500	2	6	0	0	99.92
113	1768	6	0	0	1	99.61
114	1806	58	0	1	1	96.78
115	1929	0	0	0	0	100.00
116	2256	106	20	2	0	95.47
117	1511	1	4	0	0	99.93
118	0	2250	0	1	1	0.00
119	1523	437	0	3	6	77.35
121	1787	2	52	0	0	99.89
122	2447	0	0	0	0	100.00
123	1494	2	0	0	0	99.87
124	0	1586	0	14	2	0.00
200	1717	830	1	21	3	66.80
202	2041	50	0	24	0	96.50
203	2394	388	106	57	63	83.11
205	2534	73	4	12	0	96.76
207	0	1668	0	178	17	0.00
208	1556	599	12	749	12	53.55
209	2586	380	0	1	1	87.13
210	2376	92	20	132	3	91.35
212	917	1799	0	0	1	33.75
213	2610	551	0	50	6	81.13
214	0	2222	0	14	0	0.00
215	3151	159	6	7	1	94.98
217	242	1890	0	51	0	11.09
219	2059	55	0	15	0	96.71
220	1932	92	0	0	0	95.45
221	2006	362	0	30	0	83.65
222	1061	309	975	111	0	82.90
223	1994	559	8	14	0	77.75
228	1544	292	120	69	2	82.09
230	2224	1	0	0	0	99.96
231	313	1237	0	0	0	20.19
233	2199	837	0	4	1	72.31
234	2667	50	0	3	1	98.02
Total	69999	25000	2284	2848	195	

Table 4.1: QRS Detection Results

this, the software was written in such manner that a portion in the beginning and in the end of each file is not processed. This is why the total number of beats reported in the results section for each individual file is slightly smaller than the total number of beats presented in the file. The total number of beats missed due to these shortcomings is under 2.5% of the total number of beats in the database.

A summary of the results in Table 4.1 is presented in Table 4.2. The accuracies were computed using the formula show in equation (4.1), except for the false beats accuracy, which is the percentage represented by the false beats out of the total number of beats.

$$accuracy = \frac{positive_{detection} * 100}{positive_{detection} + missed_{detection} + false_{detection}} \tag{4.1}$$

Total Accuracy	94.87 %
Normal Beat Accuracy	96.84 %
Abnormal Beat Accuracy	89.77 %
False Beats Accuracy	0.19 %

Table 4.2: QRS Detection Summary

4.2 Beat Classification

The classification was performed in MATLAB, using the Neural Network Toolbox. The activation function described in section 3.2.5 was used for the training to match the hardware implementation. The training set used contained 630 beats. Files 100, 105, 106, 122, 200, 205, and 233 were chosen to gather normal beats; files 111, 118, 124, 212, and 214 were chosen to gather left and right bundle branch block beats; files 119, 203, 208, 213, and 215 were chosen to gather PVC beats; files 107 and 217 were chosen to gather paced beat; file 209 was chosen to gather APC beats. The beats for the training set were chosen randomly from the 21 files mentioned out of the total 44 files being processed.

The threshold for deciding if a beat is normal or not was determined empirically in order to minimize the number abnormal beats being misclassified, while keeping a high

File Number	Correct Normal	Misclassified Normal	Correct Abnormal	Misclassified Abnormal	Had beats in Training Set
100	2211	2	3	30	yes
101	1823	11	2	3	no
103	2040	16	0	2	no
105	1963	508	108	8	yes
106	1435	49	451	4	yes
107	0	0	1988	100	yes
108	121	676	17	4	no
109	0	0	2431	60	no
111	0	0	884	33	yes
112	2487	13	0	2	no
113	1742	26	7	0	no
114	1185	621	55	4	no
115	1854	75	0	0	no
116	1991	265	102	4	no
117	18	1493	1	0	no
118	0	0	2251	0	yes
119	1522	1	440	3	yes
121	1727	60	1	1	no
122	2445	2	0	0	yes
123	1424	70	2	0	no
124	0	0	1577	11	yes
200	1681	36	782	51	yes
202	1578	463	48	2	no
203	1490	904	403	48	yes
205	2520	14	63	10	yes
207	0	0	1669	16	no
208	1249	307	470	141	yes
209	2549	37	301	80	yes
210	2199	177	83	12	no
212	915	2	1799	1	yes
213	2560	50	416	141	yes
214	0	0	2214	8	yes
215	3051	100	154	6	yes
217	231	11	1879	11	yes
219	1784	275	42	13	no
220	1930	2	58	34	no
221	1992	14	362	0	no
222	416	645	246	63	no
223	1356	638	534	25	no
228	970	574	288	6	no
230	1443	781	1	0	no
231	1	312	1237	0	no
233	2074	125	835	3	yes
234	2666	1	33	18	no
Total	60643	9356	24237	958	

Table 4.3: Classification Results

normal beat classification accuracy. The best results obtained are presented in Table 4.3.

The main goal of the classification is to reduce the number of abnormal beats detected as normal as much as possible, while keeping the number of successfully detected normal beats as high as possible. These two values are present in Table 4.3 in the fifth and the second column respectively. A summary of the results for the classification is presented in Table 4.4. The accuracies were computed using equation (4.1).

Total Accuracy	89.17 %
Normal Beat Accuracy	86.67 %
Abnormal Beat Accuracy	96.20 %

Table 4.4: Classification Summary

The classification results are impressive due to the fact that only 630 beats were used for the training, which means that 99.33% of the beats did not overlap the training beats and more than half the files did not have any beats in the training set. This shows that the classifier performs well when analyzing new data. From the medical perspective however, the abnormal accuracy of 96.20 % is not enough, because in the current design 958 abnormal beats from the MIT-BIH database are treated as normal and thus never sent over the wireless network.

4.3 Combined Algorithm Results

When the algorithms are combined, the missed QRS beats are added to the Table 4.3. The normal beats beats that were missed by the QRS detection get added to the *Normal beats sent* category, while the abnormal beats missed by the QRS detection get added to the *Abnormal beats removed* category. The 195 false beats detected by the QRS detection algorithm are not included because an individual analysis is required to check if the beat is stopping the transmission of abnormal data or transmitting normal data. Even though these beast can only decrease the accuracy by increasing the number in the *Normal beats sent* category or the *Abnormal beats removed* category, their effect is not significant because

File Number	Normal beats removed	Normal beats sent	Abnormal beats sent	Abnormal beats removed
100	2211	2	3	30
101	1823	12	3	3
103	2040	17	0	2
105	1963	534	109	8
106	1435	49	515	4
107	0	0	2012	100
108	121	1598	19	4
109	0	0	2442	60
111	0	0	2066	33
112	2487	19	0	2
113	1742	26	7	0
114	1185	621	56	4
115	1854	75	0	0
116	1991	285	104	4
117	18	1497	1	0
118	0	0	2252	0
119	1522	1	443	3
121	1727	112	1	1
122	2445	2	0	0
123	1424	70	2	0
124	0	0	1591	11
200	1681	37	803	51
202	1578	463	72	2
203	1490	1010	460	48
205	2520	18	75	10
207	0	0	1847	16
208	1249	319	1219	141
209	2549	37	302	80
210	2199	197	215	12
212	915	2	1799	1
213	2560	50	466	141
214	0	0	2228	8
215	3051	106	161	6
217	231	11	1930	11
219	1784	275	57	13
220	1930	2	58	34
221	1992	14	392	0
222	416	1620	357	63
223	1356	646	548	25
228	970	694	357	6
230	1443	781	1	0
231	1	312	1237	0
233	2074	125	839	3
234	2666	1	36	18
Total	60643	11640	27085	958

Table 4.5: Overall design results

they are less than 0.02% of the total number of beats. Table 4.5 shows the overall results of the chip over the database.

According to Table 4.5, 61.4 % of the MIT-BIH data will be saved from transmission, and thus transmit only 38.6 % of what the system would transmit without the proposed chip. The only downside is that 0.95 % of the data is being saved incorrectly, because the abnormal beats should all be transmitted over the wireless connection.

4.4 Chip Synthesis Results

This section presents the chip characteristics as extracted from the synthesized VHDL code. The first results are the power estimation results for the proposed chip, which are presented in Table 4.6. These results show that the power consumption is being dominated by the leakage power. This is explained by the very low clock frequency of 360 Hz. Another important thing to notice is that the classifier accounts for more than half of the total power consumption, despite the large number of registers used to pipeline the ECG signal and its DWT transform.

Circuit	Internal Power (uW)	Net Switching Power (uW)	Total Dynamic Power (uW)	Leakage Power (mW)	Total Power (mW)
Classifier	-	-	0.7915	0.7423	0.7431
Entire Chip	12.2546	16.2985	28.5531	1.2576	1.2862

Table 4.6: Power consumption estimates

Due to the magnitude of the static power consumed compared to the dynamic power consumed, implementation of static power reduction techniques can reduce the chip's power consumption. The most common technique to reduce the static power consumption is the use of high threshold devices. These devices would replace the normal devices and reduce their leakage current at the cost of increasing delay. However, the timing analysis of the circuit shows that there is plenty of room to increase the delay. The critical path of the chip was identified to have a delay of 41 us inside the classifier, when the maximum

delay allowed is 1389 us. This means that the entire chip can be implementing using only high threshold devices, which will decrease the static current consumption, without affecting the functionality due to the large amount of slack available. Unfortunately, no high threshold libraries were available for synthesis to verify how much will the static power be reduced by this technique.

According to [28], the energy consumption corresponds to different phases of the radio transmission and different states of the radio. The power consumption estimation proposed in [28] is greatly simplified in this work. The assumptions for the simplification were that the power in all other states other than the transmission state is zero, the power needed to switch between states is zero, and there is no need for retransmission. The resulting power estimation formula is shown in equation (4.2), where the average power is obtained by multiplying the transmission power by the percentage of time spent in transmission state.

$$AveragePower = TxPower * DataRate/TransmissionDataRate \tag{4.2}$$

where $AveragePower$ is the average power consumed in transmitting the ECG signal as is, the $TransmissionDataRate$ is the transmission rate of the used transmitter, the $TxPower$ is the given transmission power for the transmitter, and the $DataRate$ the data rate of the signal being transmitted. For this work it was assumed that the ECG signal is sampled at 360 Hz, using 12 bits per sample, which are the characteristics of the signals recorded in the MIT-BIH database. This leads to a $DataRate$ of 4.32 kbps.

Based on the power results given in Table 4.6, a power savings analysis is performed in Table 4.7, for several commonly used motes. The values for the Tx power from the 2^{nd} row of Table 4.7 were taken from [22]. These values were normalized for the same transmission power (0dB), they account for the different operation voltages, but it is not specified if they are normalized for the same transmission range. In order to asses the average power needed to continuously transmit the ECG signal, equation (4.2) is used. The results of this computation are shown in the 3^{rd} row of Table 4.7.

The power savings in the 4^{th} row of Table 4.7 are computed as a percentage of the

Row	Wireless type	WeC	Rene	Rene2	Dot	Mica	Mica2Dot	Mica2	Telos
1	Data rate (kbps)	10	10	10	10	40	38.4	38.4	250
2	Tx power (mW)	36	36	36	36	36	42	42	35
3	Avg. power (mW)	15.55	15.55	15.55	15.55	3.89	4.73	4.73	0.61
4	Tx savings (mW)	9.55	9.55	9.55	9.55	2.39	2.90	2.90	0.37
5	Total savings (%)	53.14	53.14	53.14	53.14	28.38	34.12	34.12	-150.01

Table 4.7: Power savings

transmission power from the 3^{rd} row. This percentage is the percentage of normal beats found by the current design over the MIT-BIH database, which is 61.4 % as shown in section 4.2. The 5^{th} row savings are calculate as a percentage of the total savings by adding the estimated power requirements of the proposed circuit to the values from the 4^{th} row. The results show consistent power improvements except for the Telos motes, which due to its higher bandwidth does not see any improvement from the proposed chip, and its power consumption is worsened.

Table 4.8 shows potential savings assuming different percentages of normal beats detected, as well as a case where the power consumption of the chip is reduced to a value of 0.371 mW by implementing several static power reduction techniques. The value of 0.371 mW was chosen as the maximum value for which the Telos motes can see power improvements even at the detection rate given by the current chip over the MIT-BIH database of 61.4 %. The 95 % detection rate was chosen as the normal beat detection rate in the case of a relatively healthy person as it can be seen in the last column of Table 4.1.

Assumptions		Mote types							
Percentage removed	Chip (mW) (mW)	WeC	Rene	Rene2	Dot	Mica	Mica2Dot	Mica2	Telos
61.4	1.286	53.13	53.13	53.13	53.13	28.32	34.18	34.18	-151.23
95	1.286	86.73	86.73	86.73	86.73	61.92	67.78	67.78	-117.63
61.4	0.371	59.01	59.01	59.01	59.01	51.86	53.55	53.55	0.06
95	0.371	92.61	92.61	92.61	92.61	85.46	87.15	87.15	33.66

Table 4.8: Potential power savings

Chapter 5

Conclusion and Future Work

The goal of investigating the improvements given by a pre-classifying chip were accomplished. The chip was shown to be able to remove 61.4 % of the data from the MIT-BIH database. This data can be dropped from transmission, and thus save the same percentage of bandwidth available at a node. The power savings are also significant, except for the Telos motes, where an increase in power consumption was seen using the proposed chip.

The chip will perform differently based on the percentage of normal beats in the patient's ECG heartbeat. The higher this percentage, which is very close to 100 % in the case of patients with no heart problems, the higher the bandwidth and power savings of the design. This means an investigation of the actual savings should be performed in each case where the chip is intended to be used, because its performance is directly affected by the data being monitored.

As directions for future work, there are many possible improvements of the current design:

- The accuracy of the classifier should be increased in the sense that no abnormal data should be dropped from transmission, attempting at the same time to also improve or at least not incur a significant loss in the accuracy of detecting the normal beats. This could be done either by handpicking a better training set for the current classification scheme, or replacing the current classification scheme altogether. Due to the variations in the ECG signal, morphological features could improve the classification

accuracy, but the detection algorithm should be changed as well in order to detect the new features.

- The number representation chosen can be reduced and thus decrease the power consumption. The current implementation uses 20 bit number representation, while the MIT-BIH uses 12 bits per sample. An investigation should be done on how much lower can the number representation be, without incurring a loss in the accuracy of the algorithm.

- The power of the current chip can be further reduced using static power reduction techniques. There is plenty of room to increase the delay due to the low frequency of the chip, which is usually the trade-off in most power savings techniques. The easiest one to implement that would be ideal for the current chip is to use high threshold devices in the entire chip, which will reduce the leakage current and thus the static power consumption.

- The power can be estimated using more accurate tools, such as NanoSim. The dynamic power can be more accurate even using Design_compiler by including a switching profile. This was not necessary in the current work due power consumption being dominated by the static power.

- The power consumption of the wireless transmitter should also be analyzed in more detail, possibly using actual measurements from one of the currently existing ECG monitoring motes.

Bibliography

[1] W. Rosamond, K. Flegal, G. Friday, K. Furie, A. Go, K. Greenlund, N. Hasse, M. Ho, V. Howard, B. Kissela, S. Kittner, D. Lloyd-Jones, M. McDermott, J. Meigs, C. Moy, and G. Nicol. Heart disease and stroke statistics-2007 update: a report from the American Heart Association Statistics Committee and Stroke Statistics Subcommittee. *Circulation*, 2007.

[2] V. Shnayder, B. Chen, K. Lorincz, T. R. F. Fulford-Jones, and M. Welsh. Sensor networks for medical care. Technical report, Harvard University, 2005.

[3] K. Grauer. *A practical guide to ECG interpretation*. Mosby, St. Louis, 2nd edition, 1998.

[4] D. Jenkins and S. Gerred. http://www.ecglibrary.com/ecghome.html, October 2002.

[5] L. Cuiwei, Z. Chongxun, and Y. Chaowei. Detection method of ECG using wavelet transform. *Chinese Journal of Biomedical Engineering*, 14(1):59–66, 1995.

[6] O. T. Inan, L. Giovangrandi, and G. T. A. Kovacs. Robust neural-network-based classification of premature ventricular contractions using wavelet transform and timing interval features. *IEEE Transactions on Biomedical Engineering*, 53(12):2507–2515, 2006.

[7] W. Jiang, S. G. Kong, and G. D. Peterson. ECG signal classification using block-based neural networks. In *International Joint Conference on Neural Networks, IJCNN 2005, Jul 31-Aug 4 2005*, volume 1, pages 326–331, 2005.

[8] R. V. Andreao, B. Dorizzi, and J. Boudy. ECG signal analysis through hidden Markov models. *IEEE Transactions on Biomedical Engineering*, 53(8):1541–1549, 2006.

[9] Z. Qibin and Z. Liqing. ECG feature extraction and classification using wavelet transform and support vector machines. In *2005 International Conference on Neural Networks and Brain Proceedings, ICNNB'05*, volume 2, pages 1089–1092, Oct 2005.

[10] I. Christov, G. Gomez-Herrero, V. Krasteva, I. Jekova, A. Gotchev, and K. Egiazarian. Comparative study of morphological and time-frequency ECG descriptors for heartbeat classification. *Medical Engineering and Physics*, 28(9):876–887, 2006.

[11] I. Daubechies. *Ten lectures on wavelets*. Society for Industrial and Applied Mathematics, Philadelphia,PA, 1992.

[12] D. F. Mix and K. J. Olejniczak. *Elements of wavelets for engineers and scientists*. Wiley-Interscience, Hoboken, NJ, 2003.

[13] J. P. Martinez, R. Almeida, S. Olmos, A. P. Rocha, and P. Laguna. A wavelet-based ECG delineator evaluation on standard databases. *IEEE Transactions on Biomedical Engineering*, 51(4):570–581, 2004.

[14] M. Bahoura, M. Hassani, and M. Hubin. DSP implementation of wavelet transform for real time ECG wave forms detection and heart rate analysis. *Computer methods and programs in biomedicine*, 52(1):35–44, 1997.

[15] L. V. Fausett. *Fundamentals of neural networks :architectures, algorithms, and applications*. Prentice-Hall, Englewood Cliffs, N.J., 1994.

[16] G. B. Moody and R. G. Mark. The MIT-BIH arrhythmia database on CD-ROM and software for use with it. In *Computers in Cardiology - Proceedings, Sep 23-26 1990*, Computers in Cardiology, pages 185–188, Chicago, IL, USA, 1991. MIT, Cambridge, MA, USA, Publ by IEEE, Los Alamitos, CA, USA.

[17] A. Taddei, A. Biagini, G. Distante, M. Emdin, M. G. Mazzei, P. Pisani, N. Roggero, M. Varanini, R. G. Mark, G. B. Moody, L. Braaksma, C. Zeelenberg, and C. Marchesi. The European ST-T database: development, distribution and use. In *Computers in Cardiology Proceedings*, pages 177–80. IEEE Comput. Soc. Press, 23-26 Sept. 1990 1991.

[18] P. Laguna, R. G. Mark, A. Goldberger, and G. B. Moody. A database for evaluation of algorithms for measurement of QT and other waveform intervals in the ECG. *Computers In Cardiology*, 24:673–676, 1997.

[19] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C. . Peng, and H. E. Stanley. Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000.

[20] M. Jiang. Tele-Cardiology Sensor for Remote ECG Monitoring. Master's thesis, RIT, 2006.

[21] B. S. Kim, S. K. Yoo, and M. H. Lee. Wavelet-based low-delay ECG compression algorithm for continuous ECG transmission. *IEEE Transactions on Information Technology in Biomedicine*, 10(1):77–83, 2006.

[22] J. Polastre, R. Szewczyk, and D. Culler. Telos: enabling ultra-low power wireless research. In *4th International Symposium on Information Processing in Sensor Networks*, volume 2005, pages 364–369, Apr 2005.

[23] K. A. Ng and P. K. Chan. A CMOS analog front-end IC for portable EEG/ECG monitoring applications. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 52(11):2335–2347, 2005.

[24] S. Mallat. Zero-crossings of a wavelet transform. *IEEE Transactions on Information Theory*, 37:1019–1033, 1991.

[25] Anantha P. Chandrakasan and Robert W. Brodersen. Minimizing power consumption in digital CMOS circuits. *Proceedings of the IEEE*, 83(4):498 – 523, 1995.

[26] Christopher Inacio and Denise Ombres. DSP decision: fixed point of floating? *IEEE Spectrum*, 33(9):72 – 74, 1996.

[27] C. Alippi and G. Storti-Gajani. Simple approximation of sigmoidal functions: Realistic design of digital neural networks capable of learning. *Proceedings - IEEE International Symposium on Circuits and Systems*, 3:1505 – 1508, 1991.

[28] B. Bougard, F. Catthoor, D. C. Daly, A Chandrakasan, and W. Dehaene. Energy efficiency of the IEEE 802.15.4 standard in dense wireless microsensor networks:modeling and improvement perspectives. *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition*, 2005.