Arab Academy for Science, Technology and Maritime Transport
College of Engineering and Technology, Cairo
Electronics and Communications Engineering Department

# Design and Implementation of a Real-time Sleep Stage Monitoring and Classification System Using Support Vector Machine, for Narcolepsy Diagnosis

By

**Eng. Fadi Sami Kamal Naguib Ayad**

A dissertation submitted to AASTMT in partial
Fulfillment of the requirements for the award of the degree of

**MASTER of SCIENCE**

In

Electronics and Communication Engineering

Supervisors

**Prof. Khaled A. Shehata**
Electronics & Communications
Engineering Department
Arab Academy for Science, Technology and
Maritime Transport, Cairo, EGYPT

**Assoc. Prof. Mohamed El-Mahallawy**
Electronics & Communication
Engineering Department
Arab Academy for Science, Technology and
Maritime Transport, Cairo, EGYPT

**Assoc. Prof. Hanady H. Issa**
Electronics & Communications
Engineering Department
Arab Academy for Science, Technology and
Maritime Transport, Cairo, EGYPT

# ACKNOWLEDGEMENTS

My deepest Appreciation goes to my supervisors, Prof. Dr. Khaled A. Shehata, Dr. Mohamed S. El-Mahallawy, and Dr. Hanady H. Issa for their continuous guidance and support during my research.

I dedicate this thesis to all my family for their support to me during my educational journey throughout the years. For supporting me and encouraging me.

I would like to express my gratitude to all my friends, for giving me hope when I needed it. And for their belief in me.

# ABSTRACT

A number of illnesses that affect people's daily life are caused by numerous sleep disorders which usually have common symptoms. People suffering from such illnesses are more vulnerable to household and work accidents. While there is no treatment for these kinds of neurological disorders, symptoms can be managed with behavioral and medical therapies to reduce the intensity and occurrence rates.

The scope of this thesis is to design and implement a portable system that will assist Narcoleptic patients, in real-time, to aid them into leading a more productive life. The system consists of two main units, a feature extraction unit based on the Wavelet Packet Transforms (WPT), and a classification unit based on the Support Vector Machine (SVM) algorithm. The SVM used is a non-linear and multi-class classification using the Radial Basis Function (RBF) as the kernel function.

The design undergoes two testing processes, the first one is a software simulation using MATLAB 2014a tool, while the second is the RTL simulation using ModelSim 6.3a. The hardware RTL simulation is compared to the software simulation to determine the average error of the feature extraction unit, and the classifier accuracy. The feature extraction unit achieved an average percentage of error 0.1618% with respect to the software simulation, while the classifier achieved an average accuracy rate of 92.14% and 90.5% by software and the RTL simulations respectively.

The design is synthesized and routed – using Quartus II V. 13 – on Cyclone IV EP4CGX30BF14C6 chip from Altera. The gate level simulation resulted with the same results as the RTL simulation.

The implementation achieved an acceptable time delay, where the feature extraction and the classification are executed in 3.85ms. The total area usage of the selected FPGA chip is 97.68% of the available logic elements, 8.42% of the available memory bits, and 32.5% of the available embedded multiplier units.

# TABLE OF CONTENTS

**Chapter 1: Introduction** ..... 1

**1.1 Introduction** ..... 1

**1.2 Narcolepsy Definition, Symptoms, and Diagnosis** ..... 1

**1.3 Thesis Objective and Motivation:** ..... 2

**1.4 Thesis Outline** ..... 3

**Chapter 2: Medical Background and Literature Review** ..... 4

**2.1 Classical Narcolepsy Classification; Standards of Sleep Stage Recording** ..... 4

**2.1.1 Sleep Study** ..... 5

**2.1.2 Alertness Stages:** ..... 6

**2.1.3 Polysomnography Waves and Their Derivations:** ..... 8

**2.1.4 General Considerations for Recording Sleep** ..... 10

**2.2 Review on Narcolepsy Assistive Devices and Classification Algorithms** ..... 11

**Chapter 3: Wavelet Transforms and Support Vector Machines** ..... 13

**3.1 Introduction to Wavelet Transforms** ..... 13

**3.1.1 Fourier Transform** ..... 13

**3.1.2 Short Time Fourier Transform** ..... 15

**3.1.3 Wavelet Transforms** ..... 15

**3.1.3.1 Continuous wavelet Transform** ..... 16

**3.1.3.2 Discrete wavelet Transform (DWT)** ..... 18

**3.1.3.3 Wavelet Packet Transform (WPT)** ..... 21

**3.2 Support Vector Machines** ..... 21

**3.2.1 Linearly Separable Binary Classification** ..... 21

**3.2.2 Binary Classification for Data that is not Fully Linearly Separable** ..... 23

**3.2.3 Nonlinear Binary Support Vector Machines** ..... 24

**3.2.4 Multi-class Support Vector Machines** ..... 25

**3.2.4.1 One Against All Method** ..... 25

**3.2.4.2 One Against One Method** ..... 26

**Chapter 4: Proposed Design and Software Simulation** ..... 27

**4.1 Proposed Design** ..... 27

**4.1.1 EEG Database Source** ..... 28

**4.1.2 Feature Extraction Unit** ..... 29

**4.1.2.1 The Wavelet Packet Transform Unit** ..... 30

**4.1.2.2 The Features Calculation Unit** ..... 32

**4.1.2.3 Features Normalization** ..... 34

**4.1.3 Classifier Unit** ..... 35

**4.1.3.1 Kernel Selection** ..... 35

**4.1.3.2 Cross-Validation and Grid-Search** ..... 36

**4.1.3.3 Support Vector Machine Classification and Prediction Script** ..... 37

**Chapter 5: FPGA Hardware Implementation of the Proposed Design** ..... 39

**5.1 VHDL Implementation of the Proposed Design and RTL Simulations** ..... 39

**5.1.1 Feature Extraction Unit** ..... 39

**5.1.1.1 Wavelet Packet Transform (WPT) Unit** ..... 39

**5.1.1.2 Features Calculation Unit** ..... 50

**5.1.1.3 Features Extraction Unit Integration** ..... 53

**5.1.2 Support Vector Machine Classifier Unit** ..... 54

**5.1.2.1 Kernel Function Unit** ..... 55

**5.1.2.2 Integration of the SVM Classifier Unit** ..... 58

**5.1.3 Narcolepsy Assistive Device System Integration** ..... 59

**5.2 Hardware Gate Level Simulation** ..... 59

**5.2.1 Gate level Simulation of the Feature Extraction Unit** ..... 60

**5.2.1.1 Wavelet Packet Transform Unit** ..... 60

**5.2.1.2 Features Calculation Unit** ..... 61

**5.2.1.3 Features Extraction Unit Integration** ..... 62

**5.2.2 Gate Level Simulation of the SVM Classifier Unit** ..... 63

**5.2.3 Gate Level Simulation of the Integrated Narcolepsy Assistive Device** ..... 64

**Chapter 6: Conclusion & Future Work** ..... 66

**6.1 Conclusion** ..... 66

**6.2 Future Work** ..... 67

**REFERENCES** ..... 68

# LIST OF ABBREVIATIONS

| | |
|---|---|
| ALU | Arithmetic Logic Unit |
| ASIC | Application Specific Integrated Circuit |
| BDI | Beck Depression Inventory |
| CWT | Continuous Wavelet Transform |
| DWT | Discrete Wavelet Transform |
| EDF | European Data Format |
| EDS | Excessive Daytime Sleepiness |
| EEG | Electroencephalography |
| EMG | Electromyography |
| EOG | Electrooculography |
| FF-NN | Feed Forward Neural Networks |
| FT | Fast Fourier Transform |
| HPF | High Pass Filter |
| LPF | Low Pass Filter |
| LVQ | Linear Vector Quantization |
| NREM | None Rapid Eye Movement |
| PNN | Probabilistic Neural Networks |
| PSG | Polysomnography |
| RAM | Random Access Memory |
| RBF | Radial Basis Function |
| REM | Rapid Eye Movement |
| RKS | Rechtschaffen and Kales |
| SV | Support Vector |
| SVM | Support Vector Machine |
| VHDL | VHSIC Hardware Descriptive Language |
| VHSIC | Very High Speed Integrated Circuit |
| WPT | Wavelet Packet Transform |
| WT | Wavelet Transform |

# LIST OF FIGURES

Figure 2.1: Standard Human Sleep Cycle ..... 5

Figure 2.2: Different Alertness Levels in the EEG Spectrum ..... 8

Figure 2.3: 10-20 EEG Electrode Placement System ..... 9

Figure 2.4: EOG electrode Placement System ..... 9

Figure 2.5: Facial Muscles ..... 10

Figure 3.1: Time-Frequency Analysis using FT ..... 14

Figure 3.2: Comparison between Different Techniques [11] ..... 16

Figure 3.3: Dual Frequency Sinusoidal Waveform ..... 17

Figure 3.4: CWT of the Dual Frequency Sinusoidal Waveform ..... 17

Figure 3.5: Discrete Wavelet Transform Decomposition by Filter Banks [12] ..... 19

Figure 3.6: Input Waveform for DWT ..... 20

Figure 3.7: Signals Generated by the Analysis Filters ..... 20

Figure 3.8: Wavelet Packet Transform Decomposition Tree ..... 21

Figure 3.9: Hyper-plane through Two Linearly Separable Classes [13] ..... 22

Figure 3.10: Hyper-plane through Two Non-linear Separable Classes [13] ..... 23

Figure 3.11: Example on Non-linear Classification ..... 24

Figure 4.1: Functional Block Diagram of the Proposed Design ..... 27

Figure 4.2: Modes of Operation Flow Chart ..... 28

Figure 4.3: Occurrence Rate for Each Class in the Testing Epochs ..... 29

Figure 4.4: Block Diagram of the Proposed Feature Extraction Unit ..... 29

Figure 4.5: 2nd Order Debauchee Filter Response ..... 30

Figure 4.6: Proposed Wavelet Packet Tree ..... 30

Figure 4.7: An EEG Epoch Filtered to 6 Frequency Bands S1 to S6 ..... 31

Figure 4.8: Software Simulation Script Flowchart ..... 34

Figure 4.9: SVM Parameter Selection ..... 36

Figure 4.10: SVM Prediction Flowchart ..... 37

Figure 5.1: Wavelet Packet Transform Block Diagram ..... 40

Figure 5.2: 2nd Order Filter Block Diagram ..... 40

Figure 5.3: 2nd Order Filter VHDL Implementation ..... 41

Figure 5.4: Filter RTL Simulation Results ..... 42

Figure 5.5: RAM Unit Implementation ..... 43

Figure 5.6: RAM Unit RTL Simulation ..... 44

Figure 5.7: Register Banks VHDL Implementation ..... 45

Figure 5.8: Register Banks Unit RTL Simulation ..... 45

Figure 5.9: Wavelet Packet Transform VHDL Implementation ..... 46

Figure 5.10: Control Unit Partial Work Plan ..... 47

Figure 5.11: WPT RTL Simulation ..... 48

Figure 5.12: Feature Calculation Unit Implementation ..... 50

Figure 5.13: Feature Calculation Unit VHDL Implementation ..... 50

Figure 5.14: Feature Calculation Unit RTL Simulation Result ..... 52

Figure 5.15: Feature Extraction Unit Integration ..... 53

Figure 5.16: Feature Calculation Unit RTL Simulation Results ..... 54

Figure 5.17: Support Vector Machine Classifier Block Diagram ..... 54

Figure 5.18: Functional Block Diagram of the Kernel Function Unit ..... 55

Figure 5.19: VHDL Implementation of the Kernel Function Unit ..... 55

Figure 5.20: RTL Simulation of the Exponential Function ..... 56

Figure 5.21: RTL Simulation of the Kernel Function Unit .....58

Figure 5.22: SVM Classifier Implementation.....58

Figure 5.23: RTL Simulation of the SVM classifier unit .....58

Figure 5.24: Narcolepsy Assistive Device System Integration .....59

Figure 5.25: Filter Pair Gate Level Simulation result.....60

Figure 5.26: WPT Gate Level Simulation for extracting the first coefficient .....61

Figure 5.27: WPT Gate Level Simulation for extracting the last coefficient .....61

Figure 5.28: Feature Calculation Unit Gate Level Simulation .....62

Figure 5.29: Features Extraction Unit Gate Level Simulation Results.....62

Figure 5.30: Exponential ROM Gate Level Simulation .....63

Figure 5.31: SVM Gate Level Simulation Results .....64

Figure 5.32: Narcolepsy Assistive Device Gate Level Simulation Results.....64

# LIST OF TABLES

Table 2.1: EEG Bands and Corresponding State of Mind ..... 6

Table 2.2: Outline of Sleep Scoring Criteria According to Standard Manual [3] ..... 7

Table 2.3: Montage for Monitoring Sleep States [3] ..... 11

Table 2.4: Classification Methods Accuracy [5] ..... 11

Table 3.1: Popular Kernel Functions [15] ..... 24

Table 4.1: 2nd Order Debauchee Filter Coefficients ..... 30

Table 4.2: Desired Frequency Bands Coefficients ..... 31

Table 4.3: Sample of Coefficients Extracted from an Epoch by MATLAB Simulation ..... 32

Table 4.4: Normalized Values of the Features Extracted for an Epoch ..... 35

Table 4.5: Different Kernel Accuracy Results ..... 38

Table 5.1: Filter Pair Test Results ..... 42

Table 5.2: Sample of Coefficients Extracted from an Epoch by the RTL Simulation ..... 49

Table 5.3: Percentage of Error in the Different Bands ..... 49

Table 5.4: Calculated Features for the First Epoch ..... 53

Table 5.5: Exponential Unit Input Error ..... 56

Table 5.6: Exponential Unit Output Error ..... 57

Table 5.7: Summary of MATLAB and RTL Simulation Result ..... 59

Table 5.8: Area Report of the Feature Extraction Unit ..... 63

Table 5.9: Area Report of the SVM Classifier Unit ..... 64

Table 5.10: Area Usage Report of the Overall Implementation ..... 65

# LIST OF PUBLICATIONS

Fadi S. Ayad, Hanady H. Issa, Mohamed S. El-Mahallawy, Khaled A. Shehata, *"Design and Implementation of a Real-time Sleep Stage Monitoring System for Narcolepsy Diagnosis"*, International Conference on Advanced Technology & Sciences, Selcuk University, Antalya, TURKEY, 2014. Pages 975 - 979, 2014.

# Chapter 1

# Introduction

## 1.1 Introduction

Research in Biomedical engineering and its applications has been rapidly growing over the past few years, as medical treatment is becoming more dependent on the technological advances. The main purposes of this field is to provide humans with better, healthier and more comfortable medical care, which reflects on the society by leading to a better productive life for its citizens.

Research in biomedical engineering is showing promising new treatments for illnesses and disorders that medical science on its own cannot provide an appropriate medication. Currently many leading universities are focusing on biomedical research for its importance to human health and lifestyle.

## 1.2 Narcolepsy Definition, Symptoms, and Diagnosis

Narcolepsy is a chronic neurological disorder of sleep regulation that affects the sleep cycle. It is characterized by uncontrollable recurring episodes of daytime sleepiness. The episodes may last from few minutes up to a complete hour and it is accompanied by intrusion of the dreaming state - known as the Rapid Eye Movement (REM) - into the waking state [1].

Its most common symptoms include Excessive Daytime Sleepiness (EDS), loss of muscle tone (Cataplexy), distorted perceptions (Hallucinations), and the inability to move or to talk (Sleep Paralysis). Referring to a survey carried by the UK Narcolepsy patient association based to the Beck Depression Inventory (BDI), degrees of depression are common among narcoleptic patients, which is an expected evolution to a series of social, educational, psychological, and financial difficulties experienced by a narcoleptic patient. Furthermore narcoleptic individuals are at an elevated risk for motor vehicle, occupational and household accidents.

Origin and causes of Narcolepsy are still unclear, but the general consensus is the patient genetics accompanied by an environmental trigger of some sort – virus, for example – may affect the brain chemicals and causes Narcolepsy.

Diagnosis of Narcolepsy commonly performed at a sleep clinic, by recording the brain activities of the patients during his sleep, followed by a visual scoring of the recorded data.

## 1.3 Thesis Objective and Motivation:

Approximately 3 million people world-wide have Narcolepsy. The prevalence of Narcolepsy is approximately 0.05% of the population, however it often goes unrecognized. Although Narcolepsy is as common as many other conditions (such as Parkinson's Disease or Multiple Sclerosis), fewer people know about Narcolepsy, and it can sometimes be mistaken for other medical conditions such as depression, psychiatric illness, epilepsy, chronic fatigue syndrome, or a side-effect of medication.

Narcoleptic patients are always at high risks due to sudden sleep episodes or cataplexy. For example if the patient is driving a car while an episode occurs, this might lead to a severe accident. Life style changes are required, such as ensuring a regular sleep schedule, avoiding periods of sleep deprivation, scheduling several short naps along the day usually before driving, avoiding occupations that involve excessive driving and operating heavy machinery, avoiding caffeine and alcohol.

Narcolepsy is diagnosed using polysomnography (PSG) according to the standard written in 1968 [2] which requires the patient to spend the night at a sleep clinic in order to record his brain activity for a later analysis that would be performed by a sleep physician. Draw backs for this classical method are:

1- Long hours spent by the patient at sleep clinic.
2- Unfamiliarity of the clinic to the patient may leads to in-accurate results.
3- Scoring the sleep stages and the classification of the disorder are made by the human eye which might lead to in-accurate results.
4- Real-time analysis is impossible with the classical scoring method.

The scope of this thesis is to design and implement a high speed real-time system that predicts narcoleptic episodes. The proposed system design consists of a memory for data logging, feature extraction block, the classifier block, and finally a user interface. The system is designed to be small enough for portability, fast enough for real-time computation, simple enough for ease of use – using the least possible number channels - while maintaining the

classifier high accuracy rate, and most importantly reducing power consumption to the minimum.

Since Narcolepsy puts the patient in danger, a real-time classifier helps narcoleptic patients live a better and more productive life as it would give a good prediction before the occurrence of an episode. These predictions will allow the patient to take proper precautions when needed to prevent accidents. It will also help improving the accuracy rate over the conventional classification as patient will sleep in his own familiar environment. The physician will be able to review the data logged at any time for the purpose of adjusting medications.

## 1.4 Thesis Outline

This thesis is organized as follows:

**Chapter 2** provides all the necessary medical background and the literature review related to the Narcolepsy disorder. A detailed discussion of the medical terminology, definitions and its standards is presented in details. Furthermore previous work in this field of research is discussed and summarized.

**Chapter 3** discusses in details the theory of wavelet transforms and the support vector machines, as the system designed in this thesis uses these discrete wavelet transform for the feature extraction and the support vector machines for its classification.

**Chapter 4** presents the full details of the proposed design and the software simulation conducted for each building block constructing this design.

**Chapter 5** presents the full implementation of the design proposed in chapter 4, along with all the RTL and post-routing simulations conducted to test this system functionality.

**Chapter 6** summarizes this thesis, discussing the results found in chapter 4, and concluding some recommendations for future work to improve the developed system. And finally the work is terminated by a list of references.

# Chapter 2

# Medical Background and Literature Review

In this chapter, both the medical background and all the recently published studies related to the scope of this thesis, are introduced. The medical background dealing with the conventional methods of narcolepsy treatments is discussed in the next section, while in the second section a discussion of the recently developed assistive devices is discussed.

## 2.1 Classical Narcolepsy Classification; Standards of Sleep Stage Recording

Narcolepsy classification is mainly based on polysomnography, otherwise called as sleep study. It's a process at which a complete recording of the brain waves is acquired while the patient is asleep during the night. These recordings consist of Electroencephalography (EEG), Electrooculography (EOG), and Electromyography (EMG) waves followed by a visual scoring of these recording by a physician [2].

Since 1968 visual sleep stage scoring has been performed according to the guidelines established by the committee led by Allan Rechtschaffen and Anthony Kales. The aim of these guidelines often called the “Manual of Rechtschaffen and Kales” (RKS), was to increase the comparability of results from different laboratories [2].

The manual is based on the psychophysiological knowledge of the sleep process at that time, providing the minimum requirements for polysomnographic sleep studies of adult humans. It also provides the classification of sleep stages, standardized terminology and definitions of the different parameters obtained from sleep recordings.

Since these guidelines were designed for paper recordings, they contains definitions for filters, gain, paper speed, pen deflection, and the number of channels recorded, etc. Some of the requirements and instructions are no longer necessary during recording with the increasing use of digital polysomnograph in which the parameters can be adjusted afterwards.

In the following sub-sections, a brief introduction of the human sleep cycle will be introduced, followed by a more detailed explanation of the different alertness stages that occurs in a normal cycle, and finally the electrode placement and signal considerations for accurate classification.

### 2.1.1 Sleep Study

Nocturnal Polysomnography (PSG) also known as sleep study, is a comprehensive recording of the bio-physiological changes that occur during sleep. It is usually performed at night, when most people sleep. The PSG monitors many body functions including brain (EEG), eye movements (EOG), muscle activity or skeletal muscle activation (EMG) and heart rhythm (ECG) during sleep.

The human sleep cycle consists of different states of alertness known as the sleep stages. Sleep stages are divided into two groups, Rapid Eye Movement (REM) and None Rapid Eye Movement (NREM). The REM stage is also known as the state at which dreams occur and it is characterized by a visible movement in the eyes. The NREM stage consists of 4 stages that describe the transition from wakefulness to the dreaming state [2].

In a normal sleep cycle, first stage occurs for approximately 10 minutes, followed by the second stage for about 30 minutes, stage 3 is usually short and followed by stage 4 that can last up to 40 minutes, finally REM state occurs from 5 to 20 minutes. In a typical 8 hours sleep, four to five complete sleep cycles occur where each cycle lasts longer as its respective REM stage extends. The standard sleep cycle can be seen in the illustration of Figure 2.1.

The sleep stages are scored according to RKS manual written in 1968 [2].

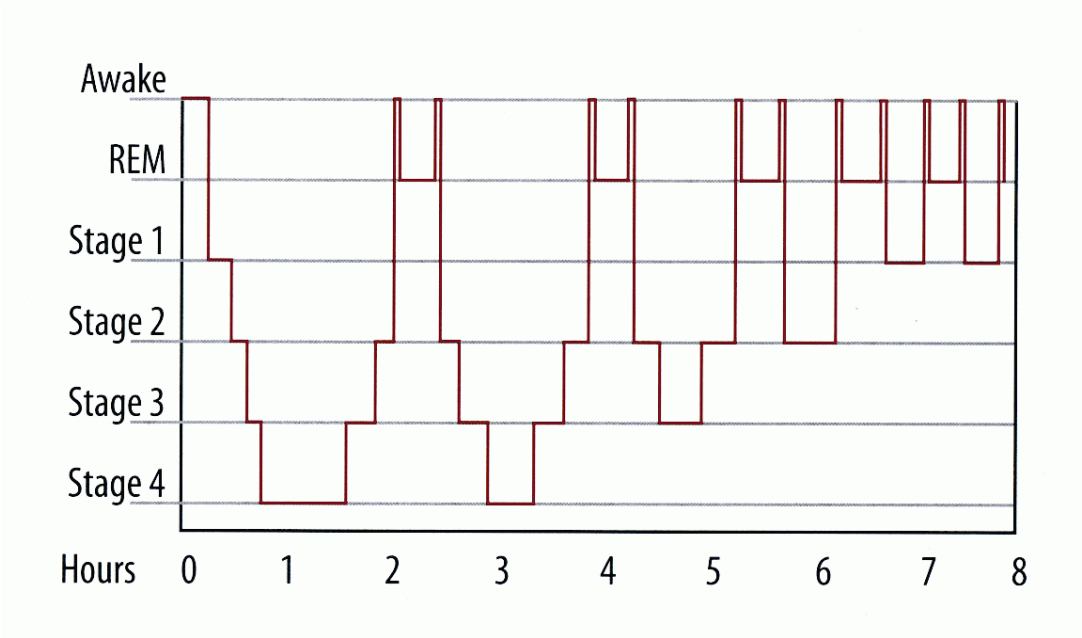


Figure 2.1: Standard Human Sleep Cycle

### 2.1.2 Alertness Stages:

In this section, a full and detailed explanation of the different stages in human sleep cycle along with their corresponding signal characteristics and descriptions is explained. The human state of mind can be determined with the following EEG bands as seen in Table 2.1

Table 2.1: EEG Bands and Corresponding State of Mind

| EEG Band | Frequency Range | State of mind |
|---|---|---|
| Beta | 14 – 30 Hz | Associated with anxiety and depression or use of sedatives.<br>Highly alert and focused |
| Spindles | 12 – 14 Hz | Represents periods where the brain is inhibiting processing to keep the sleeper in a tranquil state |
| Alpha | 8 – 13 Hz | Present when awake but eyes closed (relaxed but alert).<br>Disappear when eyes are open or with mental concentration. |
| Theta | 4 – 7 Hz | Drowsiness or Slow brain activity |
| Delta | 0.5 – 3.5 Hz | Deep Sleep state |

**Woke-up Stage (Stage W)** is characterized by low voltage (10 – 30 μV) and mixed EEG frequencies, mainly consists of alpha activity [2].

**Stage 1** is defined by a relatively low voltage, mixed frequency EEG with a prominence of activity in the 2-7 Hz range. Stage 1 occurs most often in the transition from wakefulness to the other sleep stages following body movement during sleep. Stage 1 tends to be relatively short during nocturnal sleep, ranging from 1 to 7 minutes. The highest voltage in the 2-7 Hz activity is about 50-75 μV and tends to occur in irregularly spaced bursts mostly during the latter portions of the stage. Vertex sharp waves may occur; their amplitude can reach up to 200 μV. Slow eye movement may be present during the first stage. Stage one is scored when clearly defined K complexes, a sharp negative wave followed by a slower positive component, and sleep spindles are absent and the epoch is characterized with alpha activity combined with mixed frequency EEG and the amount of alpha activity is less than 50% of the epoch [2].

**Stage 2** is characterized by wave patterns, sleep spindles, K complexes and the absence of slow waves EEG. Traces of low voltage activity in the frequency range of 12-14 Hz may begin to appear in stage 1 as the transition to stage 2 approaches, but will only be characterized as stage 2 when the rhythmic bursts are clearly visible for at least 0.5 seconds. If the time between two

succeeding occurrences of sleep spindles or K complexes is lower than 3 minutes, this interval is scored as stage 2 otherwise it is scored as stage 1 [2].

**Stage 3** is characterized by slow EEG waves slower than 2 Hz with amplitudes above 75 μV. In order to score a stage 3, the previously identified characteristics must be present for 20 to 50% of an epoch. Sleep spindles and K complexes may occur during stage 3 [2].

**Stage 4** is similar to Stage 3 but having the slow waves with 2 Hz and slower with amplitudes above 75 μV appearing in more than 50% of an epoch [2].

**Rapid Eye Movement "REM"** stage is scored when the patient's eye lids move rapidly which usually indicated a dreaming state. It is correlated with a set of psychological activities such as change in heart rate and blood pressure [2].

Table 2.2 summarizes the sleep stage scoring criteria [3], while Figure 2.2 illustrates the EEG recording for the different states from wake until REM sleep.

Table 2.2: Outline of Sleep Scoring Criteria According to Standard Manual [3]

| Stage/State | EEG | EOG | EMG |
|---|---|---|---|
| **Relaxed Wakefulness** | Eyes closed: Rhythmic alpha (8-13 Hz) Prominent in occipital.<br>Eyes open: relatively low voltage, mixed frequency | Voluntary control: REMs<br>when drowsy SEMs | Tonic activity, relatively high;<br>voluntary movement |
| **NREM** | | | |
| Stage 1 | Relatively low voltage, mixed frequency<br>May be theta 3-7 Hz activity with greater amplitude.<br>Vertex Sharp Waves<br>Synchronous high-voltage theta bursts in children | SEMs | Tonic activity, low level |
| Stage 2 | Background: relatively low voltage, mixed frequency<br>Sleep Spindles: (12- 14 Hz) ≥ 0.5 sec<br>K complexes: ≥ 0.5 sec; spindles may ride on Ks; Ks maximal in vertex; spontaneous or in response to sound. | Occasionally SEMs near sleep onset | Tonic activity, low level |
| Stage 3 | ≥ 20 % ≤ 50% high amplitude (≥ 75 μV), slow frequency (≤ 2 Hz); maximal in frontal | None, picks up EEG | Tonic Activity, low level |
| Stage 4 | > 50% high amplitude slow frequency | None, picks up EEG | Tonic Activity, low level |
| **REM** | Relatively low voltage, mixed frequency<br>Saw-tooth waves<br>Theta activity; slow alpha | Phasic REMs | Tonic Suppression; phasic twitches |
| **Movement Time** | Obscured | Obscured | Very high activity |

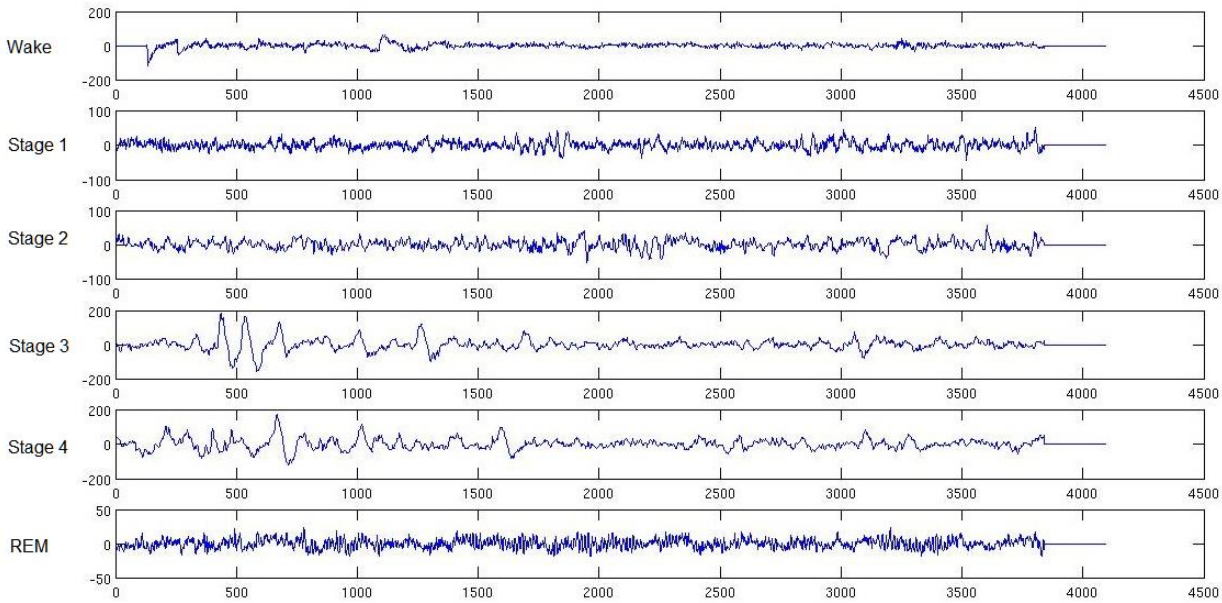


Figure 2.2: Different Alertness Levels in the EEG Spectrum

### 2.1.3 Polysomnography Waves and Their Derivations:

This section discusses the various signals recorded during a standard polysomnography along with their electrode placement and derivations.

**a) Electroencephalogram (EEG)**

For a reliable recording of EEG, it is important to carefully choose the electrode placement according to the 10-20 system represented in Figure 2.3. A grid is placed over the skull and points of intersections denote electrode placement locations. The name of the system derives from measurements made at intervals of 10% or 20% of the total distance between landmarks. The four landmarks of the system are Nasion, Inion, and left and right Pre-auricular Points. These measurements are made for each patient specifically [3, 4].

The standard manual recommends referential recording of one EEG channel either C3 or C4, referenced to an indifferent auricularly placed electrode on the contra-lateral mastoid or ear lobe: hence C3/A2 or C4/A1 [2, 3, 4]. Sleep stage scoring doesn't require measurement of focal EEG activity or regional comparisons. Rather, all of the EEG waveforms used to distinguish sleep stages are well visualized at C3 or C4 with the relatively large inter-electrode distance afforded by a contra-lateral reference. Vertex sharp waves and K complexes which are maximal over the vertex, are clearly evident at C3 and C4; high voltage slow waves characteristic of

deep NREM sleep are seen maximally in frontal regions yet seen clearly on central derivations. Alpha rhythm can be characterized centrally in most human beings, therefore, only C3/A2 or C4/A1 differential electrode is used in standard assessment of sleep stages [2, 3, 4].

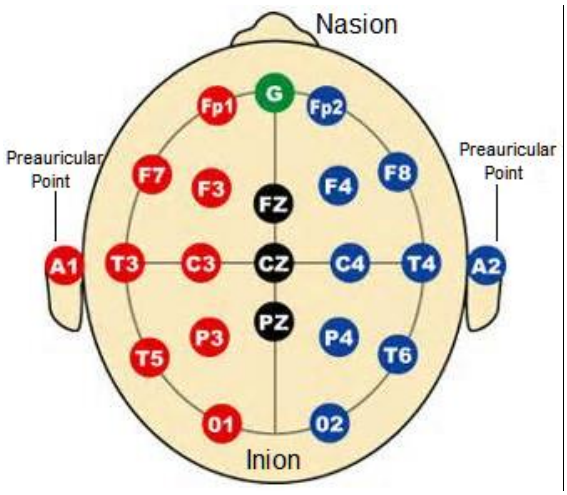


Figure 2.3: 10-20 EEG Electrode Placement System

### b) Electrooculogram (EOG)

There are two primary reasons to record eye movement activity during sleep. The first reason is to record the cardinal sigh of REM sleep which is an essential sleep stage scoring criterion, and the second one is that the onset of sleep in most human beings is accompanied by slow rolling eye movement. Although these slow eye movements (SEMs) are not essential to sleep staging, they often provide very useful information [3].

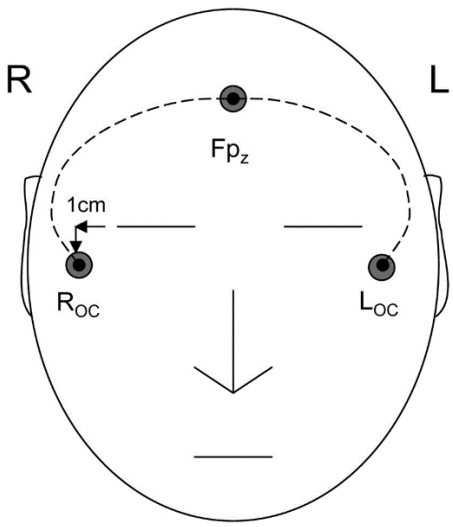


Figure 2.4: EOG electrode Placement System

The EOG recordings are based on the small electro-potential difference from the front to the back of the eye. Movement of the eye can be measured from electrodes placed beside the eyes. An electrode near to the corena will register a positive potential while an electrode near to the retina will record a negative potential. As the eye moves, the positions of the corena and retina change relative to the fixed position of the electrodes which will register a change in potential.

As recommended in the standard manual, continuous referential recording of two EOG leads must be registered. Using either (ROC/A1, LOC/A1) or (ROC/A1, LOC/A2) electrodes; where ROC and LOC placement are illustrated in Figure 2.4. In the second electrode placement the contra-lateral references maximize the signal amplitude for both EOGs, conjugate eye movement will register as out-of-phase; while EEG activity reflected in the EOG channels will be seen as in-phase deflections [2, 3].

**c) Electromyogram (EMG)**

In polysomnography recording the EMG is recorded from the muscles beneath the chin as illustrated in Figure 2.5 and used as a criterion for staging REM sleep. The electrodes are placed beneath the chin overlying the mentalis/submentalis muscles. The EMG is recorded bipolarly where the electrode pair selected should produce the record of highest quality [2, 3].

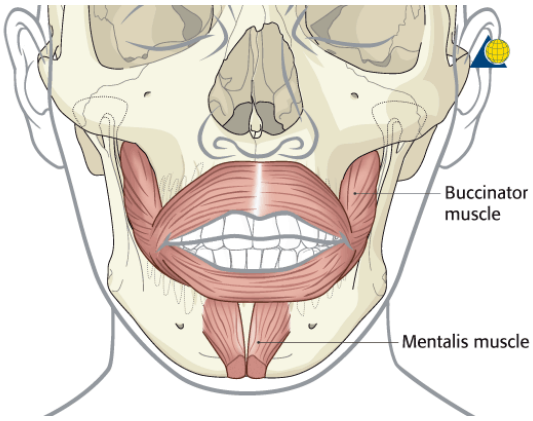


Figure 2.5: Facial Muscles

### 2.1.4 General Considerations for Recording Sleep

The minimum recommended requirement for sleep staging is four channels; one channel EEG, two channels EOG, and one channel EMG recordings are sufficient for correctly scoring sleep. From Table 2.3 [3], each channel and its recommended electrodes are summarized.

Table 2.3: Montage for Monitoring Sleep States [3]

| Channel | Electrode placement |
|---|---|
| EEG | C3/A2 or C4/A1 |
| EOG | ROC/A1 and LOC/A1 |
| EMG | Mentalis/Submentalis |

## 2.2 Review on Narcolepsy Assistive Devices and Classification Algorithms

In 2007, a thesis titled “classification of sleep staging for Narcolepsy assistive device” [5] was presented by Shuo Zhang. This research studied three different techniques for the sleep classification - Linear Vector Quantization “LVQ”, Probabilistic Neural Networks “PNN”, and Feed Forward Neural Networks “FF-NN” - while a hardware implementation was built for the most suitable technique, the FF-NN as it provided the highest accuracy. Table 2.4 summarizes the results of these classification methods [5].

Table 2.4: Classification Methods Accuracy [5]

| Classification Methods | Accuracy Results |
|---|---|
| 1- Linear Vector Quantization | 70% |
| 2- Probabilistic Neural Networks | 79% |
| 3- Feed Forward Neural Networks | 85% |

For the implementation, a micro-controller was used to control the data flow, while the DSP kit “TMS320C6711” was used to implement the feature extraction and the classifier. Continuous wavelet transform with Debauchee mother wavelet was used for the feature extraction [5, 6].

In 2008 a paper was published by S. R. I. Gabran. This paper tackled the same classification techniques as in the previous research. The results were similar to those summarized in Table 2.4 [7].

In 2009 a paper was published by S. R. I. Gabran at the IEEE EMBS annual conference [8]. This paper could be considered as continuation to the previously published paper by the same author. In this research a new techniques based on the Support Vector machine “SVM” algorithm is introduced and compared to the previous classification Techniques. SVM was found to provide a better accuracy at 96 %.

A prototype implementation was developed using the AR-B1831 single board computer with an Intel Pentium Mobile processor. The single electrode chosen for this prototype was the EEG

C3/A2 channel, while CWT with Debauchee mother wavelet were implemented for the feature extraction [8].

However previous efforts developed similar devices to our thesis, these products were not hardware optimized for performance, portability, power consumption, and real-time operation. The scope of our research is to design and implement a hardware optimized for a narcolepsy assistive device working with the SVM classifiers for best performance, yet hardware implemented using VHDL unlike the general purpose processor device used in [8].

# Chapter 3

# Wavelet Transforms and Support Vector Machines

The system designed in this thesis uses the wavelet transforms for the feature extraction and the support vector machines for classification. In this chapter a detailed explanation of the theory of both wavelets and support vector machines is elaborated.

## 3.1 Introduction to Wavelet Transforms

All the signals in practice, are time domain signals in their raw format. This representation is not always the best representation of the signal. In many cases, the most useful information is hidden in the frequency components of that signal. Frequency analysis of a signal can be processed using different kind of transforms. Each of these techniques has its own area of application, with advantages and disadvantages that are discussed in this chapter [9].

### 3.1.1 Fourier Transform

Fourier Transform (FT) is a method used to represent any given waveform by a series of sinusoidal signals. If the FT of a signal in time domain is taken, the frequency-amplitude representation of that signal is obtained, resulting in a plot with one axis being the frequency while the other is the amplitude. This plot, as example in Figure 3.1, tells us how much of each frequency exists in the given waveform [10].

The FT is a reversible transform, that is, it allows to go back and forward between the raw and transformed signals. However, only either of them is available at any given time, that is, no frequency information is available in the time-domain signal, and no time information is available in the Fourier transformed signal. Equations 3.1 and 3.2 formulates the transformation process from time domain to frequency domain and from frequency domain to time domain respectively [10].

$$X(f) = \int_{-\infty}^{\infty} x(t) \times e^{-2J\pi ft} dt \tag{3.1}$$

$$x(t) = \int_{-\infty}^{\infty} X(f) \times e^{2J\pi ft} df \tag{3.2}$$

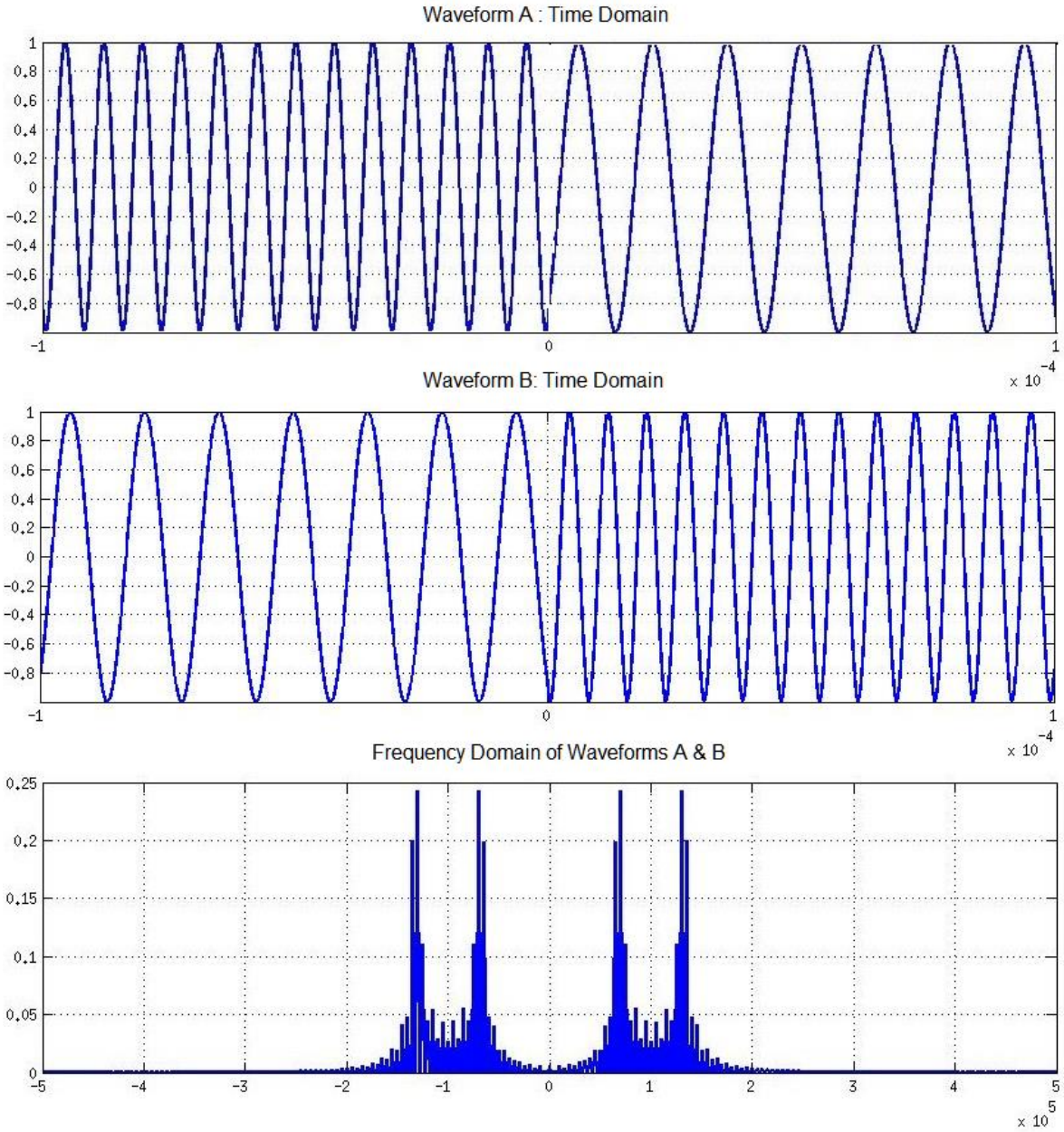


Figure 3.1: Time-Frequency Analysis using FT

Figure 3.1 illustrates the disadvantage of FT, where two completely different waveforms might yield the same frequency domain response if they obtain the same frequency components regardless of their time of occurrence. In certain applications knowing the frequency components of a given waveform is not enough, the time of occurrence of each frequency is crucial to the analysis of this signal. FT will not be useful in such applications, therefore, another method of processing is needed.

### 3.1.2 Short Time Fourier Transform

Since FT doesn't work for non-stationary signals, a different technique had to be implemented in order to process non-stationary waveforms. The Short Time Fourier Transform (STFT) assumes that some portion of the given waveform is stationary on its own. In STFT the signal is divided into small segments, where these segments of the signal can be assumed to be stationary. For this purpose a window function "w" is chosen. The width of this window must be equal to the segment of the signal where its stationery is valid.

This window function is first located at the very beginning of the signal. The window function and the signal are then multiplied. If the window is a rectangle, with amplitude "1", then the product will be equal to the signal itself. The FT of this product is to be taken, similar to taking the FT of any stationary signal.

The next step, would be shifting this window to a new location, multiplying with the signal, and taking the FT of the product. This procedure is followed, until the end of the signal is reached by shifting the window. Equation 3.3 formulates the STFT transform function [10].

$$\mathrm{STFT}(t,f) = \int_t \left[ x(t) \,.\, w^*(t-t') \right] \times e^{-J2\pi ft} \, dt \qquad (3.3)$$

However, STFT is not capable for achieving multi-class resolution, since the window function is fixed in width throughout the whole transform process; therefore the wavelet transforms were developed to overcome this problem.

### 3.1.3 Wavelet Transforms

In analogy to sinusoidal signals which are the constituting signals to the Fourier Transform (FT), there are the wavelets which are the constituting signals to the Wavelet Transform (WT). Wavelets are the foundation of a powerful new approach to the signal processing and analysis called the multi-resolution theory which STFT is not capable of achieving.

Some features within a signal might go undetected at one resolution however, they might be easily detected at another, and therefore this theory is concerned with the representation and analysis of signals at more than one resolution.

### 3.1.3.1 Continuous wavelet Transform

The Continuous Wavelet Transform (CWT) was developed as an alternative approach to the STFT to overcome the resolution problem. The wavelet analysis is done in similar way to the STFT, in the sense that the signal is multiplied with a function similar to the window function in STFT, and the transform is computed separately for different segments of the time-domain signal. However, there are two main differences between STFT and CWT [11].

1- The FT of the windowed signals are not taken.
2- The width of the window, the most significant characteristic of the WT, is changed as the transform is computed for every single spectral component.

The CWT is defined as formulated in Equation 3.4, where $\tau$ and S are the translation and the scaling parameters, respectively. $\Psi(t)$ is the transforming function, also called as the mother wavelet.

$$CWT_x^{\Psi}(\tau, S) = \frac{1}{\sqrt{|S|}} \int x(t)\, \Psi^*\left(\frac{t-\tau}{S}\right) dt \qquad (3.4)$$

Wavelets differ from Fourier Transform in that wavelets allow us to inspect a signal in both space or time domain and frequency domain at the same time, which the FT lacks, and they process the signal with different resolutions on the contrary STFT is a single resolution analysis. Figure 3.2 illustrates the frequency time characteristics in WT compared to STFT and FT techniques [11].

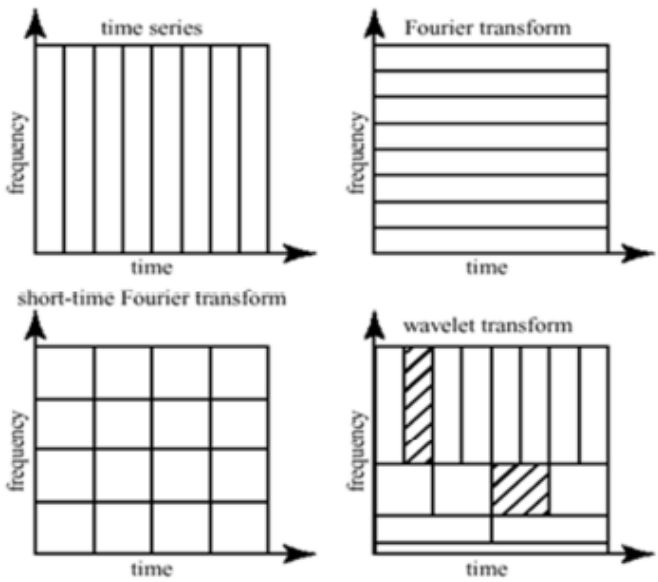


Figure 3.2: Comparison between Different Techniques [11]

The parameter scale in the wavelet analysis is similar to the scale used in maps. As in the case of maps, high scales correspond to a non-detailed global view, and low scales correspond to a detailed view. Similarly, in terms of frequency, low frequencies (high scales) correspond to a global information of a signal (that usually spans the entire signal), whereas high frequencies (low scales) correspond to a detailed information of a hidden pattern in the signal that usually lasts a relatively short time [11].

Scaling, as a mathematical operation, either dilates or compresses a signal. Larger scales correspond to dilated (or stretched out) signals and small scales correspond to compressed signals. As an example for CWT, Figure 3.3 illustrates an input waveform, while Figure 3.4 demonstrates how the CWT extracts the frequency component along with their time allocation.

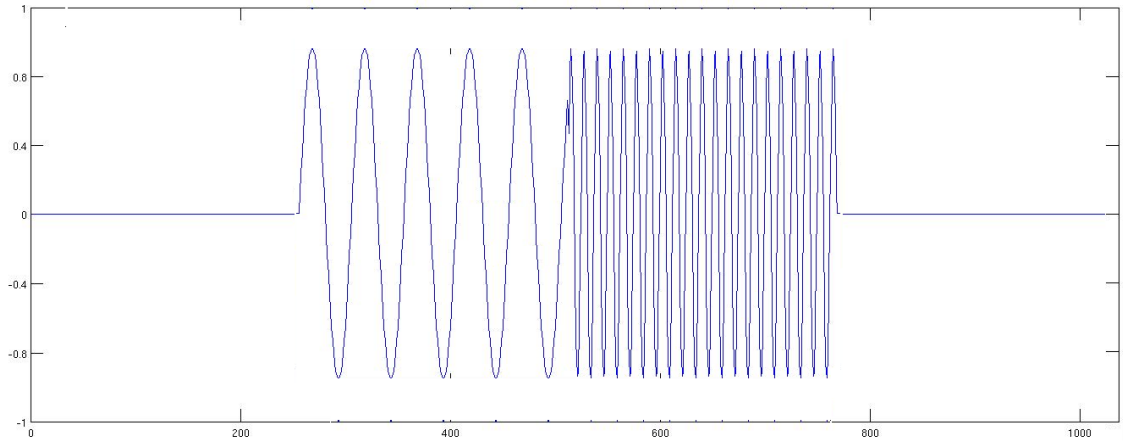


Figure 3.3: Dual Frequency Sinusoidal Waveform

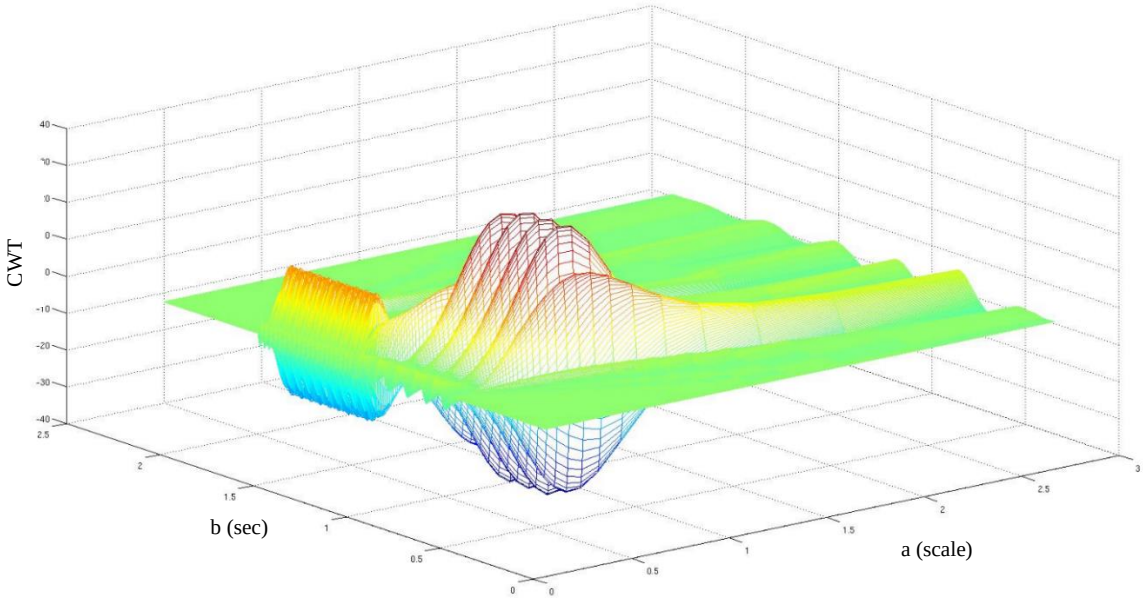


Figure 3.4: CWT of the Dual Frequency Sinusoidal Waveform

### 3.1.3.2 Discrete wavelet Transform (DWT)

Filters of different cutoff frequencies are used to analyze the signal – in its discrete form – at different scales. The signal is passed through a series of High Pass Filters (HPF) and Low Pass Filters (LPF) to analyze both the high frequencies, and the low frequencies of a certain signal [12].

The procedure starts by passing the desired signal through a half band digital low pass filter with impulse response h[n]. Filtering a signal corresponds to the mathematical operation of convolution of the signal with the impulse response of the filter. The convolution operation in discrete time is defined as in equation 3.5 [11]:

$$x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k] \,.\, h[n-k] \tag{3.5}$$

After passing the signal through a half band LPF, half of the samples can be eliminated according to the Nyquist rule. Simply discarding every other sample will subsample the signal by two, and the signal will then have half the number of points. The scale of the signal is now doubled [12].

The process of filtering the lower half of the band removes the high frequency information, but leaves the scale unchanged. Only the subsampling process changes the scale. Resolution, on the other hand, is related to the amount of information in the signal, and therefore, it is affected by the filtering operations [12].

Half band low pass filtering removes half of the frequencies, which can be interpreted as losing half of the information which reduces the resolution by half; however, the subsampling operation does not affect the resolution, since removing half of the spectral components from the signal makes half the number of samples redundant anyway. Half the samples can be discarded without any loss of information [12].

In summary, the low pass filtering halves the resolution, but leaves the scale unchanged. The signal is then subsampled by 2 since half of the number of samples are redundant. This doubles the scale. This procedure can mathematically be expressed as in equations 3.6 and 3.7 where h(n) and g(n) represents the low pass and high pass filter coefficients respectively [12].

$$Y_{high}[k] = \sum_{n} x[n] \,.\, g[2k - n] \tag{3.6}$$

$$Y_{low}[k] = \sum_{n} x[n] \,.\, h[2k - n] \tag{3.7}$$

The DWT analyzes the signal at different frequency bands with different resolutions by decomposing the signal into a coarse approximation and detail information. DWT employs two sets of functions, called scaling functions and wavelet functions, which are associated with low pass and high pass filters, respectively. The decomposition of the signal into different frequency bands is simply obtained by successive high pass and low pass filtering of the time domain signal. The original signal x[n] is first passed through a half band high pass filter g[n] and a low pass filter h[n] [12].

After the filtering, half of the samples can be eliminated. The signal can therefore be subsampled by 2, simply by discarding every other sample. This constitutes one level of decomposition. If more than one level of decomposition is needed the half band high pass and low pass filters are reused in order to further decompose the signal. Figure 3.5 illustrates the DWT multi-level decomposition of a signal x(n), where the low pass output at every level represents the details, and the final level high pass output represents the approximation [12].

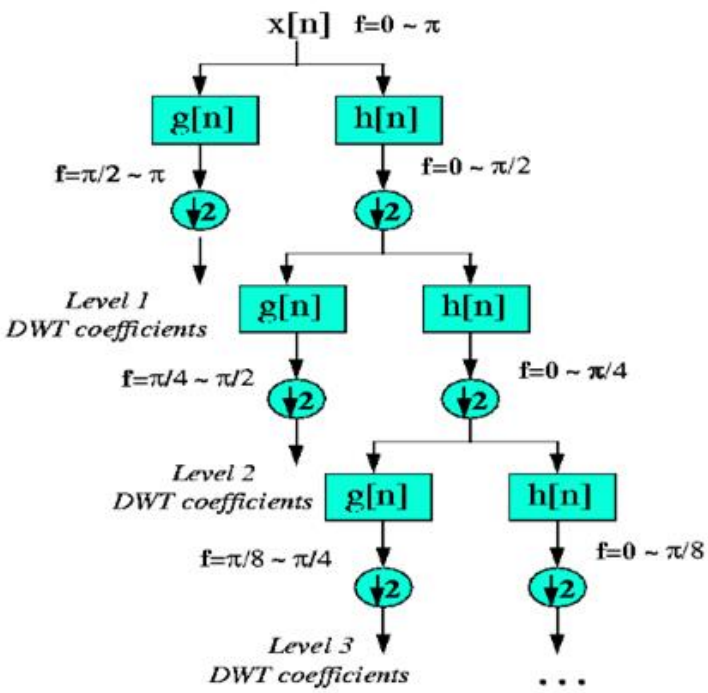


Figure 3.5: Discrete Wavelet Transform Decomposition by Filter Banks [12]

As an example to the DWT analysis. The wave form illustrated in Figure 3.6, with added noise, is processed using the DWT analysis by applying 4 level of decomposition filters. The filter responses and the output of each stage of decomposition is shown in Figure 3.7. As we can see the higher frequency components, noise, are separated from the lower frequency component which is the original signal.

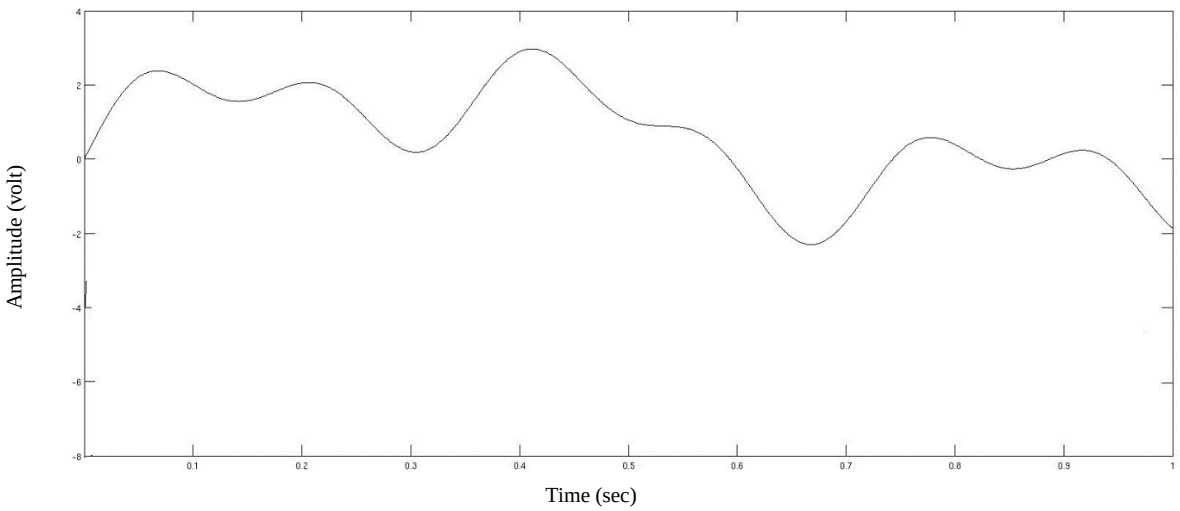


Figure 3.6: Input Waveform for DWT

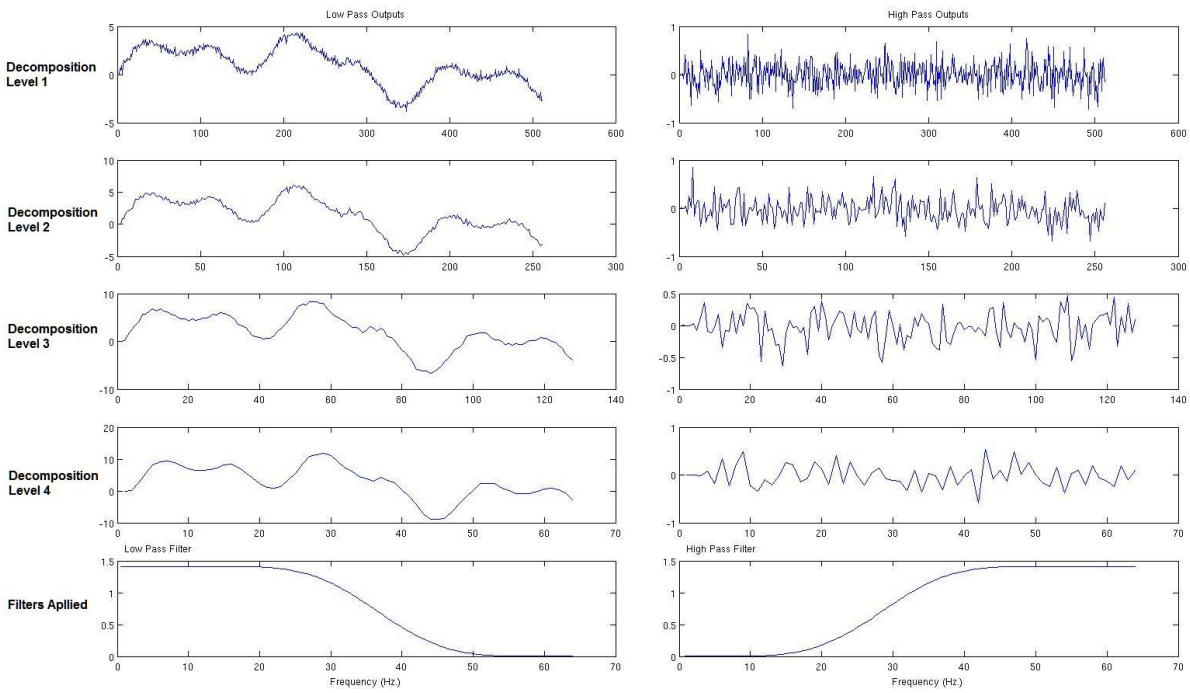


Figure 3.7: Signals Generated by the Analysis Filters

#### 3.1.3.3 Wavelet Packet Transform (WPT)

In the DWT, each level is calculated by passing the previous approximation coefficients through a high and low pass filter; however in the WPT both the details and approximation coefficients are decomposed as illustrated in Figure 3.8. For n levels decomposition the WPT produces $2^n$ different sets of coefficients as opposed to n+1 sets for DWT [12].

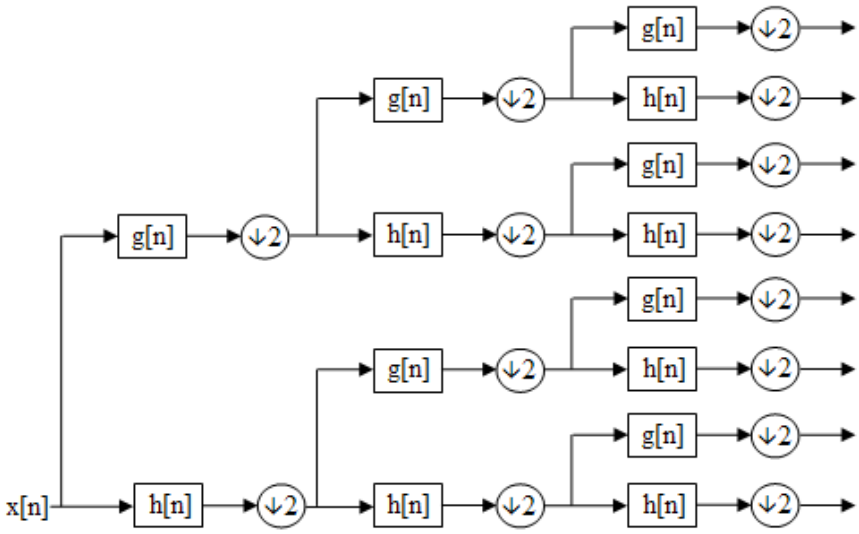


Figure 3.8: Wavelet Packet Transform Decomposition Tree

### 3.2 Support Vector Machines

Support Vector Machines (SVMs) are a set of supervised learning methods used for classification and regression detection. SVMs are effective in high dimensional spaces even when number of dimensions is greater than the number of samples, memory efficient, and versatile. There are different types of SVM that will be discussed in the next sub-sections.

#### 3.2.1 Linearly Separable Binary Classification

In a binary classification, it is desired to discriminate between the two classes labeled (+1,-1). In this case, the aim of the SVM is to orientate a hyper-plane in such a way to be as far as possible from the closet member of both classes. This hyper-plane can be described by equation 3.8. Where w is normal to the hyper-plane, and $b/\|w\|$ is the perpendicular distance from the hyper-plane to the origin [13].

$$w \cdot x + b = 0 \qquad (3.8)$$

Implementing a SVM is simply performed by selecting the variables w and b so that our classes can be described by equations 3.9 and 3.10.

$$x_i \cdot w + b \geq +1 \qquad \text{for } y_i = +1 \qquad (3.9)$$

$$x_i \cdot w + b \geq -1 \qquad \text{for } y_i = -1 \qquad (3.10)$$

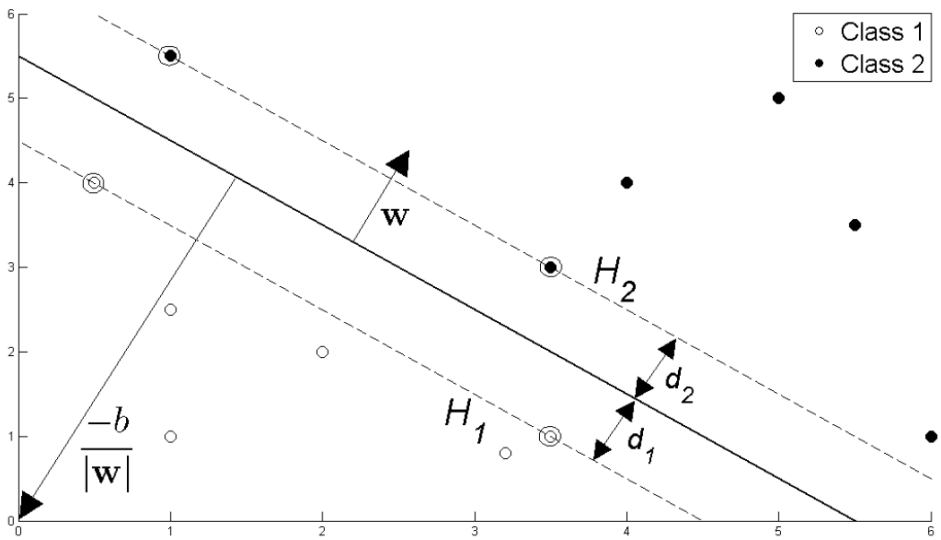


Figure 3.9: Hyper-plane through Two Linearly Separable Classes [13]

If we now just consider the points that lie closest to the separating hyper-plane, the Support Vectors shown in circles in Figure 3.9, then the two planes H1 and H2 that these points lie on can be described by equation 3.11 and 3.12

$$x_i \cdot w + b = +1 \qquad \text{for } H_1 \qquad (3.11)$$

$$x_i \cdot w + b = -1 \qquad \text{for } H_2 \qquad (3.12)$$

Referring to Figure 3.9, we define $d_1$ as being the distance from H1 to the hyper-plane and $d_2$ from H2 to it. The hyper-plane's equidistance from H1 and H2 means that d1 = d2 which is a quantity known as the SVM's margin. In order to orientate the hyper-plane to be as far from the Support Vectors as possible, we need to maximize this margin.

Simple vector geometry shows that the margin is equal to $1/\|w\|$ and maximizing it is equivalent to finding $\min 1/2\|w\|^2$.

After constructing the hyper-plane separating the two classes, the classification as simple as comparing the new data with the hyper-plane to determine on which side they are and hence knowing the class of the new data.

### 3.2.2 Binary Classification for Data that is not Fully Linearly Separable

In order to extend the SVM methodology to handle data that is not fully linearly separable, we relax the constraints for equations 3.9 and 3.10 slightly to allow for misclassified points. This is done by introducing a positive slack variable $\varepsilon_i$, $i = 1, 2, .... L$ as formulated in equations 3.13 and 3.14 [13].

$$x_i \cdot w + b \geq +1 - \xi i \qquad \text{for } y_i = +1 \tag{3.13}$$

$$x_i \cdot w + b \leq -1 - \xi i \qquad \text{for } y_i = -1 \tag{3.14}$$

In this soft margin SVM illustrated in Figure 3.10, data points on the incorrect side of the margin boundary have a penalty that increases with the distance from it. As we are trying to reduce the number of misclassifications, a sensible way to adapt our objective function is to find $\min 1/2\|w\|^2 + C\sum_{i=1}^{L}\xi_i$. Where the parameter C controls the trade-off between the slack variable penalty and the size of the margin.

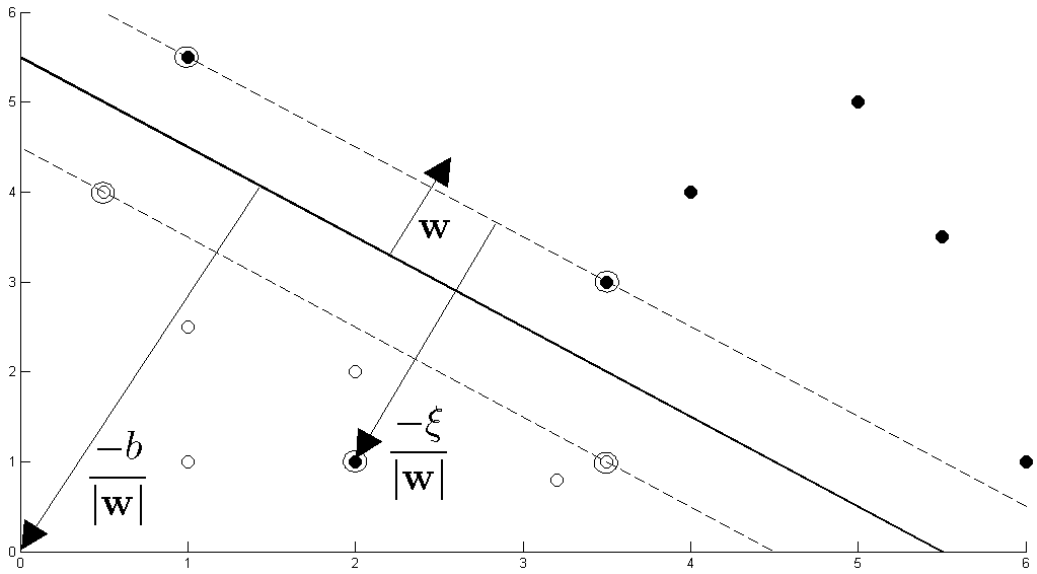


Figure 3.10: Hyper-plane through Two Non-linear Separable Classes [13]

### 3.2.3 Nonlinear Binary Support Vector Machines

In some cases, such as the example in Figure 3.11, no simple linear discriminant function will define the given classes. A quadratic function such as the circle pictured is needed. To change from a linear to nonlinear classifier, one must only substitute a kernel evaluation in the objective instead of the original dot product [13, 14].

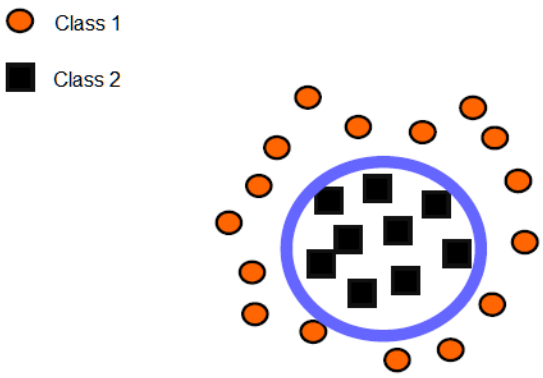


Figure 3.11: Example on Non-linear Classification

Thus by changing kernels we can get different highly nonlinear classifiers. No algorithmic changes are required from the linear case other than substitution of a kernel evaluation for the simple dot product. All the benefits of the original linear SVM method are maintained. We can train a highly nonlinear classification function such as a polynomial or a radial basis function machine, or a sigmoidal neural network using robust, efficient algorithms that have no problems with local minima. By using kernel substitution a linear algorithm that – only capable of handling separable data – can be turned into a general nonlinear algorithm. Some of the more popular known kernels are given in Table 3.1 where $x_i$ and $x_j$ are the training set and the label set respectively. New kernels are being developed to fit domain specific requirements [15].

Table 3.1: Popular Kernel Functions [15]

| Kernel Function Name | Kernel Function Equation | |
|---|---|---|
| Radial Basis Function | $K(x_i, x_j) = \exp\left(-\gamma \left\| x_i - x_j \right\|^2\right)$ | (3.15) |
| Sigmoidal Kernel | $K(x_i, x_j) = \tanh\left(a x_i \cdot x_j - b\right)$ | (3.16) |
| Polynomial Kernel | $K(x_i, x_j) = \left(x_i \cdot x_j + a\right)^b$ | (3.17) |

### 3.2.4 Multi-class Support Vector Machines

SVMs were originally designed for binary classification. However in certain application a multi-class SVM which discriminates between more than two classes is needed. The formulation to solve a multi-class SVM problem has variables proportional to the number of classes. Therefore, for multi-class SVMs, either several binary classifiers have to be constructed or a large optimization problem is needed [16].

There two different methods to solve a multi-class SVM by constructing several binary classifiers. Either solve it as “one against all” or “one against one”. Further explanation of these methods is discussed in the following sub-sections.

#### 3.2.4.1 One Against All Method

The one-against-all method is one of the earliest implementations for SVM multi-class classification [15]. In this method *k* SVM models are constructed, where *k* is the number of classes. The $i^{\text{th}}$ SVM is trained with all the examples in the $i^{\text{th}}$ class with positive label (+1), and all other examples with negative labels (-1). Thus given *l* training data $(x_1, y_1), \ldots, (x_l, y_l)$, where $x_i \in R^n$, $i = 1, \ldots, l$ and $y_i \in \{1, \ldots, k\}$ is the class of $x_i$, the $i^{th}$ SVM solves the problem in equation 3.18, where the training data $x_i$ are mapped to a higher dimensional space by the function $\phi$ and *C* is the penalty parameter [16, 17].

$$\begin{aligned} \min_{w^i, b^i, \xi^i} \quad & \frac{1}{2}\left(w^i\right)^T w_i + C\sum_{j=1}^{l} \xi_j^i \\ & \left(w^i\right)^T \phi\left(x_j\right) + b^i \geq 1 - \xi_j^i, \quad \text{if } y_j = i, \\ & \left(w^i\right)^T \phi\left(x_j\right) + b^i \leq -1 + \xi_j^i, \quad \text{if } y_j \neq i, \\ & \xi_j^i \geq 0, \; j = 1, \ldots, \; l \end{aligned} \tag{3.18}$$

Minimizing $1/2\left(w^i\right)^T w^i$ means that we would like to maximize the margin between the two groups of data. When data are not linear separable, there is a penalty term $C\sum_{J=1}^{l} \xi_j^i$.

After solving equation 3.18, there are k decision functions. Finally *x* is in the class which has the largest value of the decision function in equation 3.19

$$\text{class of } x \equiv \text{argmax}_{i=1,\ldots,k}\left(\left(w^i\right)^T \phi(x) + b^i\right) \tag{3.19}$$

### 3.2.4.2 One Against One Method

Another major method is called the one-against-one method introduced in [18]. This method constructs $k \cdot (k-1)/2$ binary classifiers where each one is trained on data from two classes. For training data from the $i^{th}$ and the $j^{th}$ classes, we solve the binary classification problem formulated in equation 3.20.

$$\begin{aligned} \min_{w^{ij}, b^{ij}, \xi^{ij}} \quad & \frac{1}{2}\left(w^{ij}\right)^T w^{ij} + C\sum_{j=1}^{l} \xi_t^{ij} \\ & \left(w^{ij}\right)^T \phi(x_t) + b^{ij} \geq 1 - \xi_t^{ij}, \quad \text{if } y_t = i, \\ & \left(w^{ij}\right)^T \phi(x_t) + b^{ij} \leq -1 + \xi_t^{ij}, \quad \text{if } y_t = j, \\ & \xi_j^{ij} \geq 0. \end{aligned} \tag{3.20}$$

Once the classification problem in equation 3.20 is solved, the voting function formulated in equation 3.21 is applied to determine which class *x* belongs to.

$$\text{sign}\left(\left(w^{ij}\right)^T \phi(x) + b^{ij}\right) \tag{3.21}$$

If *x* is in the $i^{th}$ class then the vote for the $i^{th}$ class is added by one, otherwise the $j^{th}$ is increased by one. Then we predict x is in the class with the largest vote. This method is called “Max Wins” strategy. In case two classes have identical votes, we select the class with the smaller index [19].

# Chapter 4

# Proposed Design and Software Simulation

In this chapter a detailed description of the hardware proposed design and its software simulation are discussed. A detailed description of the concept this system is designed upon is discussed in this section. All the proposed blocks are simulated using MATLAB 2014A.

## 4.1 Proposed Design

A system which processes analog signals consists of a digital to analog conversion unit, a digital signal processing unit, and finally an output interface unit with an optional memory for data logging. The scope of this work focuses on the Data Processing Unit, which is composed of a Feature Extraction Unit and a Classifier Unit. Figure 4.1, illustrates the proposed block diagram for this system.

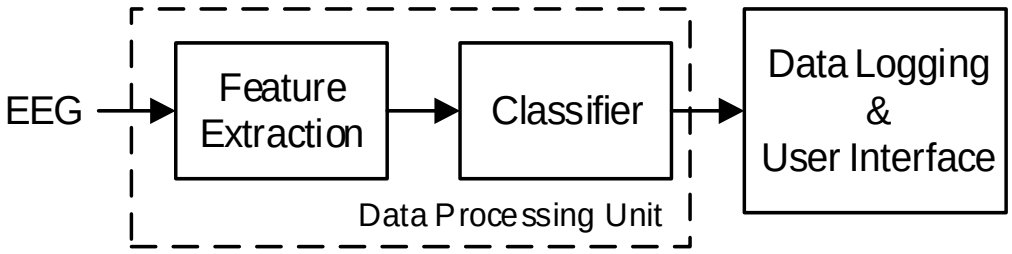


Figure 4.1: Functional Block Diagram of the Proposed Design

For all intelligent systems, there are two modes of operation, the training mode and the testing mode. In the training mode the system learns how to properly classify any given data to its corresponding output. In the testing mode the system applies its learned data to obtain an accurate classification.

As it is illustrated in Figure 4.2, these two modes of operation are similar; however the learning process is an offline task that is performed on a computer, it requires the input EEG stream along with its pre-scored values. In the current work, only the testing mode is implemented.

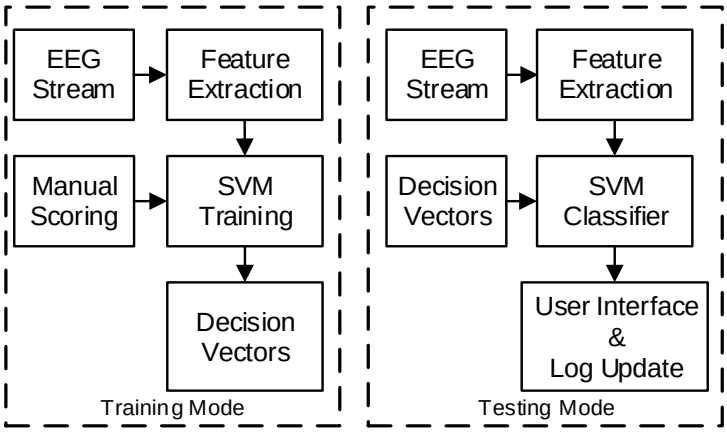


Figure 4.2: Modes of Operation Flow Chart

### 4.1.1 EEG Database Source

The Data used in the training and testing simulations conducted using MATLAB script, are obtained from an open source database "PhysioBank" [26]. This database includes PSG recordings for different types of diseases including Narcolepsy, along with recordings for a number of healthy subjects.

These PSG recordings are written using the European Data Format (EDF). The EDF file is a simple format for exchange and storage of multichannel biological and physical signals. EDF was first introduced in 1992, since then it became the de-facto standard for EEG and PSG recordings in commercial equipment. A MATLAB script is written in order to properly load the EDF file desired channels and prepare the data in arrays for training and testing.

Along with the PSG recordings, the PhysioBank database also supply scoring files that are examined by physicians [26]. These scoring files are imported to MATLAB and used for training and testing of the classifier.

A total number of 5 narcoleptic and 5 healthy patients PSG recordings are used for the training of the classifier unit. These patient files hold more than 80 hours of recorded data, which form about 9600 epoch.

And a total number of 2803 epoch 30 seconds each, are used for testing of the classifier. The different classes occurrence rate within these epochs are distributed as illustrated in **Error! eference source not found.**. Ensuring a balanced occurrence rate for the different classes as

much as possible provides a more accurate and reliable testing of the classifier to ensure its effectiveness for the different classes.

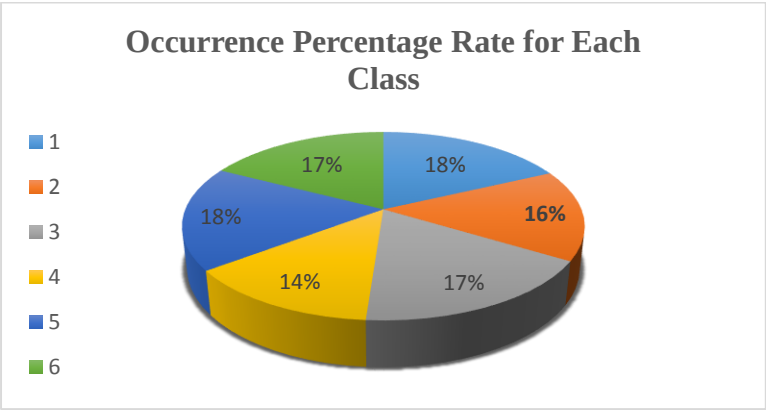


Figure 4.3: Occurrence Rate for Each Class in the Testing Epochs

### 4.1.2 Feature Extraction Unit

Feature selection is crucial to the classification process, since selection of wrong features, leads to classification error. Due to the localization of the features in both time and frequency, Fast Fourier Transform (FFT) is not suitable for processing of the EEG spectrum. Wavelet Transform (WT) is reported to be the most suitable algorithm as it specifies the presence of each frequency as well as its time allocation [5, 6, 7, 8, 11, 20, 21, 22, 23, and 24].

As previously discussed in chapter 2, there are four main frequency bands – Alpha, Beta, Theta, and Delta – that give precise description to the human alertness and state of mind. However; the beta band, being wider in respect to the rest, is often divided into two sub-bands beta1 and beta2. Moreover the sleep spindles located in the frequency range between Alpha and Beta bands gives better classification when taken into consideration. The Feature Extraction Unit illustrated in Figure 4.4 consists of two operations, extracting the Wavelet Packet Transform (WPT) coefficients, and calculating the desired features for each epoch. The RAM Unit is implemented to store the features extracted from one epoch, while the Feature Extraction Unit starts working on the next one. The following subsections explain these operations in details.

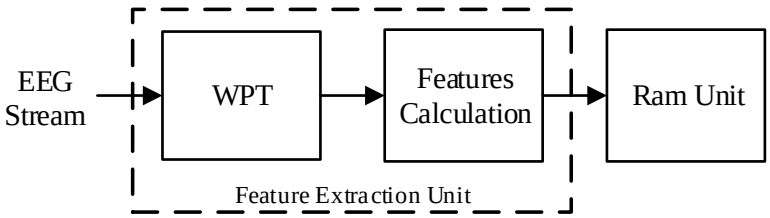


Figure 4.4: Block Diagram of the Proposed Feature Extraction Unit

#### 4.1.2.1 The Wavelet Packet Transform Unit

The WPT unit provides multi-resolution and time-frequency analysis for the EEG signal. A full decomposition tree is generated using the WPT by repeatedly applying a pair of High Pass (HPF) and a Low Pass (LPF) filters followed by a decimation by 2 [11]. Since the filters applied should be orthogonal, Debauchee second order (DB2) filter are implemented in this design, Table 4.1 represents both the HPF and LPF coefficients, while Figure 4.5 illustrate their response.

Table 4.1: 2nd Order Debauchee Filter Coefficients

| | h(0) | h(1) |
|---|---|---|
| LPF Coefficients | 0.7071 | 0.7071 |
| HPF Coefficients | 0.7071 | -0.7071 |

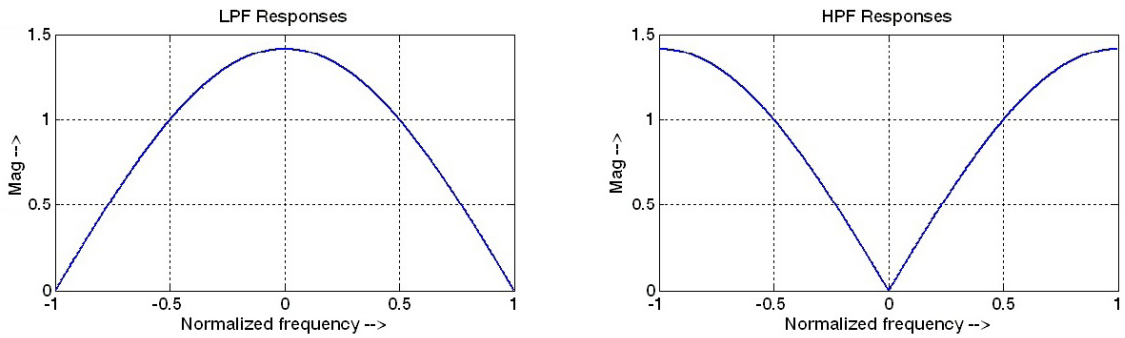


Figure 4.5: 2nd Order Debauchee Filter Response

Figure 4.6 illustrates the WPT tree adopted in this design [6], while Table 4.2 summarizes the six desired frequency bands – S1 to S6 – along with their respective coefficients.

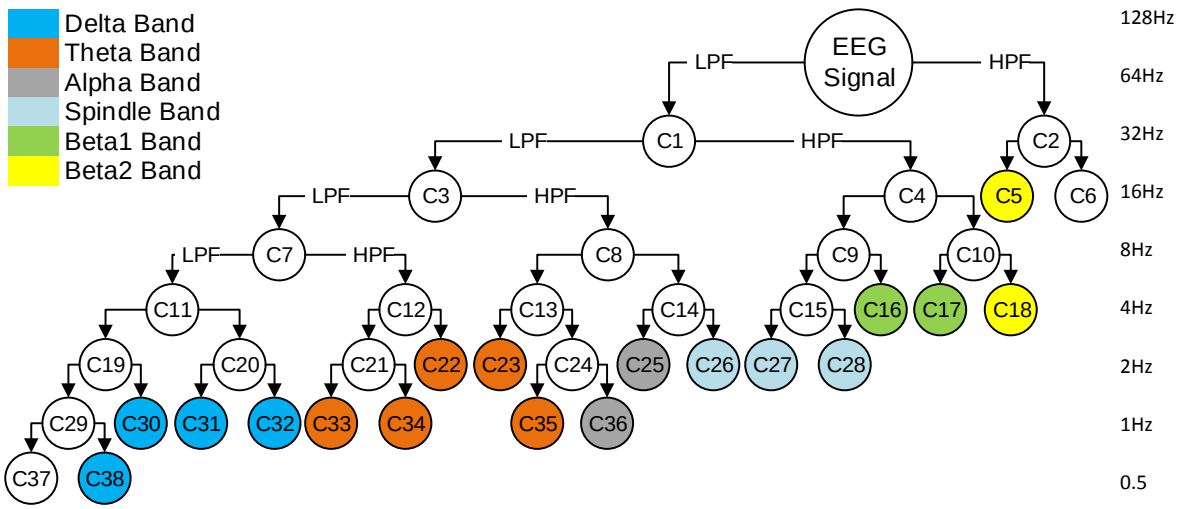


Figure 4.6: Proposed Wavelet Packet Tree

Table 4.2: Desired Frequency Bands Coefficients

| Frequency Band | Frequency Range | Coefficients | Label |
|---|---|---|---|
| 1- Delta | 00.39 – 03.13 Hz | [C38, C30, C31, C32] | S1 |
| 2- Theta | 03.13 – 08.46 Hz | [C33, C34, C22, C23, C35] | S2 |
| 3- Alpha | 08.46 – 10.93 Hz | [C36, C25] | S3 |
| 4- Spindle | 10.93 – 15.63 Hz | [C26, C27, C28] | S4 |
| 5- Beta1 | 15.63 – 21.88 Hz | [C16, C17] | S5 |
| 6- Beta2 | 21.88 – 37.50 Hz | [C18, C5] | S6 |

Figure 4.7 illustrates 7 plots, the first one belonging to one complete epoch, while the rest represent the coefficients belonging to each group of frequencies S1 to S6 respectively. These plots are the MATLAB simulation results of the WPT unit. An epoch lasts 30 seconds in time, and at the sampling rate of 128Hz, an epoch consists of 3840 sample.

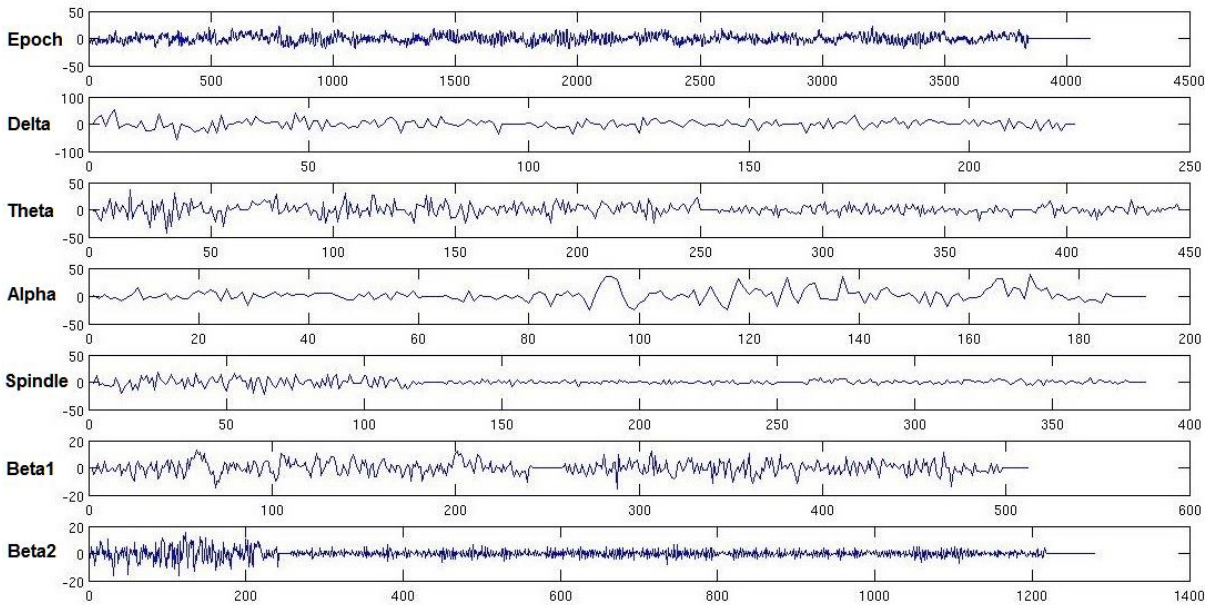


Figure 4.7: An EEG Epoch Filtered to 6 Frequency Bands S1 to S6

The coefficient C5 consists of 960 sample, C16 to C18 consists of 240 sample, C22 to C28 consists of 120 sample, C30 to C36 consists of 60 sample, and finally C38 consists of 30 sample.

The first 5 data points of each coefficient are illustrated in Table 4.3 According to the same color code used in Figure 4.6 to easily identify which band these coefficients is belonging to.

These samples are extracted by MATLAB software simulation to form a reference for the hardware implementation simulation discussed in the next chapter.

Table 4.3: Sample of Coefficients Extracted from an Epoch by MATLAB Simulation

| Coefficients | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| C5 | 3.945360195 | -5.49450549 | 0.534188034 | 4.655067155 | 6.349206349 |
| C16 | 7.726648351 | -0.19078144 | 3.021978021 | -7.60836385 | -10.8363858 |
| C17 | 8.382936507 | 4.250610500 | -13.6141636 | -4.08272283 | 8.257020757 |
| C18 | 16.86889499 | 7.822039072 | -13.3547008 | 2.327533577 | -9.66117216 |
| C22 | 24.99210208 | 57.73302390 | 24.22315583 | -48.9589425 | 36.38599683 |
| C23 | 10.69239993 | -1.62423032 | 19.50695218 | -1.73215260 | 6.070628272 |
| C25 | 12.13855848 | -19.9170568 | -20.7264739 | 29.25233410 | 12.62151069 |
| C26 | -32.3793821 | 29.26312633 | -10.1177137 | -14.1000459 | -16.5498816 |
| C27 | -12.5000981 | 2.703453124 | 7.916099267 | -2.98405105 | 2.045127213 |
| C28 | -4.67573279 | 17.17583092 | 3.523662455 | -24.2339480 | -1.67819146 |
| C30 | -51.0588369 | -68.6507936 | 0.259462759 | 146.0470085 | -86.3362332 |
| C31 | -13.4291056 | -17.9105616 | 47.95482295 | -53.1669719 | -19.9290293 |
| C32 | 29.60355616 | 15.53724053 | 33.34096459 | 150.4807692 | -16.5102258 |
| C33 | -23.5252594 | -14.1025641 | -24.6642246 | 12.59920634 | -22.7983821 |
| C34 | -12.5133547 | 4.632173382 | 16.76587301 | 60.12667887 | -22.4473443 |
| C35 | 12.85294566 | -23.0845543 | -13.0341880 | 14.88858363 | -14.1903235 |
| C36 | 6.526633089 | 12.56105006 | 22.70299145 | -23.4050671 | -13.3814102 |
| C38 | -95.6339797 | 207.777370 | -0.54500751 | -186.314326 | 82.9085938 |

#### 4.1.2.2 The Features Calculation Unit

Once all the coefficients desired for each band which are forming one complete epoch are extracted by the WPT discussed in the previous subsection. A total number of 22 features are calculated as follows [6]:

a) **Group 1: The energy of each band**

This group consists of 6 features, corresponding to the energy in each band S1 to S6. The energy of each band is formulated according to equation 4.1, where x(n) represents the coefficients within the band, while N represents the total number of coefficients within the band.

$$E_{1-6} = \sum_{n=1}^{N} x(n)^2 \tag{4.1}$$

b) **Group 2: The total energy in all bands**

The total energy in all bands, is simply the summation of the 6 different energies extracted in group 1. And can be formulated according to equation 4.2

$$E_7 = E_1 + E_2 + E_3 + E_4 + E_5 + E_6 \tag{4.2}$$

c) **Group 3: Ratio of different energy values**

Features 8, 9 and 10 are calculated according to equations 4.3, 4.4, and 4.5 respectively. They represent the energy ratio between Delta, Theta and Alpha bands.

$$E_8 = E_3 / (E_1 + E_2) \tag{4.3}$$

$$E_9 = E_1 / (E_2 + E_3) \tag{4.4}$$

$$E_{10} = E_1 / (E_2 + E_3) \tag{4.5}$$

d) **Group 4: The mean of the absolute values of each band**

This group represents 6 features corresponding to the mean of the absolute values for each band. These features are calculated according to equation 4.6.

$$E_{11-17} = \sum_{n=1}^{N} |x(n)| \tag{4.6}$$

e) **Group 5: The standard deviation for each band**

Group 5 represents the last 6 features to be calculated. The standard deviation of each band is calculated according to equation 4.7.

$$E_{18-22} = \sqrt{1/N \cdot \sum_{n=1}^{N} \left( x(n) - \text{average}[x(n)] \right)^2} \tag{4.7}$$

### 4.1.2.3 Features Normalization

Following the extraction of the features, normalization is applied for each feature set in order to maintain the same weight for all features. Linear normalization is applied to normalize the features from 0 to 1. Figure 4.8 illustrates the MATLAB simulation flowchart followed to simulate the feature extraction unit. Whereas Table 4.4 presents the normalized features extracted from the epoch previously illustrated in Figure 4.7. These values are used as reference for the hardware implementation testing that will be conducted in the next chapter.

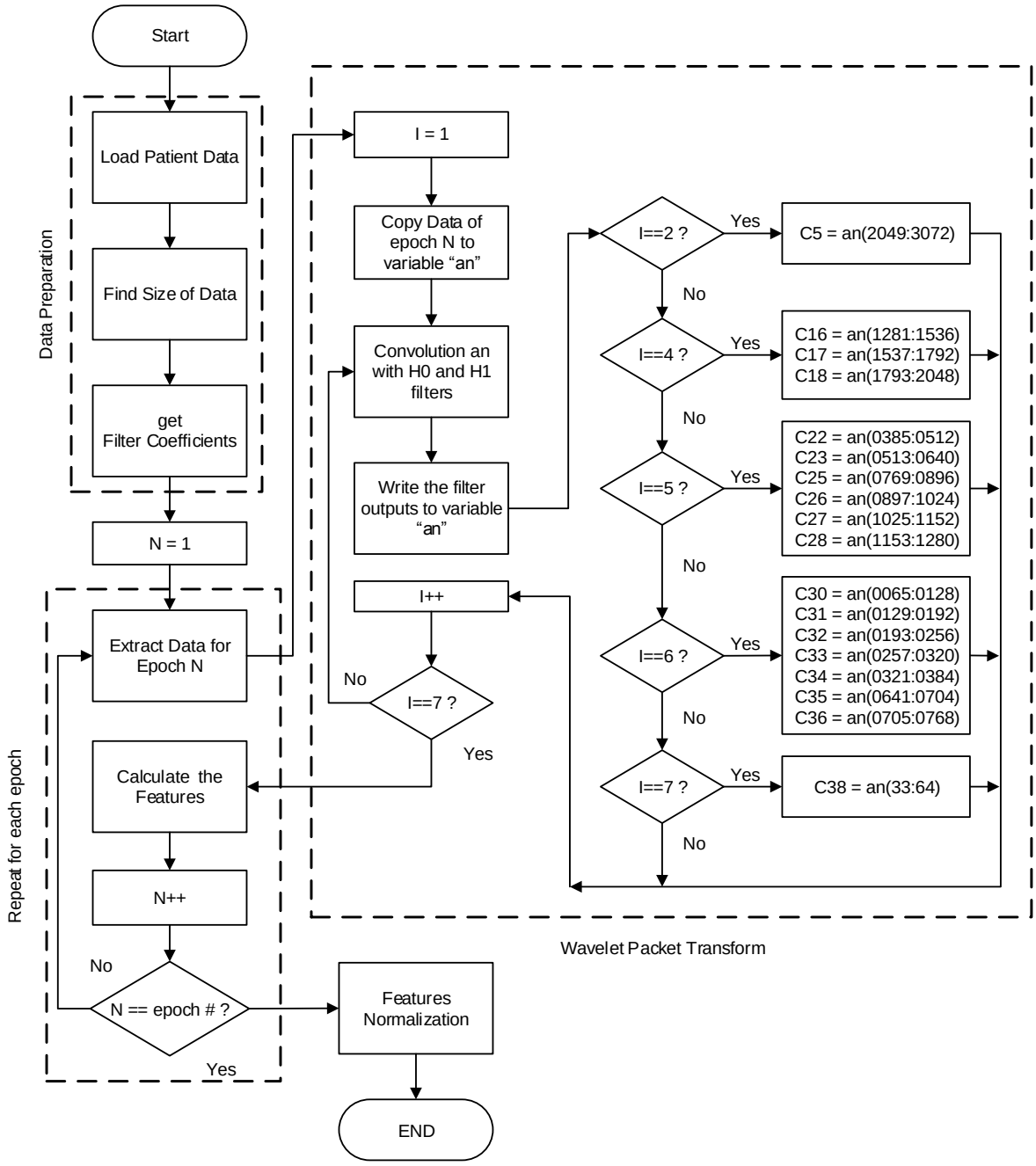


Figure 4.8: Software Simulation Script Flowchart

Table 4.4: Normalized Values of the Features Extracted for an Epoch

| Features | Normalized Values | | | | | |
|---|---|---|---|---|---|---|
| Energy of Each Band | 0.007755 | 0.009969 | 0.042676 | 0.053927 | 0.069998 | 0.050969 |
| Total Energy | 0.010928 | | | | | |
| Ratio of Energies | 0.191466 | 0.081678 | 0.147705 | | | |
| Mean of the Absolute Values of Each Band | 0.096607 | 0.111102 | 0.111592 | 0.151424 | 0.202210 | 0.160998 |
| Standard Deviation for Each Band | 0.065105 | 0.091235 | 0.136674 | 0.172973 | 0.218768 | 0.178965 |

### 4.1.3 Classifier Unit

The classifier algorithm used in this design is the Support Vector Machine (SVM). The SVM algorithm is chosen for the following reasons SVM algorithm proves to be the most effective classifier that achieves high accuracy rates compared to other classification techniques. In addition to its high accuracy rate, the SVM algorithm proves to be simple for hardware implementation [15, 25].

The classification problem that will be solved in this design consists of differentiating between six different classes using a 22 feature vector. Therefore a multi-class none linear SVM is chosen for this design. The following sections will describe the database used for training and testing, the process by which the kernel function is chosen, the cross-validation and grid search for SVM parameter selection, and finally the flowchart of the proposed SVM simulation along with the results extracted by this simulation.

#### 4.1.3.1 Kernel Selection

The SVM essentially maps the feature to a higher dimensional space and applies an optimal hyper-plane in the mapped space, which provides a better discrimination among classes [15]. The Radial Basis Function (RBF) kernel is chosen for implementation for the following reasons. The RBF kernel nonlinearly maps the samples into a higher dimensional space so it can handle the case when the relation between class labels and attributes is nonlinear, unlike the linear kernel. Furthermore, the linear kernel is a special case of RBF. Another reason is the complexity of the system, where the RBF kernel uses less hyper-parameters.

After testing different kernels – Linear, Poly, Radial Basis Function (RBF), and Sigmoid – the RBF kernel provides the highest accuracy results – as illustrated in Table 4.5 – and is selected

for the hardware implementation. The RBF kernel function is formulated in Table 3.1 where $x_i$ is the training set, $x_j$ is the label set, and $\gamma$ is the kernel parameter [15].

#### 4.1.3.2 Cross-Validation and Grid-Search

There are two parameters for the RBF kernel: C and $\gamma$. It is not known beforehand which C and $\gamma$ are best for a given problem; consequently some kind of model selection (parameter search) must be done. The goal is to identify good (C, $\gamma$) so that the classifier can accurately predict unknown data (i.e. testing data) [15].

A script is provided by the LIBSVM tool [15] to perform a "grid-search" on the kernel parameters – C and $\gamma$ – using cross-validation. Various pairs of (C, $\gamma$) values are tried and the one with the best cross-validation accuracy is picked. Figure 4.9 illustrates the tests conducted to determine the most suitable parameters. The best (C, $\gamma$) resulted in an accuracy rate 92.14%.

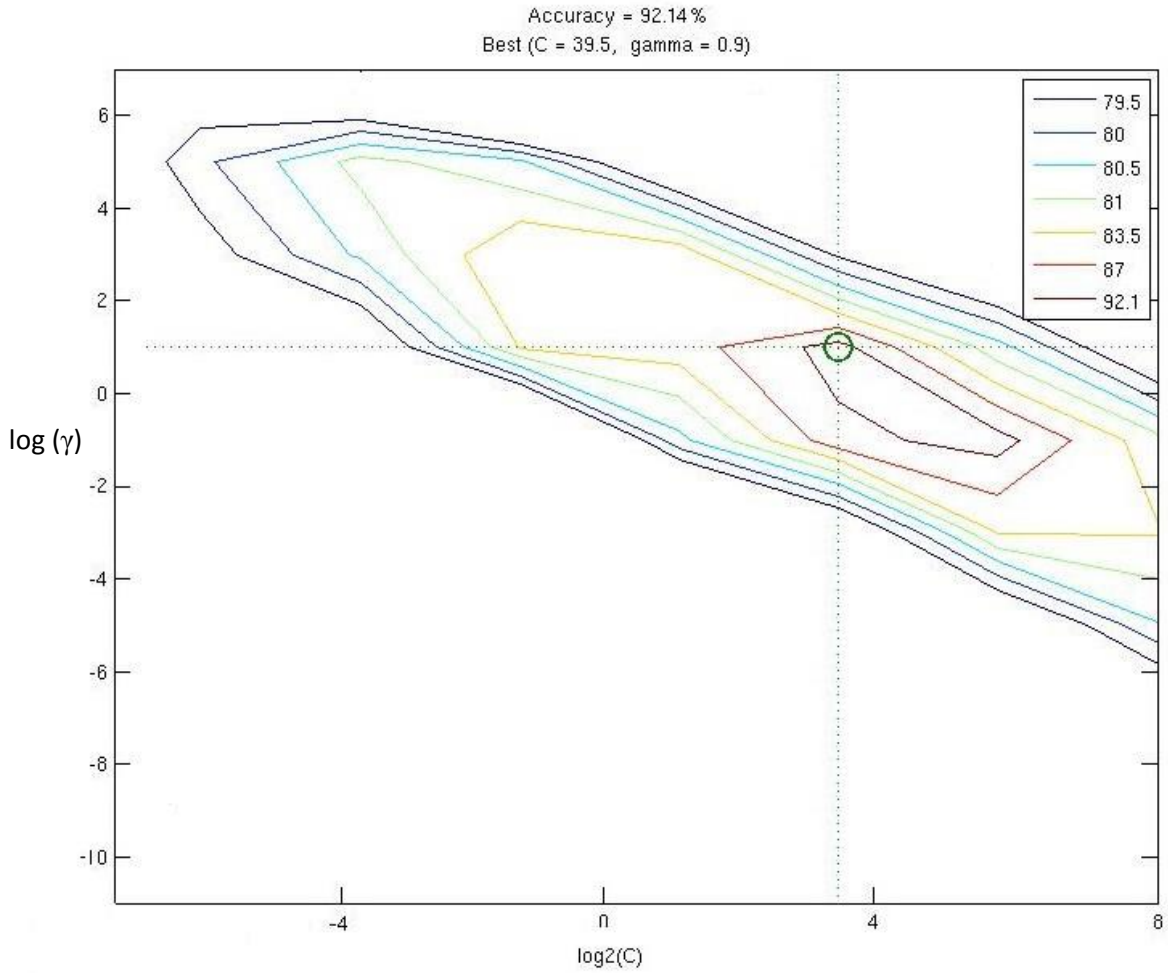


Figure 4.9: SVM Parameter Selection

#### 4.1.3.3 Support Vector Machine Classification and Prediction Script

The SVM algorithm provided by LIBSVM tool [15] is a general purpose tool, which includes different type of kernels and different type of usage options which are not used. Since the hardware complexity is one of the important factors in our design, an optimized script is written using MATLAB which will be followed later in the implementation.

The written script is composed of two sections. The first section includes the comparison of the current epoch features with the support vectors, and applying this comparison to the kernel function, whereas the second section is the voting section responsible for determining the most suitable class for this epoch. Figure 4.10 illustrates the flowchart of this script.

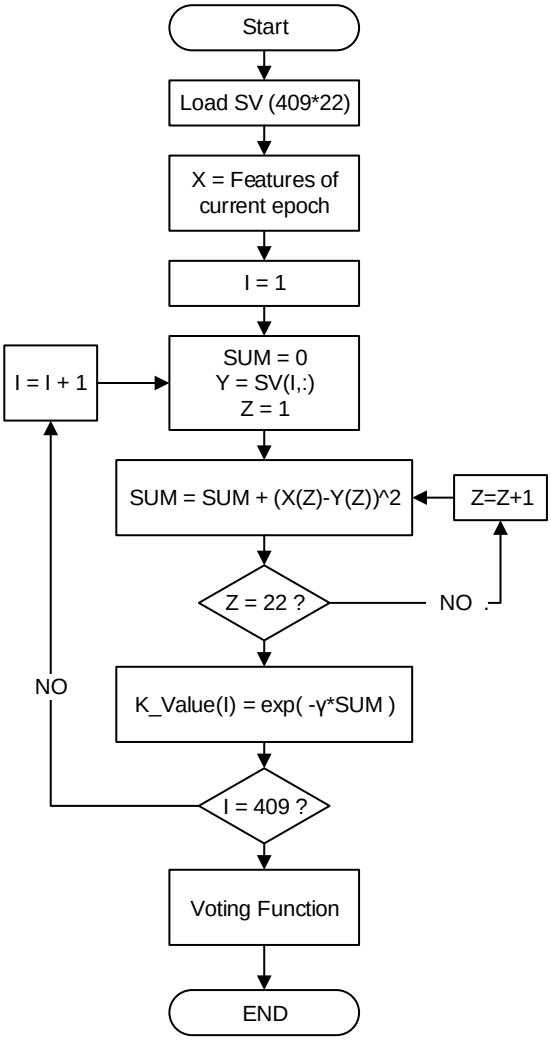


Figure 4.10: SVM Prediction Flowchart

A series of tests are conducted on the SVM classifier using the different kernels provided in the LIBSVM toolbox [15]. Table 4.5 demonstrates the results of these tests for the different kernels, where the RBF kernel provided the highest accuracy result at 92.14%.

Table 4.5: Different Kernel Accuracy Results

| Kernel | Accuracy |
|---|---|
| Linear | 85.9649% |
| Polynomial Kernel | 69.1729% |
| Radial Basis Function | 92.14% |
| Sigmoidal Kernel | 66.9173% |

# Chapter 5

# FPGA Hardware Implementation of the Proposed Design

This chapter introduces the implementation of the proposed design using the Very High Speed Integrated Circuit (VHSIC) Hardware Descriptive Language (HDL), as it provides a better hardware optimization, higher speed of operation, lower power consumption, and smaller size compared to any hardware running using a general purpose processors.

The implementation chapter is divided into two sections, the first one includes the VHDL implementation and its RTL simulations, while the second section includes the hardware routing, synthesis, and its gate level simulation.

## 5.1 VHDL Implementation of the Proposed Design and RTL Simulations

For a general consideration of the system data length, a series of tests are conducted. A 32 bit system is found to be the optimum data length which achieves high accuracy results while maintaining a fair system complexity. In the following sections a detailed implementation of the system feature extraction unit and SVM classifier is described along with the RTL simulation results. These simulations are compared to the software simulation conducted in chapter 3, and the error rate is calculated. The top level block diagram is illustrated in Figure 4.1

### 5.1.1 Feature Extraction Unit

As demonstrated previously in Figure 4.4, the feature extraction unit is composed of two blocks, the WPT Unit and the Features Calculation Unit. Each of these sub-systems will be implemented and tested separately, and they will be integrated.

#### 5.1.1.1 Wavelet Packet Transform (WPT) Unit

According to the wavelet packet tree proposed in Figure 4.6, a total number of 19 filter pairs are required. Each pair is composed of one HPF and one LPF. Each filter is implemented as a convolution of the input signal with all of the filter coefficients H(0) and H(1).

However, in order to implement the wavelet packet tree, it is nearly impossible to implement 19 filter pair, as it will require a very high system complexity. Therefore the following approach

is proposed instead. Since time is not an issue at this point, a single pair of HPF and LPF is implemented as the core of the WPT unit. While the new data points are being received from the EEG stream the filter pair unit will be allocated to further decomposition of previously processed data, therefore; some additional circuitry is implemented in order to control the data flow. Figure 5.1 shows the block diagram of this unit.

The WPT constituent units are implemented and simulated each on its own, before they are integrated. At the end of this section, the WPT unit simulation and results are discussed.

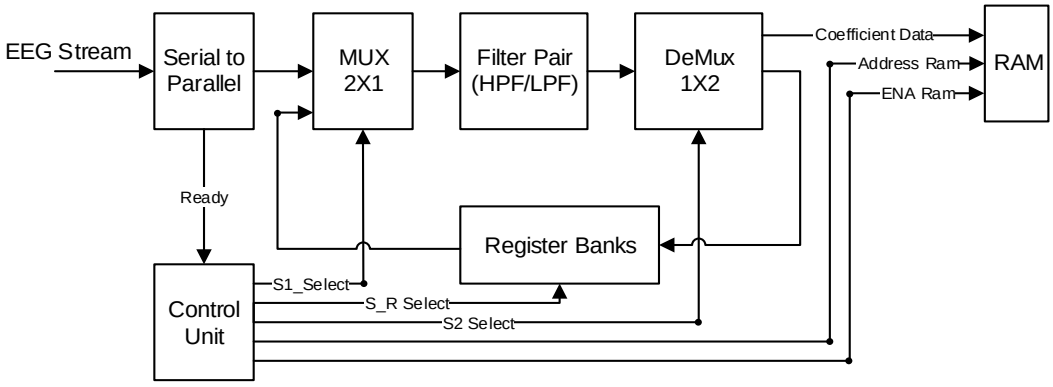


Figure 5.1: Wavelet Packet Transform Block Diagram

**a) Filter Pair Unit**

For a second order filter, a pair of multipliers and one adder are required to perform the two point convolution as illustrated in the block diagram in Figure 5.2 where h(n) represent the filter coefficients and x(n) represents the EEG input stream. A truncation unit is implemented to maintain the system data length at 32 bit.

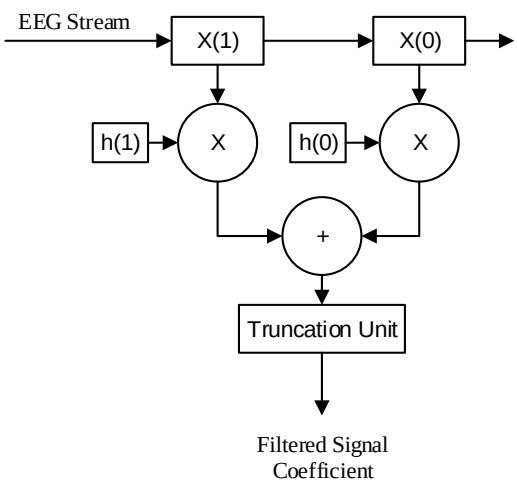


Figure 5.2: 2nd Order Filter Block Diagram

The corresponding VHDL implementation of HPF and LPF pair illustrated in Figure 5.3 consists of 3 subsystems. U0, U1, U2, and U3 are 32 bit multipliers, U4, and U5 are 53 bit adders, finally U8 and U9 are truncation units. The 32 bit inputs from the EEG stream are represented as follows, 11 bits for the integer, and 21 bits for its decimal fraction. 32 bit multiplication would normally result in a 64 bit output represented as a 22 bit integer, and 42 bit decimal. However knowing that both filters coefficients – Coeff1 and Coeff2 – are smaller than one, we can assume that the multiplication will result in a number smaller than the input itself. For optimizing the hardware implementation, the 11 bit number will not be doubled while the decimal fraction bits will be doubled, resulting in a multiplication output of 53 bits. Once the addition is complete, and in order to maintain the 32 bit format the truncation unit is implemented to truncate the extra decimal fraction bits. Furthermore the implemented algorithm for the multiplier is the modified booth Wallace tree, since it provides the best area and delay compared to other multiplication techniques [27, 28].

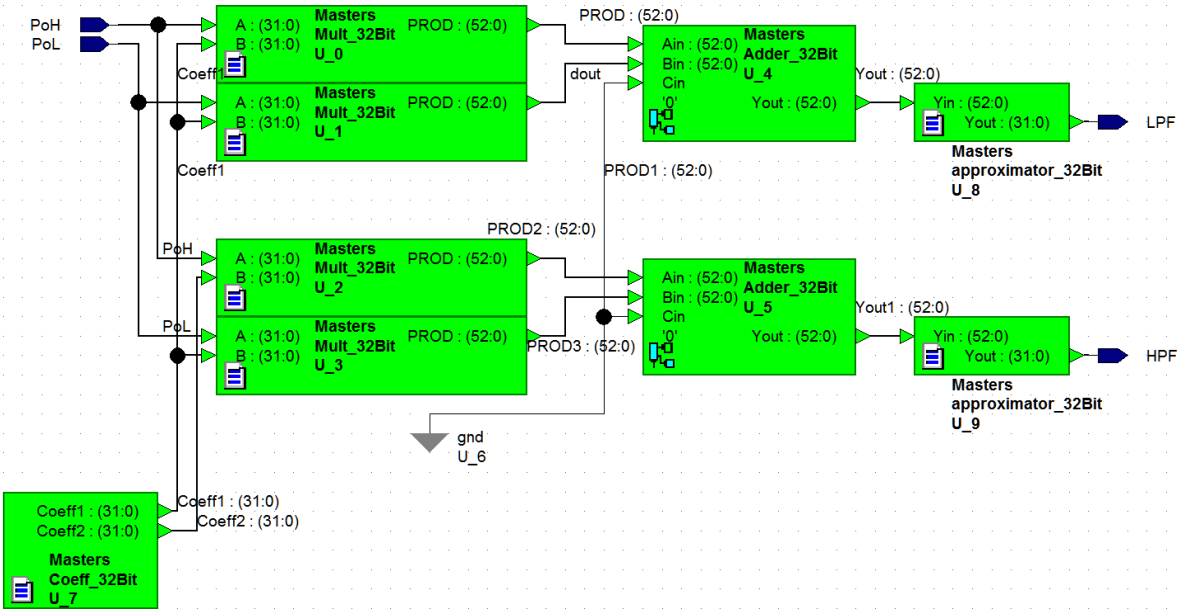


Figure 5.3: 2nd Order Filter VHDL Implementation

Figure 5.4 illustrates the RTL simulation of this implementation. Signals COEFF1 and COEFF2 hold the filter coefficients 0.707 and -0.707 respectively, and signals POH and POL represents the input x(n) and x(n-1) respectively. For both the HPF and LPF, PROD, PROD1, PROD2, PROD3 signals hold the multiplication result, while the signals YOUT and YOUT1 hold the summation result (PROD + PROD1) and (PROD2+PROD3) respectively. Finally the output is held in the LPF and HPF signals.

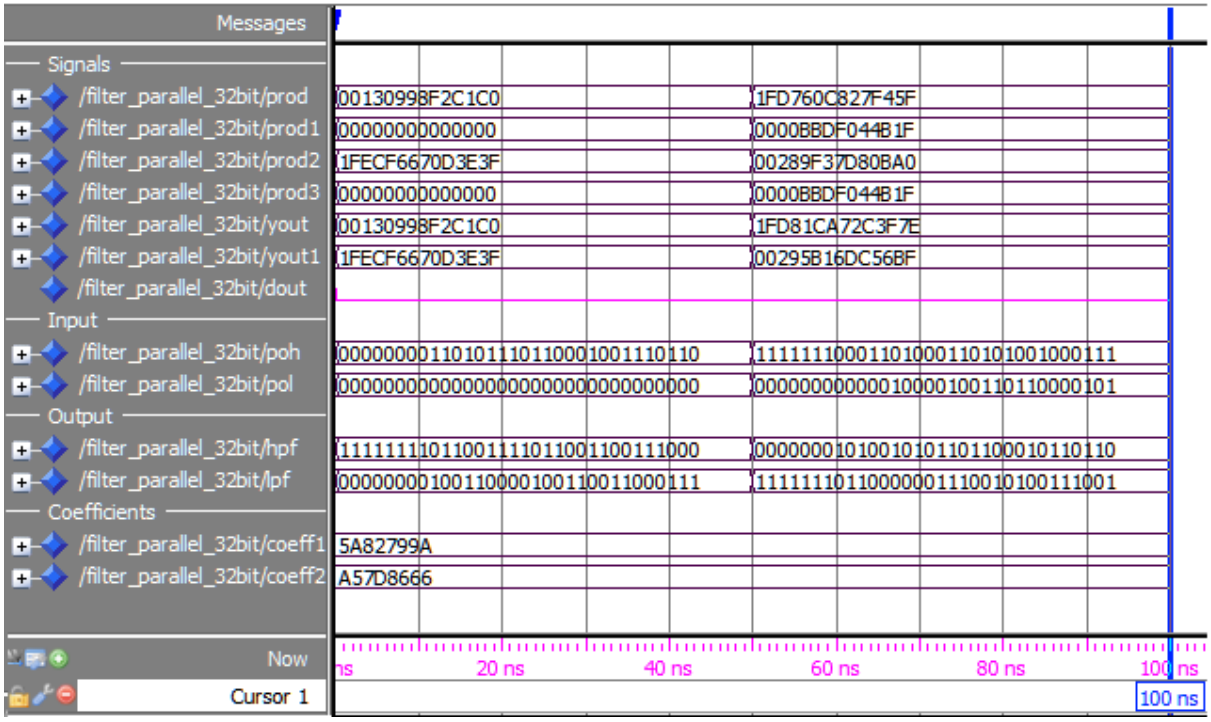


Figure 5.4: Filter RTL Simulation Results

In order to test the filter pair unit, a number of two different data sets are applied and their corresponding outputs are compared to the software simulation results. This test results are demonstrated in Table 5.1. The error in the order of E-07 is very small and acceptable error.

Table 5.1: Filter Pair Test Results

| Input | | Desired Output | | Implemented Output | Error |
|---|---|---|---|---|---|
| PoH | 0 | HPF | -4.75937256567869 | -4.7593722343 | 3.31E-7 |
| PoL | +6.73076923076 | LPF | +4.75937256567869 | +4.7593722343 | 3.31E-7 |
| PoH | +0.25946275946 | HPF | +10.3389544624040 | +10.3389539719 | 4.91E-7 |
| PoL | -14.362026862 | LPF | -9.97201870904105 | -9.9720182419 | 4.67E-7 |

**b) RAM Unit**

The RAM Unit in turn composed of two 11.1992 KB RAM as illustrated in Figure 5.5. While one of the RAM is in write mode storing the coefficients of one epoch, the other RAM is in read mode allowing the Feature Calculation Unit to access the coefficients extracted from the previous epoch.

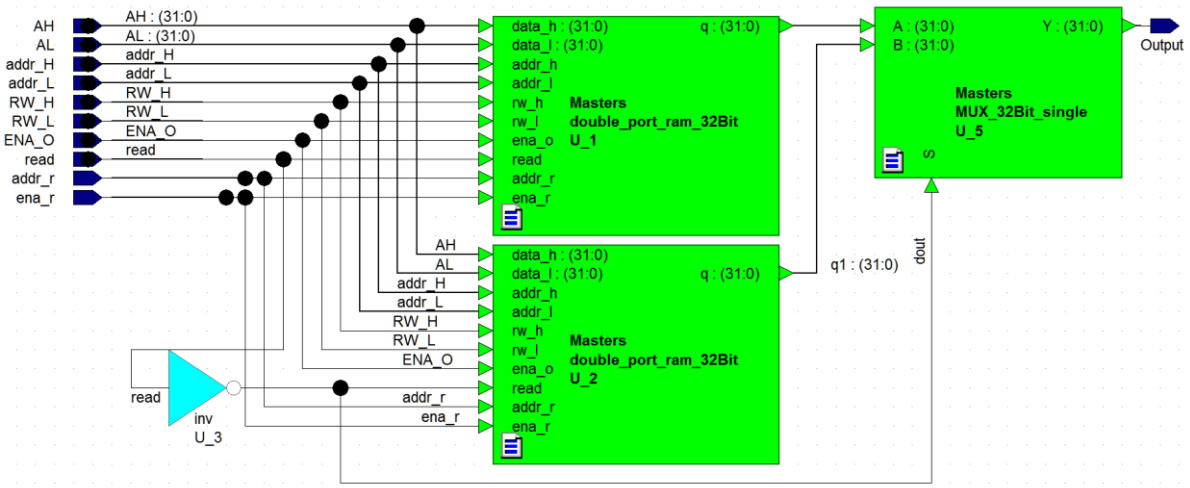


Figure 5.5: RAM Unit Implementation

The memory units implemented are dual port RAM [29], allowing the system to perform read and write operations on both HPF and LPF coefficients simultaneously. Therefore two address bus “addr_H” and “addr_L” are implemented to map both the HPF and LPF output coefficients – located at AH and AL input ports – each to its corresponding address.

Since when a write operation is executed at certain points only one coefficient is stored while at other points both HPF and LPF output coefficients are stored. Two write enable signals are implemented. “RW_L” and “RW_H” are active low signals which enables the write operation each for its corresponding input port. The signal “ENA_O” is implemented as the RAM enable at which write operations are performed.

As for the read operation, the active high signal “read” enables this mode. While the address bus “addr_r” and the enable signal “ena_r” are designed for the read operation.

The multiplexer implemented, is designed to select the RAM which is active for reading to be interfaced to the following block. The multiplexer select it connected to the inverted signal “read”. While the first RAM is in read mode, the second RAM read is disabled by the inverter and the multiplexer select is set to the first RAM output.

A segment of the RTL simulation is illustrated in Figure 5.6 showing only the address range 0 to 20 which represents the first 21 samples of the C5 coefficient.

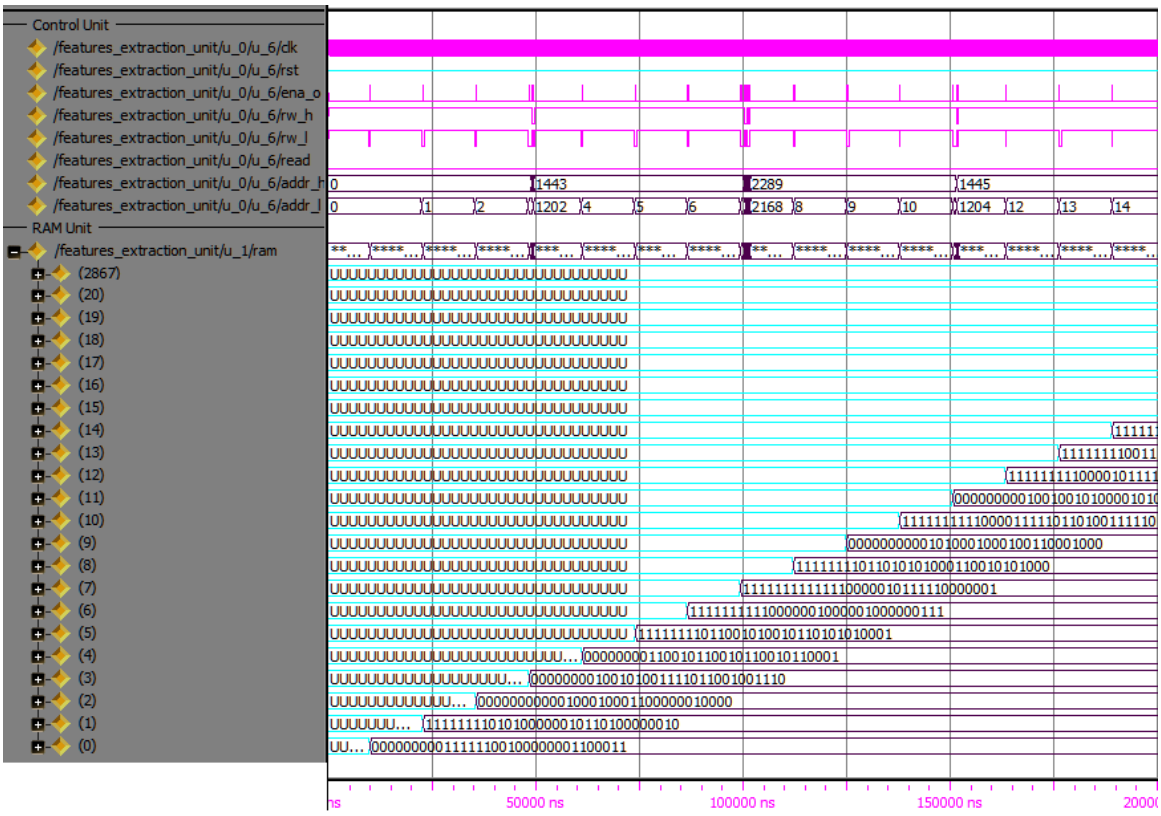


Figure 5.6: RAM Unit RTL Simulation

### c) Register Banks Unit

The register banks unit illustrated in Figure 5.7 consists of 36 register of 32 bit each, with a total size of 0.14 KB. A pair of registers is implemented for each of the following decomposition points, as illustrated in Figure 4.6, (C1, C2, C3, C4, C7, C8, C9, C10, C11, C12, C13, C14, C15, C19, C20, C21, C24, and C29).

Each pair is used to temporarily hold the data points until their content is selected for further decomposition. The two Coefficients C6 and C37 are discarded as their values are not needed – see Figure 4.6, while the desired coefficients such as (C5, C16, C17, C18, etc.) are directly routed to a RAM unit for storage.

The register banks unit is implemented with dual input ports and output ports. A read/write, and enable signals are implemented to control the data flow of this unit. The signal S, is the register select which maps the input data set to its corresponding register according its location in the WPT tree.

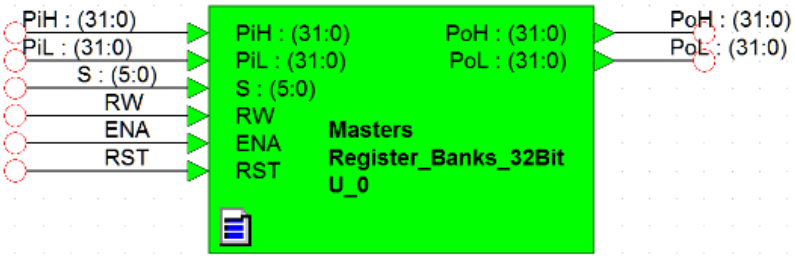


Figure 5.7: Register Banks VHDL Implementation

Figure 5.8 illustrates the register banks unit RTL simulation. As illustrated in this simulation, the select signal "S" is of length 6 bits. The most significant bit determines writing the value corresponding to x(0) or x(1), while the rest 5 bits represents the coefficient being processed. For example the selector was first set to "000000" when the data is ready, the enable signals was applied which allowed the register banks to store the first data set decomposed to its corresponding coefficients C01L and C02L.

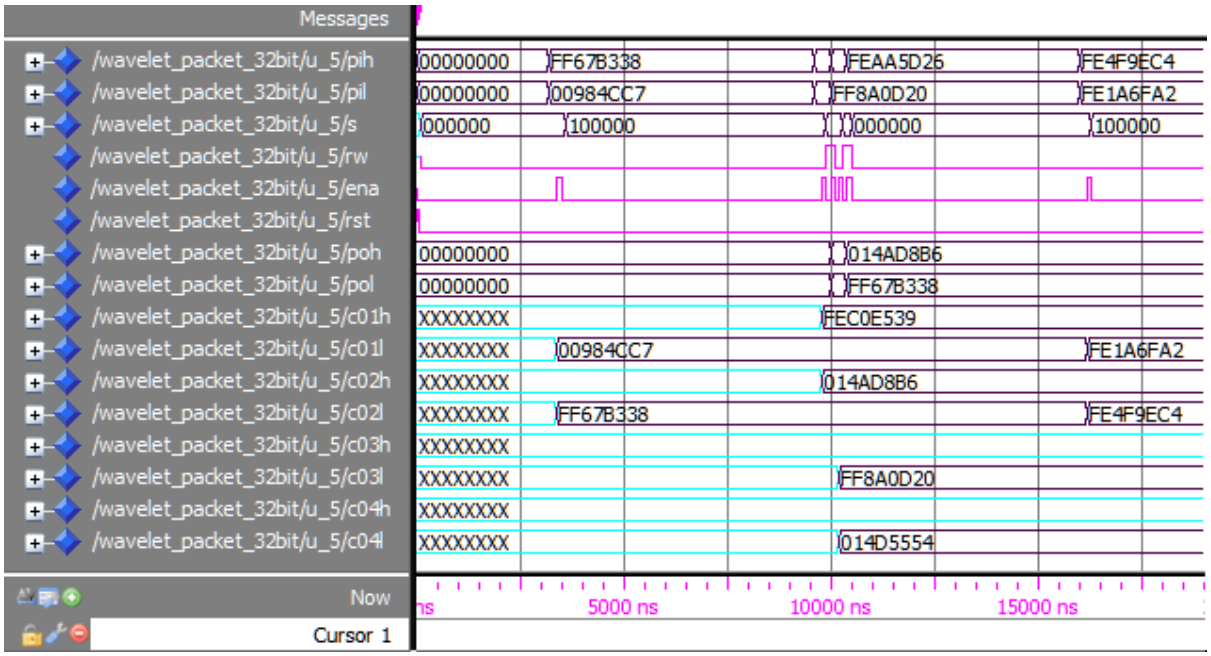


Figure 5.8: Register Banks Unit RTL Simulation

### d) Integration of the Wavelet Packet Tree Unit

Once each of the constituents of the WPT unit is implemented and simulated separately, the system is integrated as in Figure 5.1 to form the WPT unit. This implementation is illustrated in Figure 5.9.

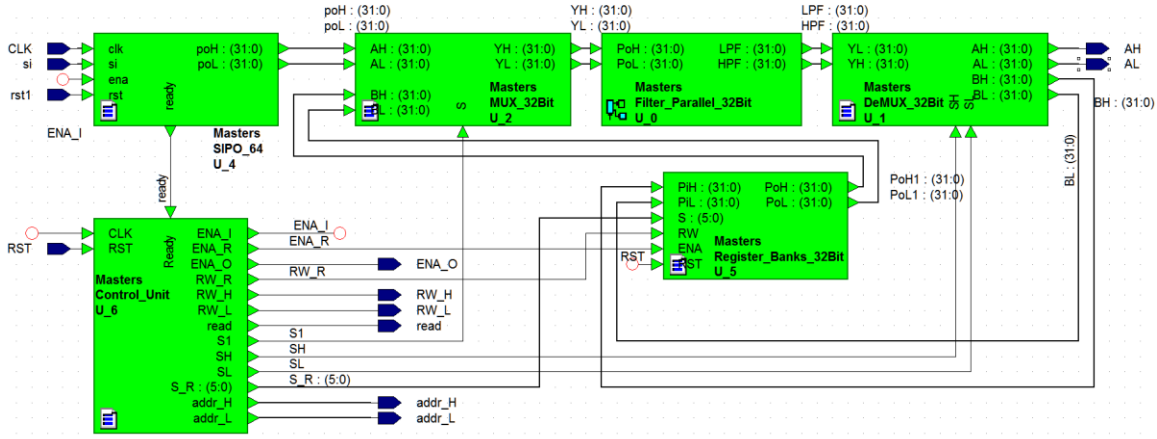


Figure 5.9: Wavelet Packet Transform VHDL Implementation

In addition to all the constituents previously discussed and simulated, a Control Unit and a Serial Input Parallel Output (SIPO_64) Unit are added to the design implementation.

1- The SIPO_64 unit is implemented to read the EEG input stream where each value is represented in 32 bits. Every two data points are read from the stream a ready signal is sent to the control unit to enable processing the newly received values. The clock of the SIPO unit is set at 4.096 KHz to match the sampling rate of 128 Hz for the input stream.

2- The Control Unit is role is to manage the data flow and synchronize the work of all the system constituents. The clock frequency of the Control unit is the double of the SIPO_64 unit clock, running at 8.192 KHz. In order to properly map each extracted value to its proper coefficient two counters are being used, the process counter and the step counter. The process counter identifies the available data for processing at the current position, while the step counter identifies the accomplished and remaining steps within the current process.

   Figure 5.10 demonstrates a partial work plan for the Control Unit implemented. At each new process two new set of input data must be ready to be convolved with the two filter coefficients h(0) and h(1), since after each level of decomposition a decimation by two is applied in time, there is no need to calculate the undesired points and then discard them. This unit is optimized to use the least possible hardware.

   At the first process, only one step is performed, which is convolving (x(0), 0) with the HPF and LPF to produce the first data set of C1 and C2 which will be named C1L and

C2L. These data set are then stored in the register banks. At the second process, new data input (x(2), x(1)) are in turn convolved with the filters to produce the second data set C1H and C2H. Now that C1 and C2 are ready for further decomposition, the next step in this process is to convolve (C1H, C1L) with the filters. This is done by loading (C1H, C2L) from the register banks and routing them to the filter unit in order to produce the coefficients of C3L and C4L which are in turn stored in the register banks. The third step in this process is the same as the second step but this time convolving (C2H, C2L) with the HPF and LPF pair, producing C5 and C6. C5 represents one of the desired coefficients and is stored in the RAM unit, while C6 is discarded. The rest is a repetition but referring to different coefficients.

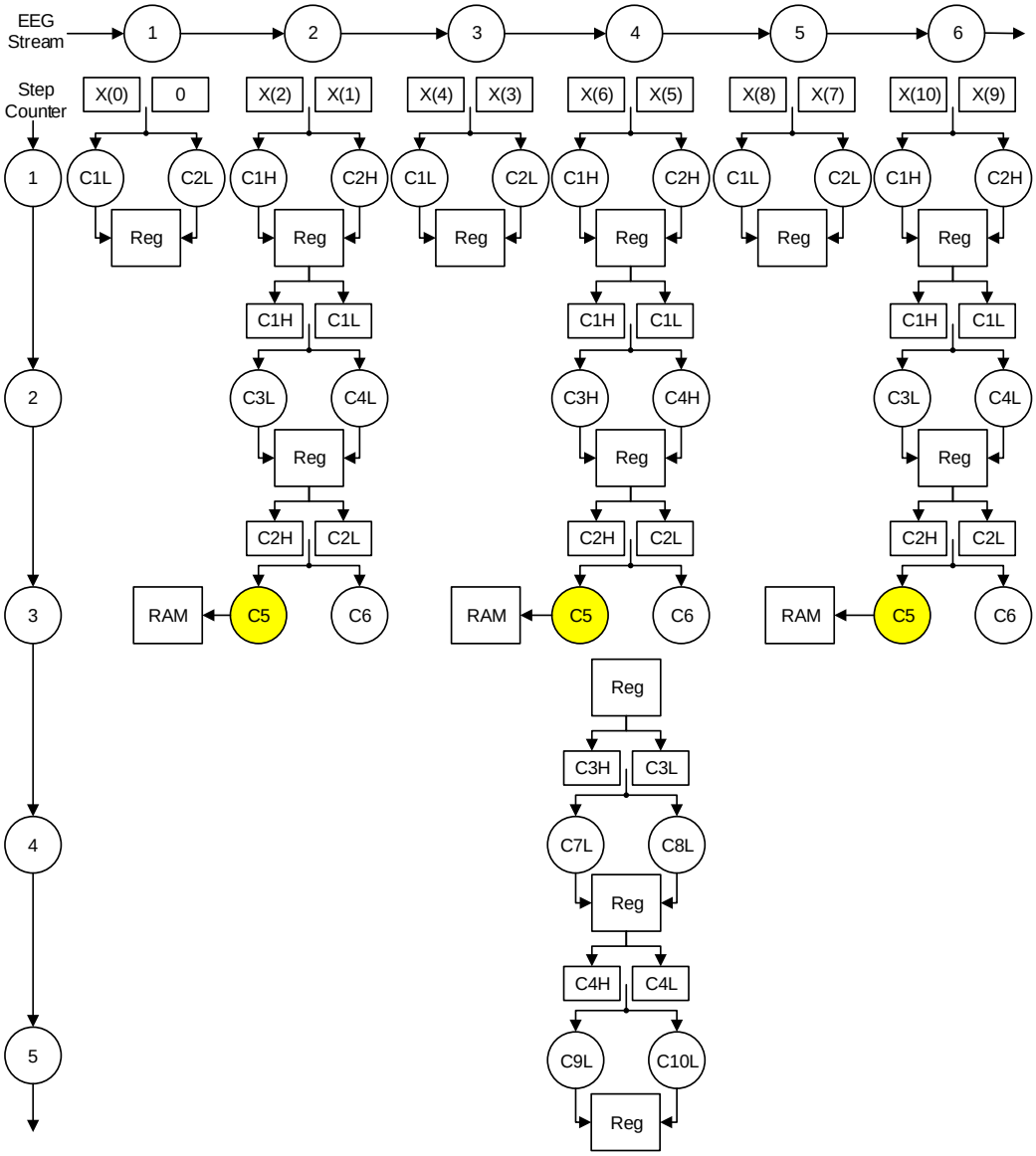


Figure 5.10: Control Unit Partial Work Plan

Figure 5.11 illustrates a small portion of the RTL simulation of the WPT unit implementation. In this simulation signals are divided into 5 groups: Control Unit, Input Stream, Filter Pair IO, Register Banks IO, and finally the RAM output group.

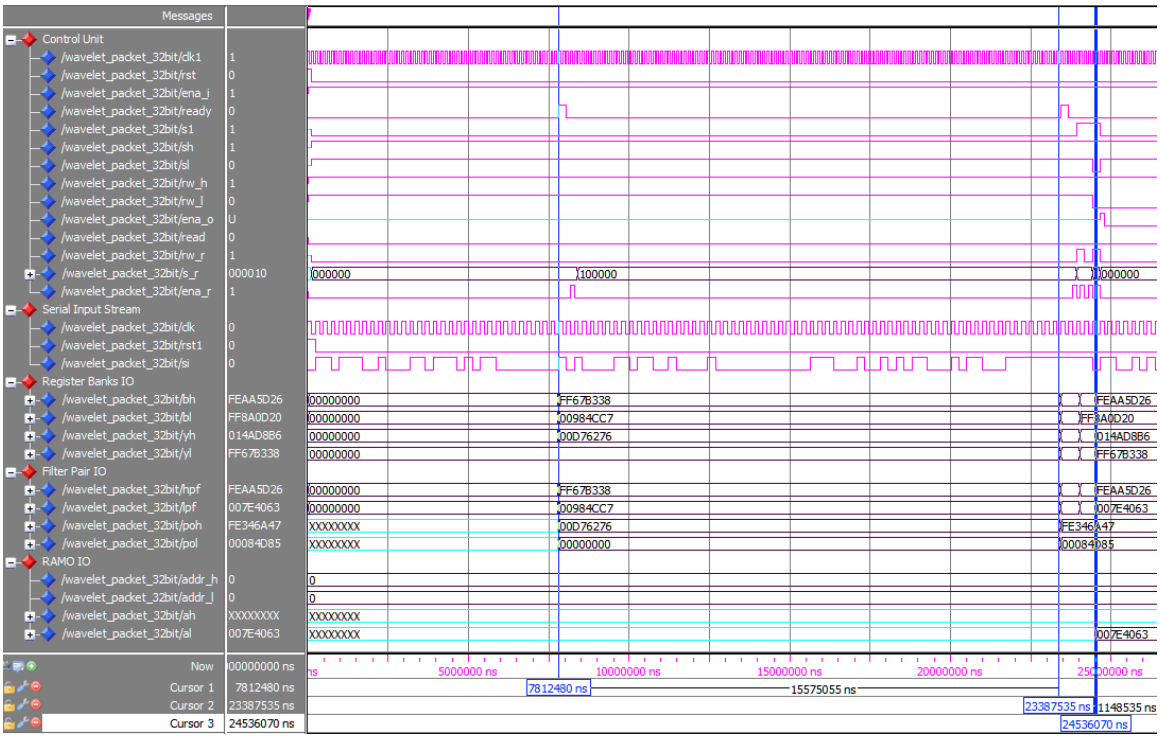


Figure 5.11: WPT RTL Simulation

Cursor 1 represents the first data read from the EEG stream, at which point a ready signal is sent to the control unit. The control unit routes the input data from the stream (x(0), 0) represented in “poh” and “pol” signals respectively to the filter pair, producing the first HPF and LPF output representing C1L and C2L. These coefficients are stored in the register banks by setting the “ena_r” signal.

Cursor 2 represents the time at which the second data set is ready. Repeating the same process done previously, and extracting the second set of coefficients, C1H and C2H. Furthermore the data (C1H, C1L) and (C2H, C2L) are processed.

Cursor 3 represents the time needed to extract the first desirable coefficient of C5 which turned out to be at 24.536070ms. The first C5 coefficient is equal to 007E4063.

Table 5.2 holds the first 5 data points of each coefficient that are extracted from the RTL simulation. The same color code is used as was used previously in Figure 4.6.

Table 5.2: Sample of Coefficients Extracted from an Epoch by the RTL Simulation

| Coefficients | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| C5 | 3.945359706 | -5.49450540 | 0.534187316 | 4.65506649 | 6.349205493 |
| C16 | 7.726647853 | -0.19078063 | 3.021977901 | -7.60836362 | -10.8363852 |
| C17 | 8.382935524 | 4.250610351 | -13.6141633 | -4.08272266 | 8.257019996 |
| C18 | 16.86889457 | 7.822039127 | -13.3547000 | 2.327533721 | -9.66117143 |
| C22 | 24.99210214 | 57.73302364 | 24.22315597 | -48.9589424 | 36.38599729 |
| C23 | 10.69239902 | -1.62423086 | 19.50695085 | -1.73215341 | 6.070627212 |
| C25 | 12.13855791 | -19.9170570 | -20.7264742 | 29.25233411 | 12.62151002 |
| C26 | -32.3793821 | 29.26312541 | -10.1177129 | -14.1000456 | -16.5498814 |
| C27 | -12.5000996 | 2.703451156 | 7.916098117 | -2.98405313 | 2.045125484 |
| C28 | -4.67573261 | 17.17583036 | 3.52366209 | -24.2339472 | -1.6781907 |
| C30 | -51.0588369 | -68.650794 | 0.259462356 | 146.0470085 | -86.3362331 |
| C31 | -13.4291052 | -17.9105615 | 47.95482254 | -53.1669721 | -19.9290289 |
| C32 | 29.60355615 | 15.53724098 | 33.34096479 | 150.4807686 | -16.5102252 |
| C33 | -23.5252599 | -14.1025643 | -24.6642251 | 12.59920549 | -22.7983822 |
| C34 | -12.5133547 | 4.632172584 | 16.76587295 | 60.12667846 | -22.4473443 |
| C35 | 12.85294532 | -23.0845541 | -13.0341882 | 14.88858270 | -14.1903243 |
| C36 | 6.526632309 | 12.56104993 | 22.70299196 | -23.4050669 | -13.3814101 |

Table 5.3 summarizes the difference found between the software and the RTL simulations by comparing the coefficients extracted in both cases which are shown in Table 4.3 and Table 5.2. The maximum error does not exceed the order of E-06 which is acceptable.

Table 5.3: Percentage of Error in the Different Bands

| Bands | Maximum Error | Minimum Error | Average Percentage of Error |
|---|---|---|---|
| Delta | 1.617985969E-06 | 6.404398789E-09 | 4.0223790E-06 |
| Theta | 1.731043337E-06 | 4.448637014E-10 | 2.1286305E-05 |
| Alpha | 1.144613852E-06 | 1.328466225E-09 | 2.3519885E-05 |
| Spindle | 2.345717838E-06 | 2.750146776E-09 | 8.2230806E-05 |
| Beta1 | 1.162690852E-06 | 5.822187176E-10 | 5.4995299E-05 |
| Beta2 | 1.292525634E-06 | 1.569924745E-16 | 7.7743567E-05 |

### 5.1.1.2 Features Calculation Unit

Once all the relevant coefficients are extracted and ready in the RAM Unit, the features are calculated using an Arithmetic Logic Unit (ALU) specifically designed to meet the needs of this system. The ALU block diagram illustrated in Figure 5.12 is composed of a 32-bit multiplier, a 64-bit divider, a 64-bit accumulator, a 64-bit registers with an embedded multiplexer, a RAM to store the calculated features, and a control unit that manages the data flow within this unit until the features are calculated and stored in the local RAM which is implemented at a size of 1.375 KB. This unit VHDL implementation is illustrated in Figure 5.13.

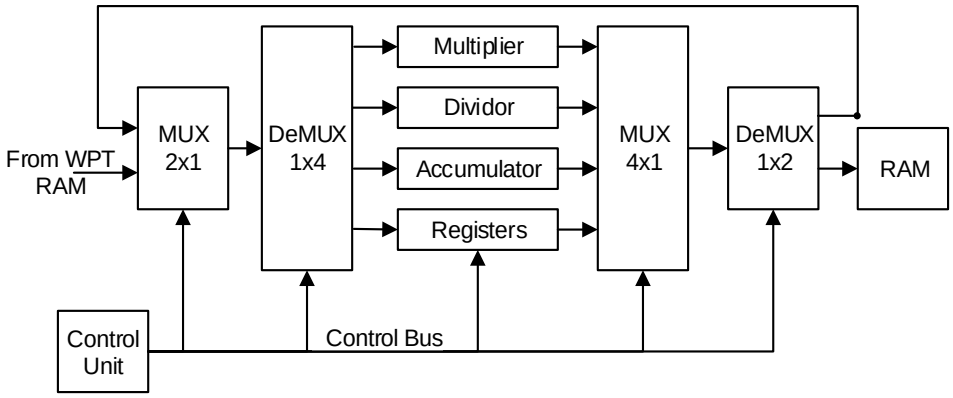


Figure 5.12: Feature Calculation Unit Implementation

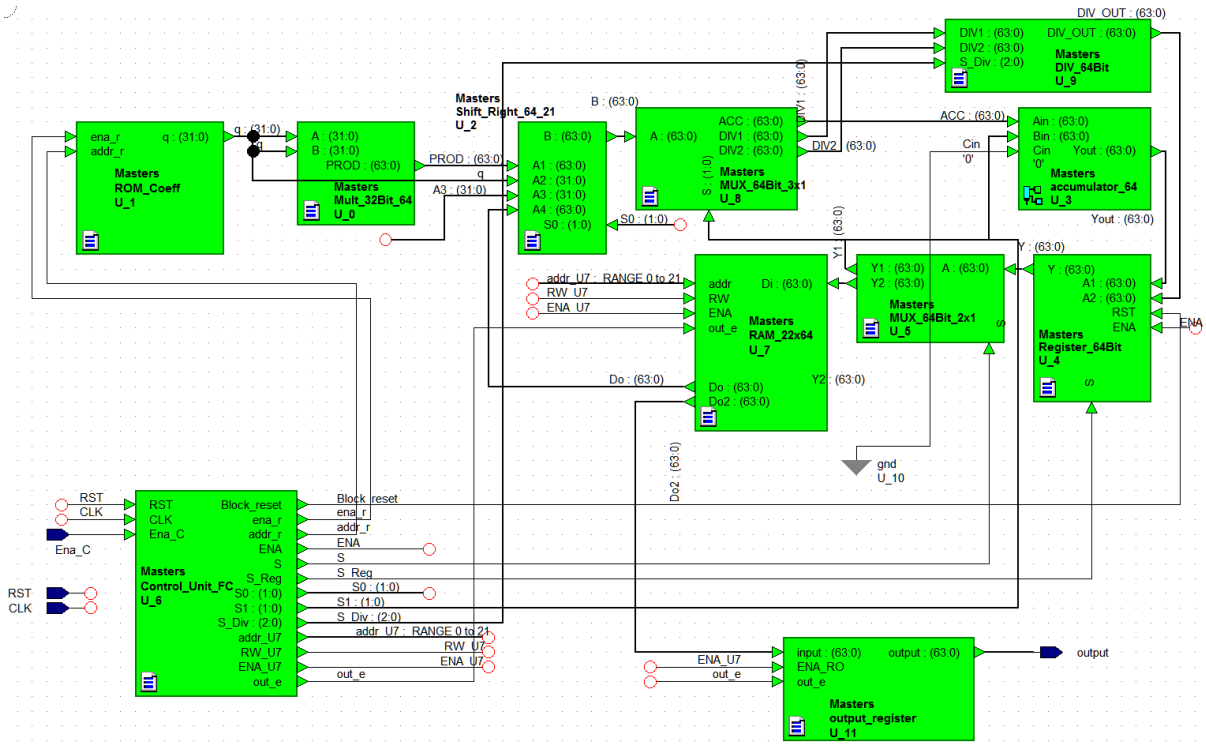


Figure 5.13: Feature Calculation Unit VHDL Implementation

Operation of this unit requires a very precise timing for handling the communication and data transfer between its different constituents. The features are divided into 5 different groups, as previously discussed in chapter 4, which form the 22 required features. The control unit operation is divided into 5 sequential tasks each one is dedicated to a group of features.

**a) Group 1: Energy of Each Band (E1 to E6)**

The energy of each band corresponding to equation 4.1, where x(n) represents the coefficients within the band. In order to compute this equation, the coefficients of each band are read from the RAM of the WPT unit, and routed to the accumulator through the 32 bit multiplier by setting the shift register – Block U2 – to select the input A1. Then the data is accumulated in the accumulator and register loop until all coefficients of the desired band are summed. Once the feature is computed, it is routed to the RAM through the multiplexer U5 and the signal "block_reset" is set to reset the accumulation loop.

**b) Group 2: Total Energy in all Bands (E7)**

The total energy – in equation 4.2 – is the summation of the previously computed energies corresponding to each band. In order to compute the total energy, the control unit sets the RAM to read mode, the Energy of each band is read and routed to the accumulator directly. Once the 6 energies are summed the result is stored at the same RAM in location 7.

**c) Group 3: Ratio of Different Energy Values (E8 to E10)**

The desired ratios are calculated as in equations 4.3, 4.4, and 4.5. For each ratio, first the summation is performed by reading the desired energies from the RAM, and directed to the accumulator, then the division is performed by routing the desired energies to the division unit. Finally the result is redirected to the RAM and stored.

**d) Group 4: Mean of the Absolute Values of Each Band (E11 to E17)**

The mean of the absolute values for each band formulated in equation 4.6, is computed similar to group1; however the coefficients absolute values are summed instead. In order to compute these energies, the coefficients of each band will be read from the input RAM, but in this case, the multiplier will be bypassed, and the coefficients will be directly routed to the accumulator by selecting the input A2 for the shift register in block U2. The shift

register is set to adjust the 32 bit input into 64 bit format while removing the sign. The rest is similar to the process conducted for group 1.

**e) Group 5: The Standard Deviation for Each Band (E18 to E22)**

The standard deviation of each band formulated in equation 4.7, is computed as follows: First the average of the coefficients corresponding to the desired band must be calculated first, same as done in groups 4, however, the coefficients are summed as signed numbers. After calculating the average value, it is redirected to the subtraction unit, where it will be subtracted from each coefficient. The output of the subtraction is squared and accumulated to the previous coefficients. Finally the result is divided by N in the divider unit. As for the square root function, it is implemented as a lookup table. The final stored in the RAM.

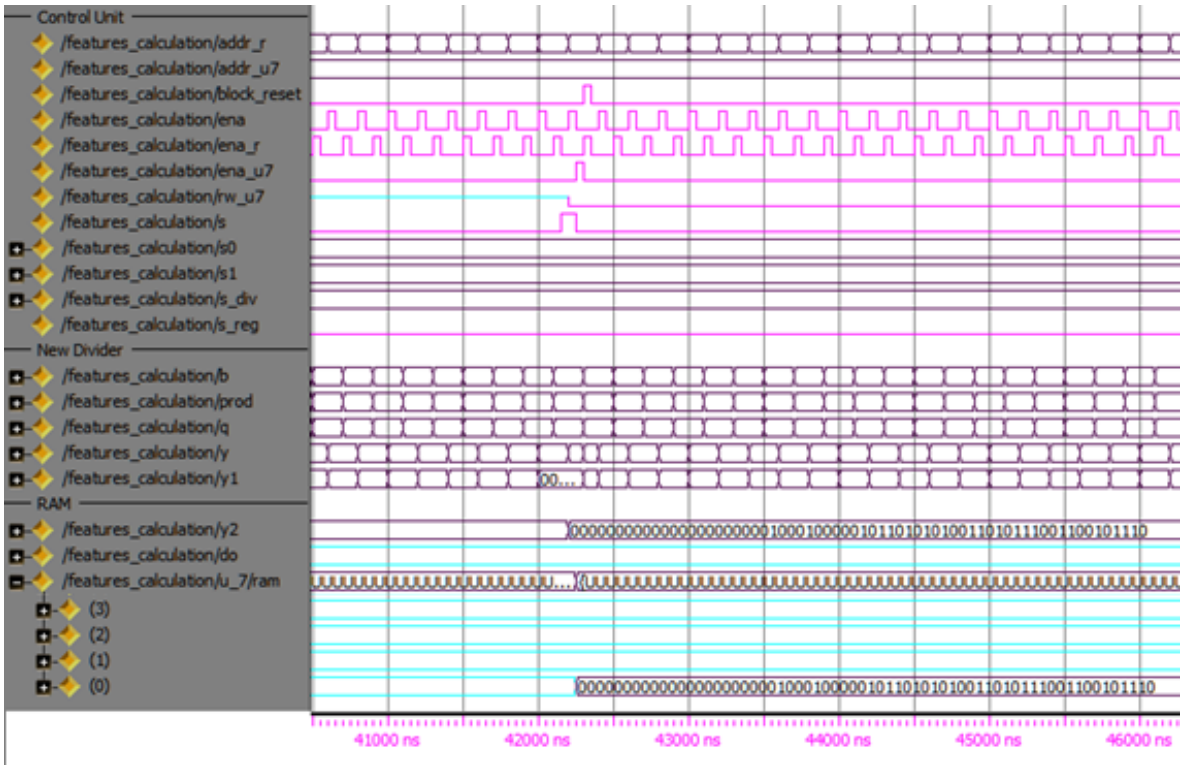


Figure 5.14: Feature Calculation Unit RTL Simulation Result

The visible portion of the RTL simulation illustrated in Figure 5.14 demonstrates the calculation of the first feature. The rest of the feature are calculated in the same method and stored in the RAM. The extracted features of the first epoch along with their respective percentage of error are summarized in Table 5.4.

Table 5.4: Calculated Features for the First Epoch

| RAM Address | Value (HEX) | Error % | RAM Address | Value (HEX) | Error % |
|---|---|---|---|---|---|
| 0 | 220B54D732E | 0.0007143504 | 1 | 0964864C968 | 0.0051795076 |
| 2 | 1EEED74EB0 | 0.0107828309 | 3 | 304CCD594B | 0.0138114445 |
| 4 | 0499215A453 | 0.0120902704 | 5 | 3988D64574 | 0.0386395353 |
| 6 | 3895442CE58 | 0.0058338642 | 7 | 0000016C9C | 0.0101601585 |
| 8 | 00000602FF5 | 0.0054090944 | 9 | 000085AAF | 0.0041059074 |
| 10 | 0006A1B326 | 0.0501886394 | 11 | 00024E9958 | 0.1133407177 |
| 12 | 0001A8D791 | 0.1522239308 | 13 | 00015EFE44 | 0.1618829049 |
| 14 | 00015E01B1 | 0.114845313 | 15 | 0000B3220B | 0.1346122821 |
| 16 | 0008E9930C | 0.0111301565 | 17 | 00034F381D | 0.0091201362 |
| 18 | 00024BB4A2 | 0.0099201500 | 19 | 000208FEC5 | 0.0141201300 |
| 20 | 0000232E148 | 0.0251241607 | 21 | 000134CD9F | 0.0054141203 |

### 5.1.1.3 Features Extraction Unit Integration

Once each of the system constituents are implemented and tested, the features extraction unit is integrated by combining the WPT, the RAM and the Features Calculation Unit as illustrated in Figure 5.15.

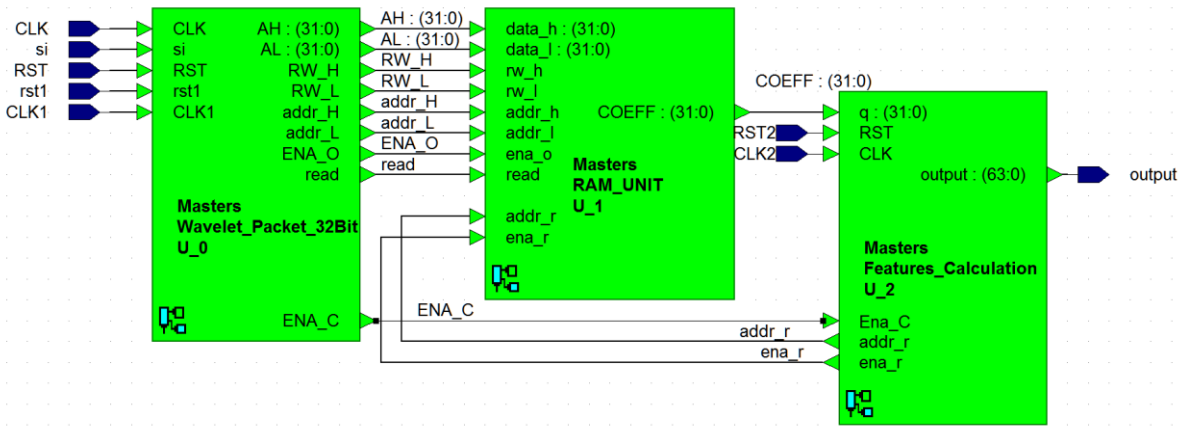


Figure 5.15: Feature Extraction Unit Integration

Figure 5.16 illustrates the RTL simulation of the Features Calculation Unit system integration. The simulation shows that the results are matching to the results extracted by simulating each unit separately. The time needed to calculate all the features after the system integration is 30.001567 seconds as shown at cursor 1.

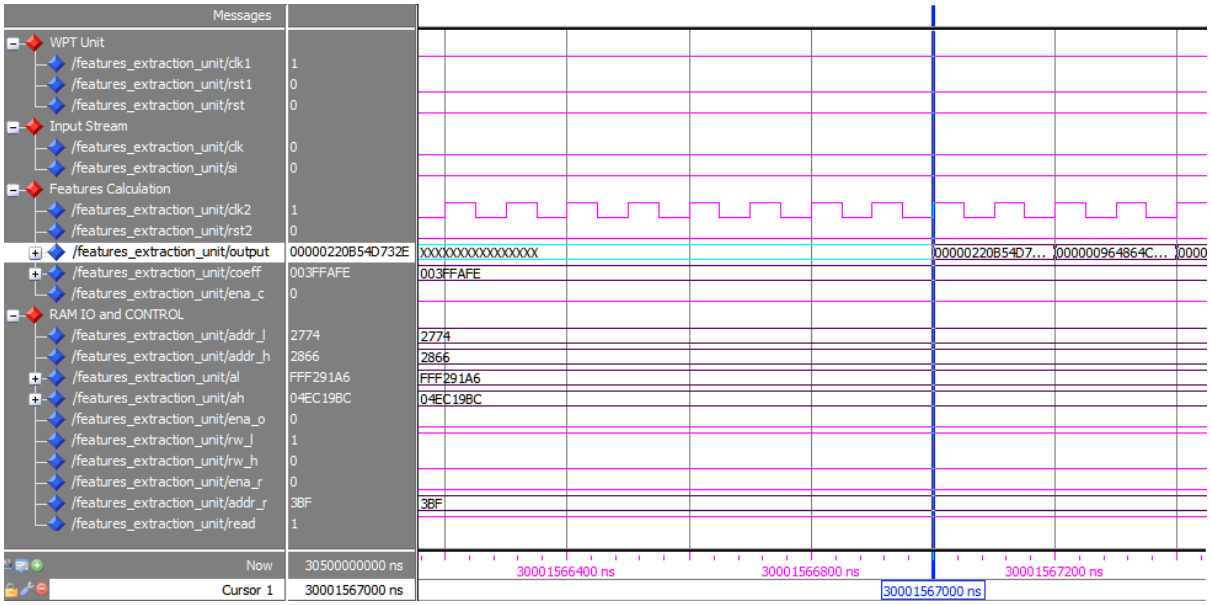


Figure 5.16: Feature Calculation Unit RTL Simulation Results

### 5.1.2 Support Vector Machine Classifier Unit

The support vector machine classifier illustrated in Figure 5.17 consists of three main units. The first one – Kernel Function Unit – is the unit at which the kernel function is applied, Whereas the second one – Voting Unit – where the correct label that corresponds to the current epoch is selected. The third unit is the control unit which controls the classifier unit operation.

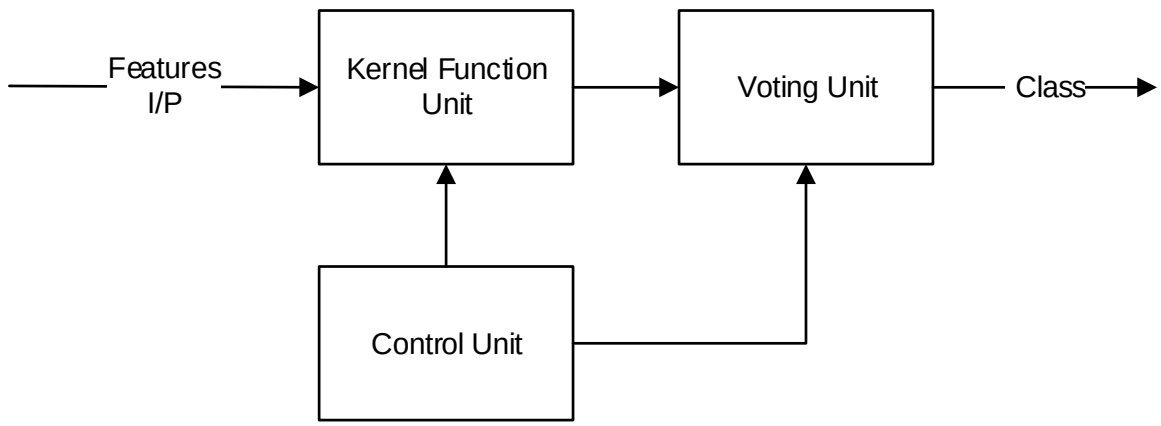


Figure 5.17: Support Vector Machine Classifier Block Diagram

### 5.1.2.1 Kernel Function Unit

Applying the kernel function produces a number of coefficients called the kernel values (K_value). Given the model parameters used in this implementation, a total number of L*I support vectors (SVs) are used, where L = 409 and I = 22. The SVs are pre-calculated by running the training mode offline, and stored in the ROM_SV unit [29]. Figure 5.18 illustrates the functional block diagram for the kernel function whereas Figure 5.19 represents its detailed hardware implementation.

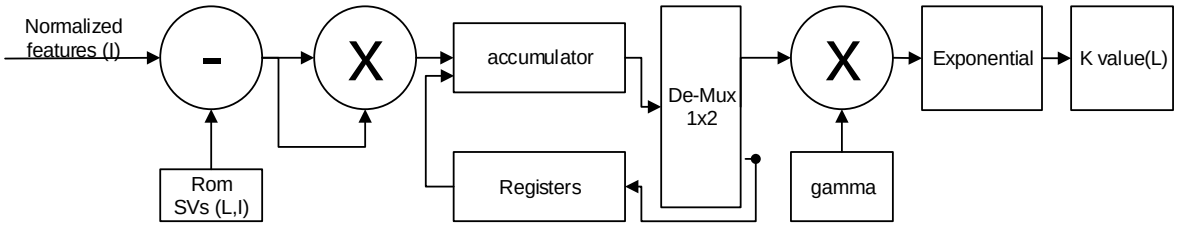


Figure 5.18: Functional Block Diagram of the Kernel Function Unit

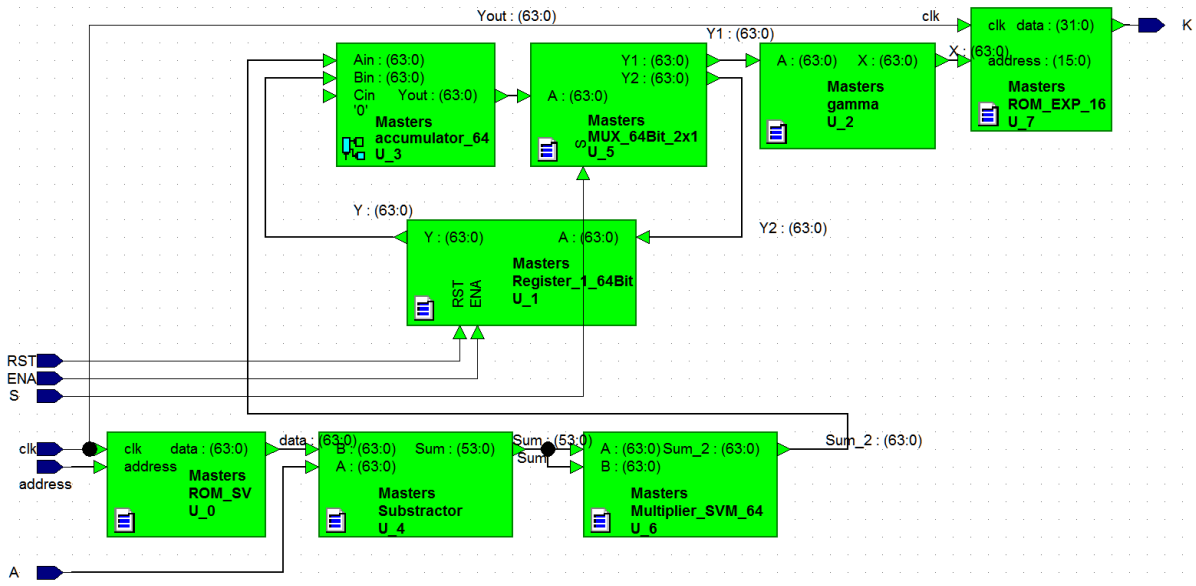


Figure 5.19: VHDL Implementation of the Kernel Function Unit

The operation of this unit is as follows; the first 22 SVs are subtracted from the 22 features of the epoch being processed, and their resultant are squared and accumulated, and applied to the RBF kernel function in Table 3.1 to produce the first kernel value, where x represents the previously mentioned summation. This operation is repeated L times to cover all the SVs producing a total number of 409 kernel values.

The exponential function is implemented as a lookup table stored in the ROM_EXP_16 [29] unit as shown in Figure 5.19. As a matter of optimization regarding the area usage, the exponential lookup table is implemented to cover only the region where x is confined between 0 and -7.9998779296875. A 16 bit address is used with the decimal point located after the 3rd bits. The ROM_EXP_16 module consists of 65536 location with 32-bit each giving a total size of 0.25 MB. The difference between each two successive memory locations is 0.0001220703125.

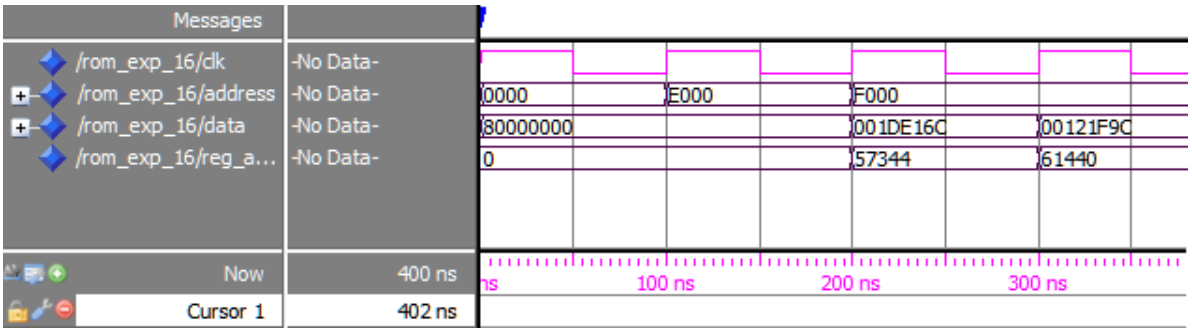


Figure 5.20: RTL Simulation of the Exponential Function

Figure 5.20 illustrates the RTL simulation of the exponential function, while Y illustrates the RTL simulation of the kernel function unit.

As the exponential function is the crucial function when getting the kernel values, a series of tests are simulated to the exponential unit to determine its input error due to the approximation. Table 5.5 summarizes sample of these results for different inputs at different ROM locations. For example, at address 893 the accurate input should be “0.109008789” while due to the approximation error, the implemented input is off by 4.07776-05.

Table 5.5: Exponential Unit Input Error

| Accurate Input | Approximated Input | ROM Address | Input Approximation Error |
|---|---|---|---|
| 0.109008789 | -0.109049567 | 893 | 4.07776E-05 |
| 0.083007813 | -0.083026011 | 680 | 1.81984E-05 |
| 0.220336914 | -0.220357577 | 1805 | 2.06628E-05 |
| 0.165649414 | -0.165630527 | 1357 | 1.88874E-05 |
| 0.276733398 | -0.276712799 | 2267 | 2.0599E-05 |

| Accurate Input | Approximated Input | ROM Address | Input Approximation Error |
|---|---|---|---|
| 0.145629883 | -0.145625825 | 1193 | 4.05742E-06 |
| 0.049926758 | -0.049879027 | 409 | 4.77305E-05 |
| 0.120239258 | -0.120247522 | 985 | 8.26404E-06 |
| 0.074462891 | -0.074420763 | 610 | 4.21274E-05 |
| 0.087036133 | -0.087003695 | 713 | 3.24376E-05 |

While Table 5.5 discusses the error due to input approximation, Table 5.6 shows sample of the total error between the exactly mathematical output and the output of ROM_EXP_16 module, taking into consideration both input error and representation error due to a 32 bit output representation. The average output error is found to be 2.5659 E-05.

Table 5.6: Exponential Unit Output Error

| Applied Input | Expected Output | Implemented Output | Output Error |
|---|---|---|---|
| 0.109008789 | 0.896685971 | 0.896722536 | 3.65654E-05 |
| 0.083007813 | 0.920327208 | 0.920343957 | 1.67485E-05 |
| 0.220336914 | 0.802231887 | 0.802248464 | 1.65764E-05 |
| 0.165649414 | 0.847359253 | 0.847343249 | 1.60042E-05 |
| 0.276733398 | 0.758272242 | 0.758256623 | 1.56194E-05 |
| 0.145629883 | 0.86448111 | 0.864477602 | 3.50773E-06 |
| 0.049926758 | 0.951344504 | 0.951299097 | 4.54069E-05 |
| 0.120239258 | 0.886700932 | 0.88670826 | 7.32788E-06 |
| 0.074462891 | 0.928281025 | 0.92824192 | 3.91054E-05 |
| 0.087036133 | 0.916673708 | 0.916643974 | 2.97344E-05 |

Figure 5.21 illustrates a portion of the kernel function evaluation where the first K value is calculated and stored, this portion of the process is repeated 409 time to produce all the desired kernel values, which are needed for the proper voting of the class label.

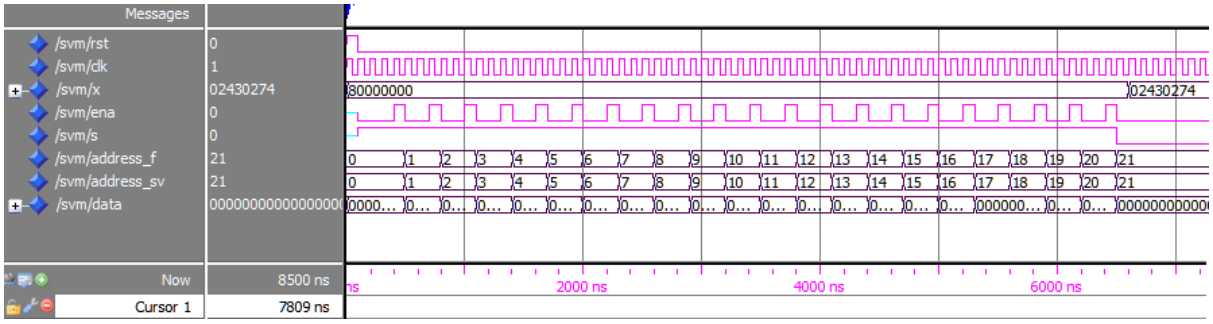


Figure 5.21: RTL Simulation of the Kernel Function Unit

### 5.1.2.2 Integration of the SVM Classifier Unit

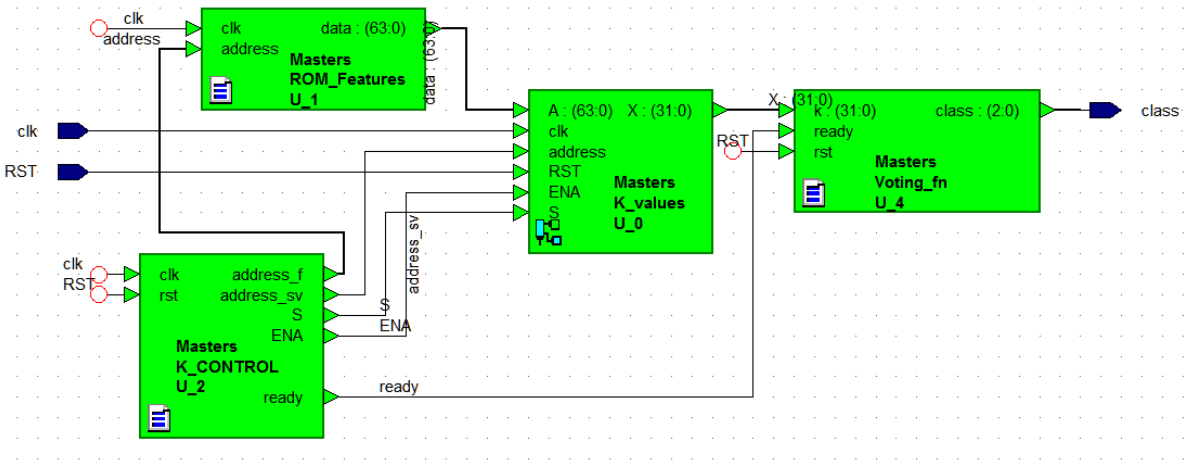


Figure 5.22: SVM Classifier Implementation

The voting function – U3 in Figure 5.17 – is the block where the decision is made to determine which class this epoch belongs to. Figure 5.23 illustrates the operation of the SVM classifier, where the detected class for the current epoch was found to be class 1.

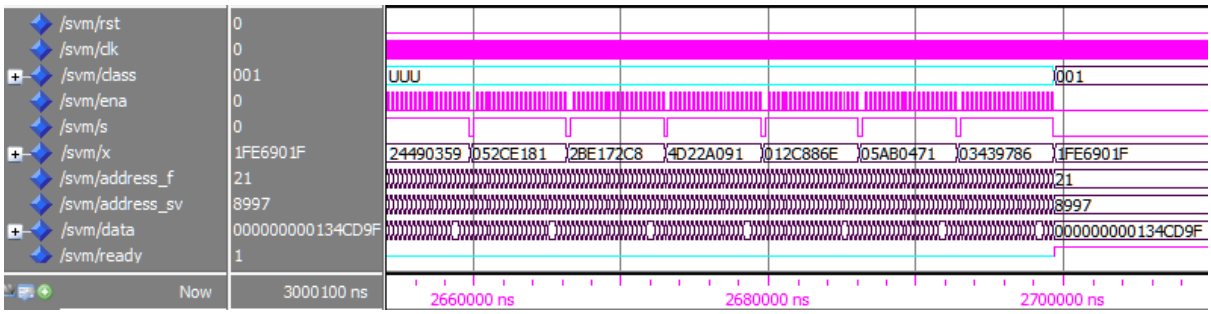


Figure 5.23: RTL Simulation of the SVM classifier unit

### 5.1.3 Narcolepsy Assistive Device System Integration

Once each of the main system constituents is simulated and tested, these unites are integrated as illustrated in Figure 5.24 to form the desired Narcolepsy assistive device. The integrated design consists of the following input ports, clock, serial input, and reset signals; and one output port which holds the current sleep stage class.

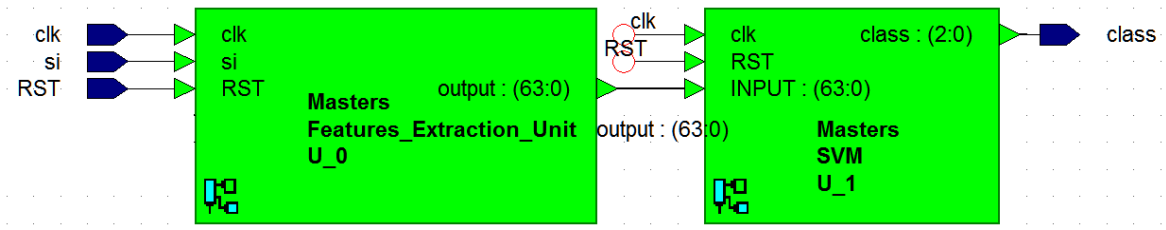


Figure 5.24: Narcolepsy Assistive Device System Integration

A number of narcoleptic EEG recordings – acquired from the database in [26] – are used to test the SVM classifier unit implemented, the number of occurrences for each class are balanced for proper classification and accuracy calculation. Table 5.7 summarizes these numbers, the MATLAB and RTL simulation results as well as their accuracies for each band separately, the average accuracy for the VHDL implementation was found to be 90.5% whereas the software simulation average accuracy is 92.14%.

Table 5.7: Summary of MATLAB and RTL Simulation Result

| Class | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| Number or occurrence (database) | 505 | 438 | 491 | 380 | 501 | 488 |
| Number of detection (MATLAB) | 446 | 410 | 450 | 360 | 468 | 445 |
| Number of Detection (VHDL) | 438 | 402 | 448 | 342 | 435 | 441 |
| Software Simulation Accuracy | 88.31% | 93.6% | 91.64% | 94.73% | 93.41% | 91.18% |
| RTL Simulation Accuracy | 86.73% | 91.78% | 91.24% | 90% | 92.94% | 90.36% |

## 5.2 Hardware Gate Level Simulation

After the hardware RTL design functionality is simulated – as discussed in section 5.1 – gate level simulation is conducted. Gate level simulation results are displayed in the form of charts to visualize all the delays which occur within the design implementation circuit. The delays mentioned consist of the gate delays and the propagation delays. All the gate level simulations

are performed using ModelSim 10.1c and “QuartusII” Version 13. The FPGA Chip selected is Cyclone IV, EP4CGX30BF14C6 from Altera.

### 5.2.1 Gate level Simulation of the Feature Extraction Unit

#### 5.2.1.1 Wavelet Packet Transform Unit

The WPT unit, synthesis, and gate level simulation are conducted on two stages, first the stage includes testing and implementing of the Filter Pair Unit which is the core of the WPT unit. While the second stage concerns the synthesis and testing of the whole WPT unit.

**a) Filter Pair Unit**

The filter pair unit gate level simulation illustrated in Figure 5.25 shows that a total delay of 20ns is needed in order for the correct output to be computed. As it can be seen at 20ns the output, HPF and LPF, of the first data set is ready. Also at 70ns the output of the second data set which are applied at 50ns is in turn computed.

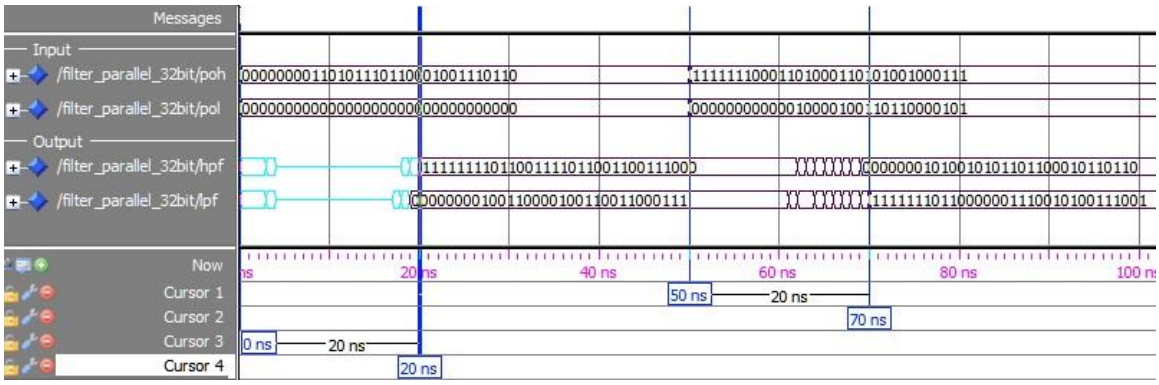


Figure 5.25: Filter Pair Gate Level Simulation result

**b) WPT Unit**

The WPT is tested with the same input stream as simulated in the RTL simulation. Figure 5.26 illustrates a part of this simulation in which the first coefficient of C5 is extracted. The coefficient with the value “007E4063” was extracted at 24.536070msec.

Furthermore Figure 5.27 illustrates the last portion of the gate level simulation, where the final coefficients are extracted. This simulation shows that the total time needed to finish processing one complete epoch is equal to the epoch time window – 30 sec – in addition to the time needed to finish the processing of the last segment of data received, 0.41148ms.

Hence the total time needed is 30.00041148s. The gate level simulation proved to an exact match to the RTL simulation in chapter 4.

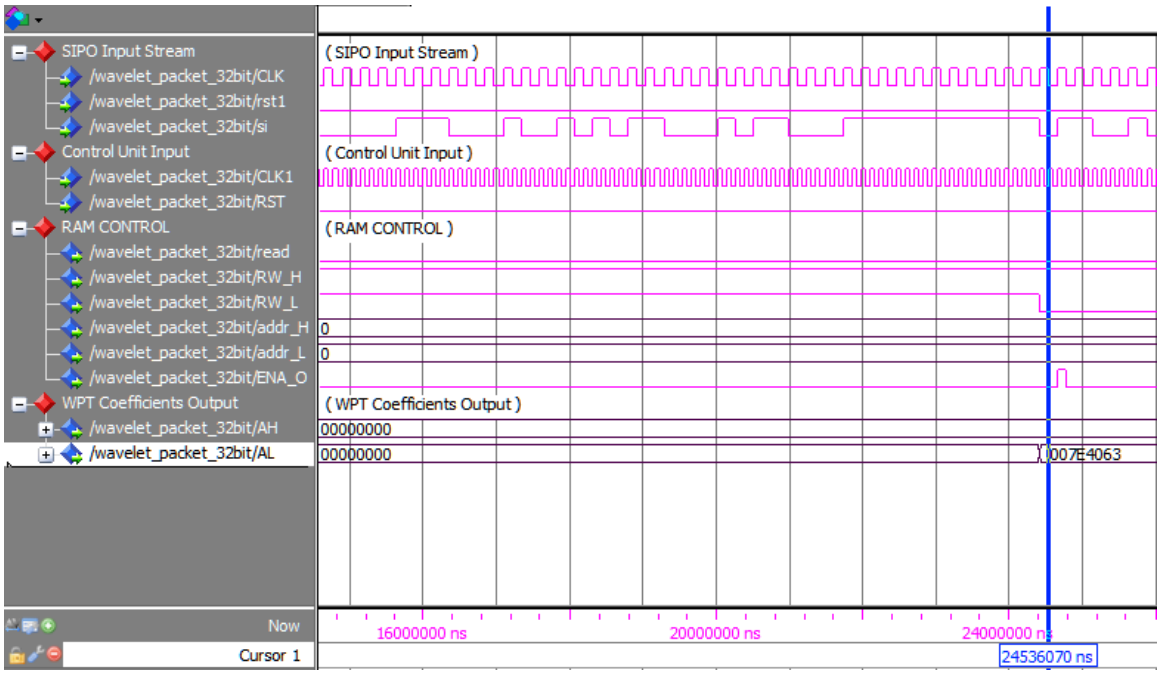


Figure 5.26: WPT Gate Level Simulation for extracting the first coefficient

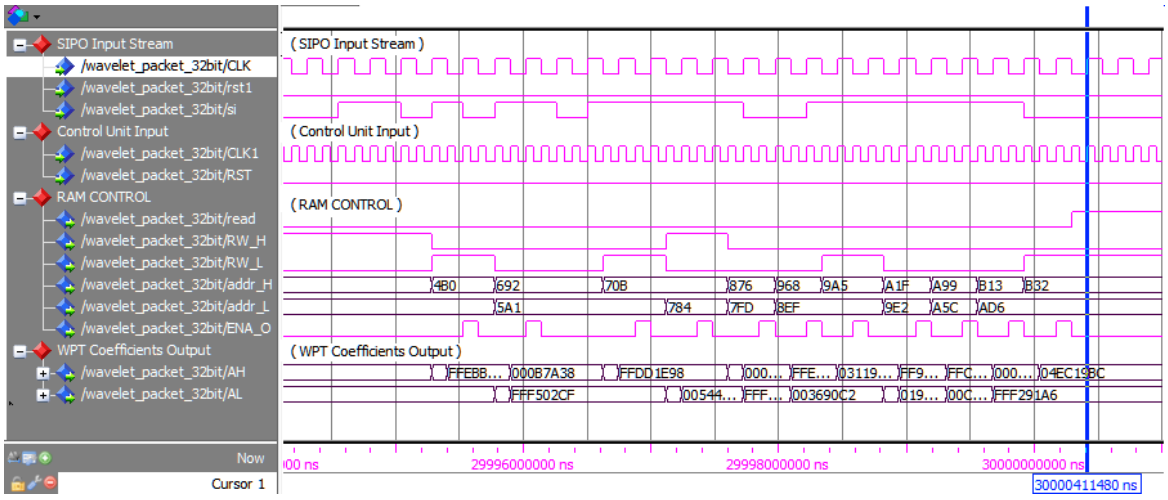


Figure 5.27: WPT Gate Level Simulation for extracting the last coefficient

### 5.2.1.2 Features Calculation Unit

Once the VHDL file is synthesized and routed on the selected FPGA chip, and the delay files are extracted, the gate level simulation is performed. However; in the gate level simulation the internal signals and memories are not available for monitoring. Therefore an additional

function is added to the control unit, to allow the dump of the memory content on a port designed for the sole reason of verifying the memory content.

The simulation illustrated in Figure 5.28 shows that a total time of 1.155ms is required for this unit to complete its main task before the memory dump begins.

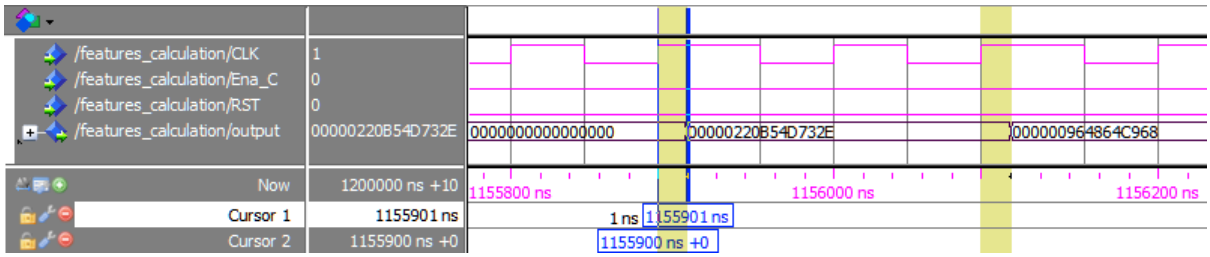


Figure 5.28: Feature Calculation Unit Gate Level Simulation

The calculated features – of the first epoch – after conducting the gate level simulation, are exactly similar to the results acquired by the RTL simulation which are summarized in Table 5.4.

### 5.2.1.3 Features Extraction Unit Integration

Figure 5.29 illustrates the gate level simulation of the Features Extraction Unit integration. This simulation demonstrates that the time needed to process one complete epoch is equal to the sum of the time needed by the WPT Unit along with the time needed to calculate all features by the Features Calculation Unit. The total time is 30.00156 seconds as shown at cursor 1.

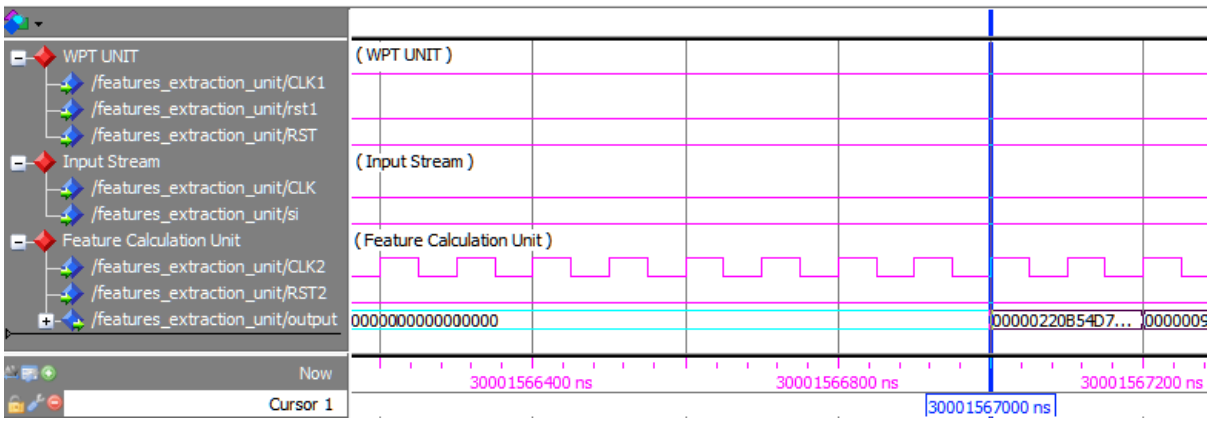


Figure 5.29: Features Extraction Unit Gate Level Simulation Results

The area report of the feature extraction unit is illustrated in Table 5.8.

Table 5.8: Area Report of the Feature Extraction Unit

| | Used Area | Available Area | Percentage |
|---|---|---|---|
| Logic Elements | 18655 | 29440 | 63.36% |
| Memory Bits | 93152 | 1105920 | 8.42% |
| Embedded Multipliers | 32 | 160 | 20% |

### 5.2.2 Gate Level Simulation of the SVM Classifier Unit

The SVM classifier Unit is implemented on the same FPGA Chip. The gate level simulation is conducted for both the Exponential function unit – as it is the core of the kernel function – and the complete unit implementation.

Figure 5.30 illustrates the gate level simulation of the Exponential Function, where the delay for each output can be clearly measured from the following figure. For example the address applied at 101ns, should produce an output at the next following clock rising edge at 200ns as it was illustrated in the RTL simulation in Figure 5.20. However a total time delay of 1ns is found as it is illustrated that the desired output was available at 201ns.

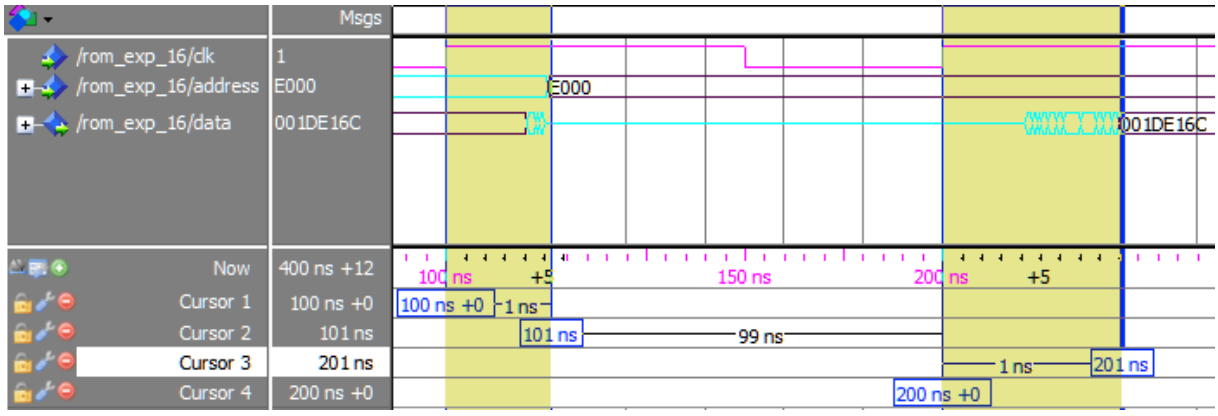


Figure 5.30: Exponential ROM Gate Level Simulation

Figure 5.31 illustrates the gate level simulation of the classifier unit, where the total time needed to reach a decision is 2.7ms. After testing the synthesized unit via the gate level simulation, the results are matched to the RTL simulation results shown in Figure 5.23. The area usage report of the SVM classifier unit is illustrated in Table 5.9.

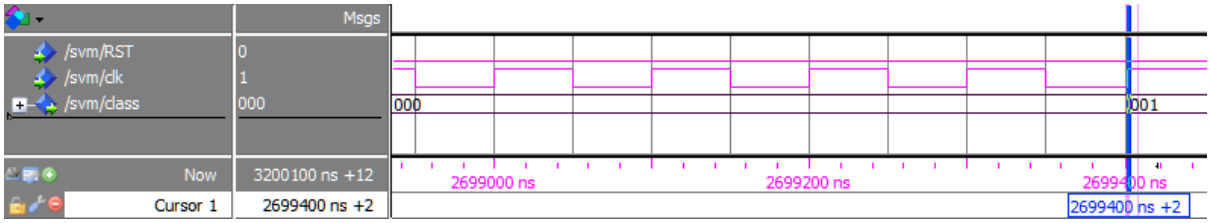


Figure 5.31: SVM Gate Level Simulation Results

Table 5.9: Area Report of the SVM Classifier Unit

| | Used Area | Available Area | Percentage |
|---|---|---|---|
| Logic Elements | 10104 | 29440 | (34.32%) |
| Memory Bits | 0 | 1105920 | 0% |
| Embedded Multipliers | 20 | 160 | (12.5%) |

### 5.2.3 Gate Level Simulation of the Integrated Narcolepsy Assistive Device

Figure 5.32 illustrates the post routing simulation of the Narcolepsy Assistive Device implemented, where it shows that the total time needed by the system to classify one epoch is 30.00426738 seconds. Where the feature extraction unit finishes processing the epoch in 1.567381ms, in order to calculate the desired features. And finally the SVM classifier needs 2.7ms to determine the class of the current epoch.

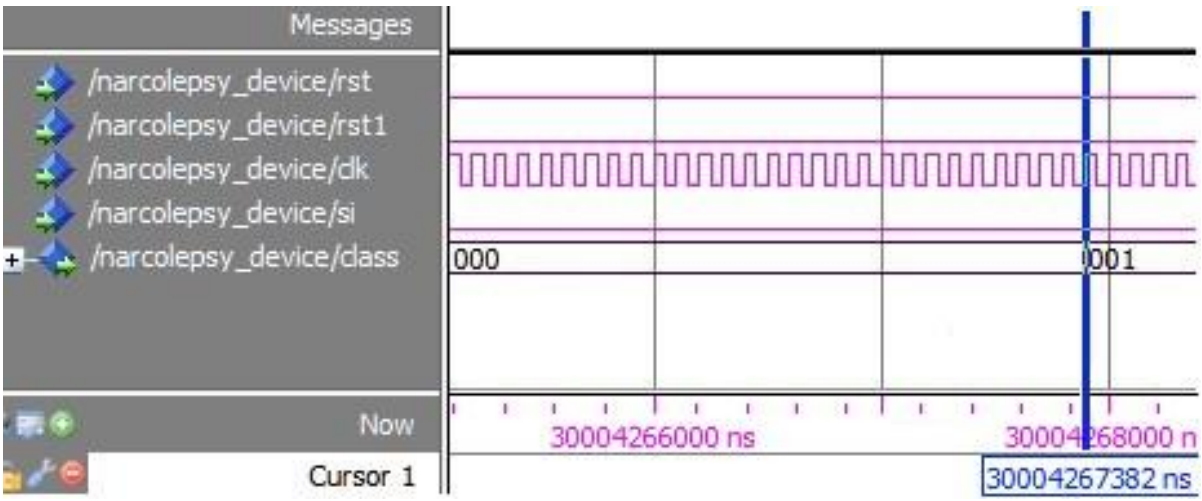


Figure 5.32: Narcolepsy Assistive Device Gate Level Simulation Results

Table 5.10 summarizes the total area used for the overall system implementation; where it clearly shows that the implemented design fits in the selected FPGA chip.

Table 5.10: Area Usage Report of the Overall Implementation

| | Feature Extraction | SVM Classifier | Total Usage |
|---|---|---|---|
| Logic Elements | 63.36% | 34.32% | 97.68% |
| Memory Bits | 8.42% | 0% | 8.42% |
| Embedded Multipliers | 20% | 12.5% | 32.5% |
| IO PINS | | | 7.4% |

# Chapter 6

# Conclusions & Future Work

## 6.1 Conclusions

Narcolepsy – a neurological sleep disorder – is often miss-diagnosed. However it forms a real life difficulty to its patients. In order to overcome the drawbacks of the conventional diagnosis and treatment performed at sleep clinics, and to help narcoleptic patients lead a more productive life, real-time devices have been developed. In this thesis, an alternate hardware implementation is demonstrated. This implementation is characterized by its real-time application, its portability and its high accuracy to guarantee a proper classification and alertness.

This device is designed and simulated using MATLAB version 2014a, and implemented using VHDL. The system implementation is synthesized on a Cyclone IV EP4CGX30BF14C6 FPGA chip by Altera using QuartusII version 13 tool. The RTL and gate level simulations are simulated using ModelSim 6.3a and 10.1c respectively.

This implementation consists of a Feature Extraction Unit and a Classifier Unit. The proposed design for this system introduces the WPT for the Feature Extraction Unit, and the SVM multi-class algorithm for the Classification Unit. Furthermore the RBF kernel is selected for the SVM kernel implementation as it proved to be more effective with the highest accuracy rate among all kernels tested.

A series of simulations and tests are conducted for each of the implemented design constituent units. The data used for testing are acquired from the medical database available at PhysioBank. This database includes a number of PSG recordings for healthy and narcoleptic patients along with the conventional visual scoring conducted by a physician.

The feature extraction unit achieved an acceptable average percentage of error 0.1618% in comparison to the MATLAB software simulation results; while the classifier unit design achieved an accuracy rate of 92.14% and 90.508% in software and RTL simulations respectively.

The implemented system is able to differentiate between the 4 different sleep stages, the wake and the REM states; thus overcoming the need for narcoleptic patients to spend long night hour at sleep clinics for the sleep monitoring.

The implemented system is portable and provides a real-time monitoring, which helps narcoleptic patients to lead a more productive life by raising the patient alertness to the current state of mind.

## 6.2 Proposals for Future Work

As future work, a data logging unit can be implemented providing the physician enough data which can be used to adjust medications. Hardware can be optimized by overlapping the epochs to the maximum time possible – 29.995734 seconds – for faster classification and a better real-time alertness.

Furthermore a complete system download will be conducted as a final testing stage before beginning to design an Application Specific Integrated Circuit (ASIC) implementation for this system.

# REFERENCES


[1] Carskadon, Mary A., and William C. Dement. "Normal human sleep: an overview." Principles and practice of sleep medicine 4 (2000): 13-23.

[2] Rechtschaffen A. Kales A. “A Manual of Standardized Terminology, Techniques and Scoring System for Sleep Stages of Human Subjects” Los Angeles: Brain Information service/Brain Research Institute, 1968.

[3] Carskadon, Mary A., and Allan Rechtschaffen. "Monitoring and staging human sleep" Principles and practice of sleep medicine 3, 2000.

[4] K. Susmakova, “Human Sleep and Sleep EEG” Institute of Measurement Science, Slovak Academy of Sciences – Measurement Science Review, Volume 4, Section 2, 2004.

[5] Shuo Zhang, “Classification of Sleep Staging for Narcolepsy Assistive Device” University of Waterloo, 2007.

[6] F Ebrahimi, M Mikaeili, E Estrada “Automatic Sleep Stage Classification Based on EEG Signals by Using Neural Networks and Wavelet Packet Coefficients” 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society EMBC, 2007.

[7] S. R. I. Gabran, S. Zhang, M. M. A. Salama, R. R. Mansour, C. George “Real-Time Automated Neural-Network Sleep Classifier Using Single Channel EEG Recording for Detection of Narcolepsy Episodes”, 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society EMBC, 2008.

[8] S. R. I. Gabran, W. W. Moussa, M. M. A. Salama “Portable Real-Time Support-Vector-Machine-Based Automated Diagnosis and Detection Device of Narcolepsy Episodes” 31st Annual International Conference of the IEEE EMBS, 2009.

[9] Robi Polikar, “The Wavelet Tutorial Part I: Fundamental the Fourier Transform and the Short Time Fourier Transform”, Rowan University, 1999.

[10] Robi Polikar, “The Wavelet Tutorial Part II: Fundamental Concepts & an Overview of the Wavelet Theory”, Rowan University, 1999.

[11] Robi Polikar, “The Wavelet Tutorial Part III: Multi-resolution Analysis & the Continuous Wavelet Transform”, Rowan University, 1999.

[12] Robi Polikar, “The Wavelet Tutorial Part IV: Multi-resolution Analysis & the Discrete Wavelet Transform”, Rowan University, 1999.

[13] Tristan Fletcher, “Support Vector Machines Explained”, London’s Global University, March 1, 2009.

[14] Kristin P. Bennett, Colin Campbell, “Support Vector Machines: Hypte of Hallelujah?”, SIGKDD Explorations, Vol. 2; Issue 2, December, 2000.

[15] Chih-Chung Chang and Chih-Jen Lin, LIBSVM: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1--27:27, 2011.

[16] Chih-Wei Hsu, Chih-Jen Lin, “A Comparison of Methods for Multi-class Support Vector Machines”, IEEE Transactions on Neural Networks, 2001.

[17] L. Bottou, C. Cortes, J. Denker, H. Drucker, I. Guyon, L. Jackel, Y. LeCun, U. Muller, E. Sackinger, P. Simard, and V. Vapnik. “Comparison of classifer methods: a case study in handwriting digit recognition”, International Conference on Pattern Recoginition, IEEE Computer Society Press, 1994.

[18] S. Knerr, L. Personnaz, and G. Dreyfus, “Single-layer learning revisited: a stepwise procedure for building and training a neural network”, J. Fogelman, editor, Neuro-computing: Algorithms, Architectures, and Applications. Springer-Verlag, 1990.

[19] J. Friedman, “Another Approach to Polychotomous Classification” Technical report, Department of Statistics, Stanford University, 1996.

[20] Vimal Krishnan V. R., Babu Anto P. “Features of Wavelet Packet Decomposition and Discrete Wavelet Transform for Malayalam Speech Recognition”, International Journal of Recent Trends in Engineering, No. 2, May 2009.

[21] Subasi, Abdulhamit. "Automatic recognition of alertness level from EEG by using neural network and wavelet coefficients" *Expert Systems with Applications* 28, no. 4, 2005.

[22] Kiymik, M. Kemal, Mehmet Akin, and Abdulhamit Subasi. "Automatic recognition of alertness level by using wavelet transform and artificial neural network" Journal of neuroscience methods 139, no. 2, 2004.

[23] Ting, Wu, et al. "EEG feature extraction based on wavelet packet decomposition for brain computer interface" Measurement 41, no. 6, 2008

[24] Wang, Deng, Duoqian Miao, and Chen Xie. "Best basis-based wavelet packet entropy feature extraction and hierarchical EEG classification for epileptic detection" *Expert Systems with Applications* 38, no. 11, 2011.

[25] Fatma Guler, N., Elif Derya Ubeyli. "Multiclass support vector machines for EEG-signals classification" Information Technology in Biomedicine, IEEE Transactions 11, no. 2, 2007.

[26] Goldberger AL, Amaral LAN, Glass L, Hausdorff JM, Ivanov PCh, Mark RG, Mietus JE, Moody GB, Peng C-K, Stanley HE. PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. Circulation 101(23):e215-e220 [Circulation Electronic Pages]; 2000 (June 13).

[27] Athira Koranath, Sonali Agrawal. "Comparison of different multiplier algorithms and 1D-DWT as an application" Journal of Electronics and Communication Engineering, 2012.

[28] Jagadeshwar Rao M, Sanjay Dubey. "A High Speed Wallace Tree Multiplier Using Modified Booth Algorithm for Fast Arithmetic Circuits" Journal of Electronics and Communication Engineering, 2012.

[29] Altera, "Internal Memory (RAM and ROM) User Guide", Altera Corporation, Available at: http://www.altera.com/literature/ug/ug_ram_rom.pdf, June, 2014.