**University of Southampton**

Faculty of Physical and Applied Sciences

Electronics and Computer Science

# Bit-serial Artificial Neural Networks for Epilepsy Seizure Detection

by

**Si Mon Kueh**

Thesis for the degree of Doctor of Philosophy

May 2021

UNIVERSITY OF SOUTHAMPTON

FACULTY OF PHYSICAL AND APPLIED SCIENCES
Electronics and Computer Science

## ABSTRACT

Thesis for the degree of Doctor of Philosophy

# BIT-SERIAL ARTIFICIAL NEURAL NETWORKS FOR EPILEPSY SEIZURE DETECTION

by Si Mon Kueh

Fifty million of the world's population are afflicted with epilepsy and 80% of these epileptic patients lives in developing countries. It is crucial to develop a low cost, power saving and reliable home-based seizure detection system for those disabled individuals who have insufficient access to seizure detection equipment.

This research presents three contributions. The first demonstrates that simple bit-serial architecture can be used when designing extremely low-power and low-cost neural network processors to detect epileptic seizures. The proposed design is tailored to be cost effective by employing variable bit precision to allow for compromise between the detection accuracy and the hardware cost.

The second contribution highlights extensive studies of epileptic seizure detection by DPU arrays, using bit-serial neural networks (BSNN) where the control module consists of only simple finite state machines. It has been demonstrated that epilepsy detection through such low-cost and low-energy dedicated neural network is feasible and there is potential for massively parallel network configuration. Different network configurations with variable numbers of network nodes and layers were designed and tested on FPGAs. The best performing version of the complete system has been implemented on an ALTERA Cyclone V FPGA which uses 3931 ALMs with an average recognition rate of 89%.

The third contribution illustrates the development of a dedicated feature extraction component to be used as part of the proposed epilepsy detection system. Two different dedicated feature extraction hardware systems have been designed to provide inputs to the neural network in order to facilitate the classification of EEG waveforms. The EEG features extracted in this research are the slope and mean energy in EEG waveforms. Through multiple experiments, it was found that using a combination of both features as input to the proposed BSNN provides a detection accuracy of 90%.

Results of this research have been published in three conference papers and also in the IEEE Journal on Translational Engineering in Health and Medicine.

# Contents

# List of Figures

# List of Tables

# Abbreviations and Parameters

| | |
|---|---|
| $a$ | amplitude values of EEG signal spikes |
| $ADL$ | Activities of Daily Living |
| $AED$ | Anti-Epileptic Drugs |
| $ANN$ | Artificial Neural Network |
| $ANFIS$ | Adaptive neuro fuzzy inference system |
| $ATPG$ | Automatic Test Pattern Generation |
| $ApEn$ | approximate entropy |
| $AR$ | Autoregressive |
| $ASP$ | Advanced Sensor Processor |
| $ASIC$ | Application Specific Integrated Circuit |
| $ASLAN$ | Automatic methodology for Sequential Logic ApproximatioN |
| $AURA$ | Advanced Uncertain Reasoning Architecture |
| $b_k$ | bias value |
| $BBNN$ | Block-based Neural Network |
| $BFGS$ | BroydenFletcherGoldfarbShanno algorithm |
| $BIC$ | Bayesian Information Criterion |
| $BMBI$ | Brain Machine Brain Interface |
| $BPN$ | Back Propagation Network |
| $BSNN$ | Bit Serial Neural Network |
| $c_i$ | centre of the $i$-th basis function |
| $c_j$ | real number for respective hidden node |
| $CBV$ | Compact Bit Vector |
| $CA$ | Cellular Automata |
| $CMAC$ | Cerebellar model articulation controller |
| $C-RBM$ | Convolutional Restricted Boltzmann Machine |
| $CDNN$ | Convolutional Deep Neural Network |
| $CETs$ | Candidate for epileptiform transients |
| $CGB$ | Conjugate Gradient with Powell/Beale Restarts |
| $CMM$ | Core Correlation Matrix Memory |
| $CNN$ | Convolutional Neural Network |
| $CPLD$ | Complex Programmable Logic Device |
| $CORDIC$ | COordinate Rotation DIgital Computer |

| | |
|---|---|
| $CV$ | cross validation |
| $CWT$ | Continuous Wavelet Transform |
| $DaDN$ | DaDianNao |
| $DMA$ | Direct Memory Access |
| $DNN$ | Deep Neural Network |
| $DPU$ | Data Processing Unit |
| $DWT$ | Discrete Wavelet Transform |
| $DSP$ | Digital Signal Processor |
| $DTC$ | Decision Tree Classifier |
| $d$ | error margin for desired 99% confidence level |
| $d(x_j, x_k)$ | distance between point $x_j$ and $x_k$ |
| $E(v, h; \theta)$ | joint energy of visible and hidden units |
| $EA$ | Evolutionary Algorithm |
| $ECG$ | Electrocardiogram |
| $ED$ | Epileptiform Discharges |
| $EEG$ | electroencephalogram |
| $EN$ | Elman Network |
| $EMG$ | Electromyography |
| $EPSRC$ | Engineering and Physical Sciences Research Council |
| $FA$ | Full Adder |
| $FD$ | fractal dimensions |
| $FDFM$ | Few DSP slices and Few block RAMs |
| $FFT$ | Fast Fourier Transform |
| $FFBP$ | Feed Forward Back Propagation |
| $FIR$ | Finite Impulse Response |
| $FNN$ | Forward Neural Network |
| $FN$ | False Negative |
| $FP$ | False Positive |
| $FPGA$ | Field Programmable Gate Array |
| $FSM$ | Finite State Machine |
| $g$ | equivalent to an activation function |
| $GA$ | Genetic Algorithm |
| $GB$ | GigaByte |
| $GHA$ | General Hebbian Algorithm |
| $GP$ | Genetic Programming |
| $h$ | hidden variable |
| $h'_j$ | target output range |
| $HDMEA$ | high density multiple-electrode arrays |
| $HI-POCT$ | Healthcare Innovations and Point of Care Technologies |
| $HRQOL$ | Health Related Quality of Life |
| $ICANNML$ | International Conference on Artificial Neural Networks and Machine Learning |

| | |
|---|---|
| $JTEHM$ | Journal of Translational Engineering in Health and Medicine |
| $I$ | Query Input Pattern |
| $ICD$ | Implantable Cardioverter Defibrillators |
| $LE$ | Logic Element |
| $LLE$ | Largest Lyapunov Exponent |
| $LM$ | Levenberg-Marquardt |
| $LNA$ | low noise amplifier |
| $LOOCV$ | Leave-One-Out Cross-Validation |
| $MEA$ | Multiple Electrode Array |
| $MLP$ | Multiple Layer Perceptron |
| $mse$ | mean square error |
| $MVML$ | Multi-Views Multi-Learners |
| $MVSL$ | Multi-Views Single-Learners |
| $N$ | Negative sample: Normal Patient EEG |
| $N_{CDP}$ | total number of correctly detected patterns |
| $N_{APP}$ | total number of applied patterns |
| $N_x$ | number of input layer nodes |
| $N_h$ | number of hidden layer nodes |
| $N_o$ | number of output layer nodes |
| $NB$ | Naive Bayes |
| $NLG$ | Neural Logic Gate |
| $NPV$ | Negative Predictive Value |
| $NRMSE$ | normalized root mean square error |
| $O$ | Output Pattern |
| $OA$ | Overall Performance |
| $OAT$ | Optimal Allocation Technique |
| $P$ | Positive Sample: Epileptic Patient EEG |
| $P(C_i\|Y)$ | posterior probability |
| $P(C_i)$ | prior probabilities |
| $P(Y\|C_j)$ | posterior probability of Y |
| $P(c\|x)$ | posterior probability of a target given specific attribute |
| $P(c)$ | prior probability of the target |
| $PEMS$ | Predictive emission monitoring systems |
| $PID$ | Proportional-Integral-Derivative |
| $POMS$ | Profile of Mood State |
| $PNN$ | Probabilistic Neural Network |
| $PPV$ | Positive Predictive Value |
| $\psi$ | optimal wavelet basis function |
| $P(x)$ | prior probability of an attribute |
| $P(x\|c)$ | probability of an attribute given the target |
| $P(Y)$ | prior probability of $(Y)$ |

| | |
|---|---|
| $QOLIE-89$ | Quality of Life of Epilepsy questionnaire |
| $RAM$ | Random Access Memory |
| $RBF$ | Radial Basis Function Network |
| $RBM$ | Restricted Boltzmann Machine |
| $ROM$ | Read Only Memory |
| $RR$ | recognition rate |
| $RTL$ | Register Transfer Level |
| $s$ | scale |
| $SDNN$ | Space Displacement Neural Network |
| $SNN$ | Spiking Neural Network |
| $SOM$ | Self-Organizing Map |
| $SSSS$ | Small System Simulation Symposium 2018 |
| $STFT$ | Short Time Fourier Transform |
| $STR$ | Stripes |
| $\tau$ | shift |
| $TE$ | Tenesse Eastman |
| $TN$ | True Negative |
| $TP$ | True Positive |
| $th$ | threshold value |
| $TLE$ | Temporal Lobe Epilepsy |
| $TPR$ | True Positive Rate / Sensitivity |
| $TNR$ | True Negative Rate / Specificity |
| $u$ | sum of products of the BSNN |
| $v$ | visible variable |
| $v_{i,j}$ | weight of each connection |
| $VLSI$ | Very Large Scale Integration |
| $VNS$ | vagus nerve stimulation |
| $w_n$ | weight for number, n neuron |
| $WASET$ | World Academy of Science, Engineering and Technology |
| $WHO$ | World Health Organisation |
| $WSNs$ | wireless sensor networks |
| $X$ | real number |
| $X_{Mean}$ | mean |
| $X_{Median}$ | median |
| $X_{Mode}$ | mode |
| $X_{StdDev}$ | standard deviation |
| $X_{Q1}$ | first quartile |
| $X_{Q3}$ | third quartile |
| $X_{IQR}$ | inter-quartile range |
| $X_{skew}$ | skewness |
| $X_{kurtosis}$ | kurtosis |

| | |
|---|---|
| $X_{Min}$ | minimum |
| $X_{Max}$ | maximum |
| $x_n$ | input x for number, n neuron |
| $\xi(t)$ | Seizure Spike Representation |
| $y$ | output from activation function / output |
| $Z(\theta)$ | normalisation constant |
| $z$ | value used to achieve desired 99% confidence level |

# Chapter 1

# Introduction

## 1.1 Background Information

In 2018, the World Health Organization (WHO) statistics revealed that 50 million of the world's population is affected by epilepsy [1]. Approximately 80% of the reported epileptic cases are found in developing countries. These countries may not have readily available treatment facilities and medications. Many epileptic cases are not reported especially in developing countries because epileptic patients and their families are afraid of being stigmatised and discriminated upon. This research addresses the problem of detecting epileptic seizures that affects these individuals around the world. Epilepsy is caused by abnormal impulses generated in the brain, the most complex part of the human body. It is well known that only 10% of the $10^{13}$ cells are involved in the information processing and communicating part of the brain [1]. The cells that make up the most integral part of the human body are neurons. The brain itself is enclosed in a skull, and protected by a dura matter which is the dense protective fibre like layer. The brain consist of three main parts: cerebrum; cerebellum; brain stem.

Currently, epilepsy treatments are usually provided in the form of anti-epileptic drugs (AEDs) [2]. In the 1920s, the ketogenic diet was popularised as an alternative form of treatment for epilepsy. This diet was popular among children with epilepsy as it had a success rate of 30% to 50% which was good in those days. Another popular form of treatment was the vagus nerve stimulation (VNS) technique which was used to treat problematic seizures. In 1997, this device was approved by the U.S Food and Drug Administration unfortunately it is less effective than AEDs as only 50 percent of the 40% of patients treated respond to the treatment [3]. In 1970, the development of seizure prediction research was started by Viglione and colleagues. Seizure prediction analyses have several different categories: time-domain analysis; frequency-domain analysis; non-linear dynamics [4]. Most of the previous work will be explained in more detail in Chapter 2. Unfortunately, nowadays there is still no home-based seizure detection system that can specifically differentiate between epileptic and non-epileptic seizure electroencephalogram (EEG) patterns.

To address the problem of epilepsy seizure detection, this thesis has reviewed different state of the art seizure detection methods. These methods are categorized as linear methods and non-linear methods. Linear methods have the advantage of simplicity and versatility compared with non-linear methods that are more capable of addressing the non-stationary nature of the EEG signals. It was decided to use one of the linear methodology to develop the epilepsy detection system which requires lower computational power [5]. The Artificial Neural Network (ANN) is a form of classifier that works in conjunction with feature extraction for epilepsy detection. ANN provides a more reliable seizure prediction with a range of error from 89 to 100%, verified with the datasets tested in this research. Thus, ANN have been chosen as the basis of the classifier designed in this thesis to overcome the problem of epilepsy detection.

Furthermore, the research project known as Ambient Assisted Living (AAL) have been of interest internationally in recent years [6],[7]. Through technological advancement which contributes to various smart objects with the capability of identifying, locating, sensing and connecting and thus leading to new forms of communication between people and things. Ambient Assisted Living (AAL) involves the use of various technical systems to support elderly people in their daily routine to allow an independent and safe lifestyle as long as possible. Personal communication between elderly people, their environment and relevant groups of care givers is an important aspect in AAL. Therefore, it is possible that epileptic patients also can benefit from the use of AAL in their daily lives. In order to accomplish such a feat, there is the need of identifying the best architecture may it be hardware plus software or just software for this seizure detection system. The thesis hopes to incorporate a viable seizure detection system into the AAL ecosystem and possibly future smart homes.

## 1.2 Specification and Approach

This section provides the approach and specification of this research. It should be noted that other literature have used the assumption that EEG is generated by a highly complex linear system [8]. This assumption is also being used in our research.

### 1.2.1 Why a bit-serial neural processor design?

In order to develop a wearable, home-based seizure detection system, the size and energy efficiency issue of the system will need to be addressed. The bit-serial architecture was chosen as the basis of this design because this method is generally perceived and recognised as a method of choice for producing low cost and low power processors. When compared with the alternative which is the bit-parallel approach which prioritize speed over cost, the bit-serial architecture methodology is preferred for developing a neural processor for this research as a wearable system needs to prioritize cost over speed.

### 1.2.2 Why an artificial neural network?

A neural processor would be the ideal design to simulate a biological neuron as it is also commonly used in brain modelling [9]. As part of the epilepsy detection system, the performance and speed of the system can be compromised to obtain a minimum hardware cost. It is expected that the small design developed in this research would be more energy efficient than a general processor and therefore used for this particular application. In this research, the developed system is mainly implemented on a Field Programmable Gate Array (FPGA) board. It is anticipated that in the near future, it will be necessary to completely design and fabricate an ASIC model of this neural hardware. It will then be fully tested to gauge its energy efficiency and feasibility to be used by epilepsy patients during their daily activities.

### 1.2.3 Specification of our design

At present, there is still no standard specifications for good epileptic seizure detection. Therefore, this research has set specifications based on reference literatures i.e. the paper by Painkras et al. [9] which is closely related to this area of studies. The design proposed in this research have met certain specifications required for epileptic seizure detection. As the design decision involves a wearable system, the size of the proposed system should be less than 20% of the provided hardware resources of an FPGA chipset. In this research, comparisons have been made using three different chipsets, including Cyclone IV, Cyclone V and Stratix IV. As the research is still a proof of concept, evaluation of the results mainly focus on correct recognition rate of EEG patterns and hardware cost. Estimation of power and latency have also been included. However, the author strongly advised that these are estimated values which can be further optimised.

In order to compete with designs shown in other literature, the average accuracy of the proposed system must be over 80% as to convince experts world wide that this device can successfully detect epileptic seizure patterns. This specification is reinforced in a paper presented by Raygoza-Panduro et al. [10] which have an accuracy of 80%. The paper has been accepted and published by IEEE.

### 1.2.4 Hardware and software used for simulation and hardware testing

Software simulation have been proven to be faster and more accurate for ANN testing. However, software simulation requires the use of a laptop with an i5 processor, 8 GB RAM and a 64-bit operating system which is both bulky and cumbersome. In contrast, the proposed hardware is portable and can be used without the need of a connected laptop. In this research, an FPGA board have been used for hardware implementation. Synthesis for this design was conducted suing an Altera Quartus II software.

## 1.3 Aims and Contributions

In this section, the author wishes to address the research aims and objectives that led to the completion of this thesis. This research was started on the premise that it is vital for an epileptic patient to have a early seizure detection device that will prevent any unnecessary injuries or accidents. This objective then led to the various questions of developing a wearable seizure detection system. In order to complete this task, the author have established a few research aims that were addressed in this thesis. Firstly, the wearable seizure detection system aims to be developed using simple hardware as to minimize the hardware cost and power consumption. Secondly, this device must also run on an algorithm that provides an acceptable compromise between speed and accuracy (recognition rate). Finally, the research aims to be a proof of concept for future research such as smart homes and ambient assisted living.

Over the course of this research, three different contributions were made and published [11, 12, 13]. The completed system designs are tested and synthesised on FPGAs. The complete epilepsy detection system is illustrated in the Healthcare Innovations and Point of Care Technologies (HI-POCT) revised journal paper.

1. **Proposal of a low-cost and low-energy data processing unit (DPU) [11]**

   At the beginning of this research, a detailed literature review was done in order to fully comprehend the state of the art epilepsy detection systems as well as the basic principles of their application. This literature review is included in Chapter 2 of this thesis. It was found that most epilepsy detection systems analyses EEG signals using software implementation techniques which consumes high computational power and therefore is not desirable for use in a portable, home-based epilepsy detection device. Thus, the research proposed a prototype low-cost and low-energy portable epilepsy detection system using bit serial ANN.

   The data processing unit (DPU)was proposed to fully implement the functionality of a biological neuron. This proposed DPU found in Chapter 3 is based on a bit-serial architecture which has the capability of minimizing the hardware cost in terms of logic elements (LE) needed for the dedicated neural hardware when compared with other bit-serial architecture processors. The advantages and disadvantages of these proposed bit serial techniques and hardware is analysed in the critical analysis section, section 2.12.

2. **Development and testing of dedicated hardware ANNs [11, 12]**

   The research progressed further to develop a fully functional ANN using the proposed dedicated DPU [11] in Chapter 4 . The various ANN configurations developed in this research are based on bit-serial architecture. The first set of experiments involves the implementation of a single neuron accurately. The developed ANNs is known as the bit-serial neural network (BSNN). The BSNN uses identical DPU as each network nodes. Each layer of the network will have a number of the DPUs aligned in a vector arrangement and controlled easily using simple finite state machines (FSM). Next, the BSNN have a central controller FSM. This removes the need of complex program and further reduces the necessary hardware cost.

   As part of the verification process of testing the functionality of the DPU, various forms of simple ANNs have been designed and experiments were conducted. In these tests, EEG input have also been used to test the BSNN. First, the tests used to detect epilepsy involve a single neuron with multiple inputs. Furthermore, multiple hidden neurons were used to further improve the accuracy of the network. Multiple experiments were also conducted using a number of hidden layers to find the best network configuration.

   The proposed hardware neural network was synthesised on different FGPAs as a form of comparison and to find the best development chip for our design. The hardware cost of each system are shown in section 4.4. Different bit-precision were also used as a method of choosing the best trade off between performance and size. The tests are then evaluated with different metrics such as sensitivity (TPR), specificity (TNR), positive predictive value (PPV) and negative predictive value (NPV). Further details of these experiments can be found in Chapter 4.

3. **EEG feature analysis and implementation of the optimal BSNN design [11]; [12]; [13]**

   In order to fully implement the epilepsy prediction system, a feature extraction component was designed to provide the inputs to the BSNN. Two different feature extraction hardware were designed and multiple experiments have been conducted to test for an optimal feature extraction component. The first component is a simple slope calculator. The other component calculates the mean energy value of a single EEG window.

   Experiments conducted using the slope calculator have an accuracy over 85% while the mean energy system have a 62% accuracy. Chapter 5 discuss the reason behind such a low accuracy for the mean energy system. A combination of both feature extraction hardware has a 90% accuracy which only accord a 2% improvement when compared with the slope calculator system. With this in mind, the slope calculator system was preferred.

The network configurations used in our experiments were closely based on a recent work that used ANNs in their classification process [14]. The results were then closely compared with a suitable EEG benchmark in section 5.4. In order to keep human error to a minimum when writing the hardware codes, this research provides an automated method to complete the hardware codes for a massive parallel neural network with simple Python scripts. Section 6.1 presents some further work which would be to improve and implement this wearable epilepsy detection system physically for an epilepsy patient. This involves synthesis of the physical layout of the DPU and the full system using an ASIC technology. This will help in estimating the area and power needed for the dedicated neural hardware system.

## 1.4 Thesis Organization

This thesis is broken down into several chapters. Chapter 1 introduces the problem and the reason for conducting this research, the different challenges, a brief explanation of any previous work and address the contributions achieved in this research. Chapter 2 is the literature review which discusses research conducted by various groups over the past decades. It gives details of the state of the art seizure detection technology, neural processors, different forms of neural networks and other useful information. Chapter 3 then proposes a novel approach with a new bit-serial data processing unit (DPU) design along with various tests that were conducted to test the functionality of the processor. Chapter 4 presents a novel bit-serial neural network (BSNN) using the DPU proposed in Chapter 3 for a simple and wearable epilepsy detection system. Multiple experiments were also conducted to find out the feasibility and efficiency of the BSNN. The cost of various BSNN configurations is also discussed. Chapter 5 includes EEG feature analysis and BSNN optimization. This chapter presents two dedicated extraction hardware that is incorporated into the complete epilepsy detection system. Chapter 6 concludes the thesis, explains associated limitations of this research and presents future possibilities of this research.

# Chapter 2

# Background Research

This chapter reviews previous/ongoing research in epilepsy detection. As EEG waveforms are mainly used to detect and diagnose patients with epilepsy, section 2.1 briefs the reader on the EEG research related to epilepsy detection. Section 2.1.1 then presents different EEG waveform analysis methods used in the EEG research. Section 2.2 then presents a thorough analysis on different conventional epilepsy detection methods. It is crucial to review this topic in order to improve existing techniques or develop new equipment. This chapter also include a general review of existing parallel learning systems in section 2.3 which have provided inspiration for this research. Section 2.4 presents different artificial neural network (ANN) used in software implementation as part of the process to find the best ANN solution for the hardware implementation conducted in this research. Section 2.5 gives a review on Deep Learning Neural Network which is used as comparison against the research proposed design. Section 2.6 presents different state of the art neural processors which allows the research to find inspiration and weakness that can be exploited. Section 2.7 and 2.8 further explains the potential of ANN in prediction applications along with the state of the art real time hardware devices. This gives the research more opportunity to draw inspiration from the different papers. As part of the specification of the research, section 2.9 gives a review on bit-serial architecture and the hardware that have been developed in this area. Section 2.10 and 2.11 presents certain related work that the research have used as inspiration for hardware neural network.Critical analysis of these reviewed methodologies is presented in section 2.12.The tables in that section are included to allow the reader to fully comprehend the work that has been reviewed. Through the reviewed literatures discussed in this chapter, this research was able to develop a fully parallel bit-serial neural network (BSNN) for epilepsy detection. A summary of this chapter is then included in Section 2.13.

## 2.1 EEG Research in Epilepsy Detection

In general, an EEG signal is a non-stationary biomedical signal. This signal is also used to determine physiological and psychological activity in the brain [15, 16, 17]. EEG analysis is considered to be the most common methodology when trying to detect epilepsy in the medical world. Epilepsy can be characterized by recurrent seizure spike patterns in the EEG signal. There are a few specifications of an EEG signal that can be useful during a seizure event. From an original EEG wave, four sub signals (delta, theta, alpha and beta) are extracted. Alpha wave (8-12Hz) is the natural frequency of the brain [17]. During an epileptic seizure, the delta (0-4Hz) and theta (4-8Hz) waves have unique characteristics which present as low frequency and high magnitude waveforms. The brain also produces beta waves that have low magnitude and a higher frequency (>13Hz) compared to the other waves [17].

EEG signals emitted on the outer layer of the cerebral cortex are recorded by positioning the electrode on different locations on the scalp of the brain. This procedure is used to determine the brain activity involved [18, 19]. When EEG is used, a healthy patient's EEG scan [20] presents itself in the form of low-voltage spikes. However, these spikes increase in magnitude in particular areas during the occurrence of a seizure. During a seizure event, rhythmic and sharp spikes are recorded. It was also found that the ictal or EEG signals during seizure events is very different from normal brain activity with reference to frequency and neural firing patterns [21]. Interictal signals are part of the EEG signal between epileptic seizures [22].

Different forms of waveform analysis have been researched and formed to analyse EEG in recent decades. The wavelet-chaos analysis has been used specifically to analyse the EEG sub-bands, in order to determine possible parameters which can be used in seizure and epilepsy detection [21].

The traditional procedure of analysing an EEG scan would require expensive personnel where a specialist is needed to review the whole recording of the EEG signal. This method takes time. As part of the ongoing research into epilepsy detection, an automatic seizure identification method will be preferable in this area. Different techniques for EEG analysis have been considered, such as Wavelet Transform and Autoregressive (AR) modelling. These methods have superior resolution for short data segments, and these methods have the advantage when real-time data processing is required.

### 2.1.1 EEG Waveform Analysis Methodology

There are a few state of the art waveform analysis methods which include Short Time Fourier Transforms, Wavelet Transforms, Lyapunov Exponent, Autoregressive Modelling etc. The frequency components of an EEG signal can be extracted using Short Time

Fourier Transform (STFT) as the basic Fast Fourier Transform (FFT) method suffers from large noise sensitivity [23]. Next, the magnitude of the signal is measured using electrodes placed on the surface of the scalp. The average sum of electric potential emitted by the group of neurons is recorded by specific placement of the electrodes [20].

With the Rosenstein algorithm, the Largest Lyapunov Exponent (LLE), rate of separation of infinitesimally close trajectories for the EEG signals can be calculated. It is a measure of the sensitivity dependence on certain initial conditions as well as a quantitative measure of the chaotic characteristic of the EEG signal. This algorithm can then be combined with a fuzzy-logic based system which enables the detection of an epilepsy seizure event [24]. The wolff algorithm, mainly used for Monte Carlo simulation of the Ising model is the other alternative algorithm that can be used here. However, this algorithm is very sensitive to noise in time series and degree of measurement. Thus, the research [24] chose to use Rosenstein algorithm which utilizes the method of delays. It is also very accurate even when there are changes related to the following quantities: embedding dimension; time delay; divergence of nearest trajectories; noise level; the size of the datasets being used for analysis. This is accomplished with the use of least-squares to fit a line to the data [25].

The recorded EEG signals can also be divided using low and high wavelet coefficients and these are further divided into high and sub low coefficients. Akmin *et al.* [17] presents the theory for wavelet transform in their recent work. They proposed the method of acquiring a dataset using PCI-MIO-16-E4 for computer-based analysis (202 samples in 6 seconds). In their work, the assume that a stationary signal is a signal that does not change over a long period of time. This allows the team to apply Fourier Transform to the obtained signal. Brain activity recorded in EEG signal displays a combination of many non-stationary or transitory characteristics. Wavelet transform employs the use of STFT that studies a small section of the signal which maps the signal in a 2D function of time and frequency (Hz).

Wavelet Transform can be automated to identify the epileptic features within the EEG signals. The optimal wavelet basis function ($\psi$) is first designed using genetic algorithm (GA) to adapt the spikes of the EEG signals through a series of function of scale ($s$) and shift ($\tau$). This function can also be used as matching filters for identifying seizure spikes represented by the $\xi(t)$ in equation 2.1. The seizure spikes are extracted from the EEG recordings with Wavelet Transform and threshold-based estimation [26]. This method is applied and evaluated using different clinical samples of real EEG data of epileptic patients. The group obtained both a high sensitivity and selectivity over 90%. The implementation was conducted using MATLAB and the stopping criteria was based on a high number of generations (i.e. number of iterations). The rationale for using this method is to locate a convergence, and this will give an optimum fitness function value using function (2.1).

$$f(\tau, s) = \int |\xi(t) - \psi(\frac{t-\tau}{s})| dt \tag{2.1}$$

A combination of GA and genetic programming was proposed for EEG analysis [26]. In a recent work [27], Genetic Programming (GP) was used to develop an ANN in order to solve complex problems. This process removes the need for human participation, but further hardware based research is needed before any feasible clinical trial can be conducted. The hardware research is the main focus in many core applications with the implementation of specific processors.

In 2006, Sarang [28] considered three methods of Wavelet Transform application to detect EEG signal spikes. These methods are:

1. Complex Continuous Wavelet Transform
2. CWT via multi scale view
3. Discrete Wavelet Transform

Autoregressive (AR) modelling methodology increases the resolution of the EEG spectrum using the assumption that the EEG signal continues to some extent outside the EEG window. Next, AR modelling also reduce spectral leakage by applying the use of smoothing windows rather than finite sampling record. Further, the data records of autoregressive modelling is shorter as compared to FFT [23]. The optimum order of an AR model is determined by the Bayesian Information Criterion (BIC) and the AR parameters of an EEG signal. The sub-bands are based on a paper written by Mousavi [29]. Extracted parameters are used as a feature to categorize the EEG signal by the group using the multilayer perceptron (MLP) classifier. The output signals are categorised as different signals: healthy (normal EEG), interictal (EEG signal between epileptic seizure events) and ictal signals (seizure event EEG signals).

Another recent work uses a Hebbian eigenfilter with a General Hebbian Algorithm (GHA). The few main contributions of this research are the data streaming method, a stream-based General Hebbian Algorithm and a new learning kernel architecture. The stream based method and the hardware architecture are being evaluated thoroughly [30]. This approach is based on an assumption that the recorded neural activities are considered infinite and stable over a long period of time. This assumption is necessary for automated epileptic detection. However, it is not a realistic assumption as EEG signals are not stationary.

An original Automatic Test Pattern Generation (ATPG) algorithm was proposed by Chakradhar *et al.* [31] which was based on stochastic neural networks. Certain parameters and stochastic operations used in the ATPG algorithm were not used in the new algorithm proposed by Masatoshi Arai *et al.* [32] which is based on strictly digital neural network (SDNN). In this case, the author and their team managed to develop a new logic circuit to obtain a preliminary set of test patterns. This method is quite efficient when it comes to large scale problems. Neural action potential, or "spike" detection, is a necessary first step in neural recording. In 2015, Y. Yang *et al.* [33] proposed an approach to reduce the amount of neural data by compressing the entire neural signal on-line, and by reconstructing complete neural waveform off-line for any form of data processing process. This will reduce processing time and the needed computational power.

## 2.2 Conventional Classification Methods for Epilepsy Detection

This section reviews different conventional classification techniques for machine learning, which are used for medical diagnosis applications, including epilepsy. These methods are Naive Bayes (NB) classifier, decision trees, k-nearest-neighbors (k-NNs) and logistic regression. These techniques are analysed and reviewed in critical analysis section 2.12. The logistic regression model is a special case of a linear regression classifier that utilizes a linear function [34].

The NB classifier is a simple probabilistic classifier which utilizes the Bayes Theorem. This theorem uses the probability of certain causes and their conditional probability as a technique to compute the conditional probability of each possible causes for a given observed outcome. Thus, it is considered as a conditional probability model. NB classifier uses the independence assumption that focus on each feature independent of each other, while ignoring any possible correlation between the different features [35]. This assumption has been widely criticized as unrealistic. The advantage of using an NB classifier in medical data mining is the limited use of training data for classification. The equation 2.2 and 2.3 are used for classification when using a NB classifier. The parameters in the equations are the prior probabilities $P(C_i)$; $P(Y)$ the prior probability of $(Y)$;$P(C_i|Y)$ is the posterior probability; $P(Y|C_i)$ is the posterior probability of Y that depends on the condition, $C_i$

$$P(C_i|Y) > P(C_j|Y) \text{ for } i \leq j \leq n, j \neq i \tag{2.2}$$

and

$$P(C_i|Y) = \frac{P(Y|C_i)P(C_i)}{P(Y)} \tag{2.3}$$

It is also applicable in automated medical diagnosis when it is used to diagnose different medical problems. The NB classifier outperforms the other algorithms that were being used in this comparative analysis [35]. The example with relation to our research is provided below. Table 2.1 includes a few EEG samples taken from publicly available EEG data [36]. The classification for each feature, Spikes, Noise and Chaotic were made using human observation. The decision to use human observation is to emulate a specialist observing an EEG recording.

| **Example EEG** | **Spikes** | **Noise** | **Chaotic** | **Seizure / Free Seizure** |
|---|---|---|---|---|
| 1 | Large | Weak | Less | Seizure |
| 2 | Large | Weak | Less | Seizure |
| 3 | Small | Dominant | More | Free Seizure |
| 4 | Small | Dominant | More | Free Seizure |
| 5 | Large | Weak | More | Seizure |
| 6 | Small | Weak | More | Seizure |
| 7 | Small | Dominant | More | Free Seizure |
| 8 | Small | Weak | More | Free Seizure |
| 9 | Large | Dominant | Less | Seizure |
| 10 | Large | Weak | Less | Seizure |

Table 2.1: EEG sample data related to thesis research using NB classifier [36]

The test example to be classified is a large spike, with dominant noise but more chaotic EEG signal. We use equation 2.2, 2.3 to classify this example. First, these probabilities need to be calculated.

| **Description** | **Value** |
|---|---|
| $P(Large\|Seizure)$ | 0.78 |
| $P(Dominant\|Seizure)$ | 0.75 |
| $P(More\|Seizure)$ | 0.4 |
| $P(Large\|FreeSeizure)$ | 0.5 |
| $P(Dominant\|FreeSeizure)$ | 0.63 |
| $P(More\|FreeSeizure)$ | 0.6 |
| $P(Seizure)$ | 0.5 |
| $P(FreeSeizure)$ | 0.5 |
| $P(Seizure)*P(Large\|Seizure)*P(Dominant\|Seizure)*P(More\|Seizure)$ | 0.117 |
| $P(FreeSeizure)*P(Large\|FreeSeizure)*P(Dominant\|FreeSeizure)*P(More\|FreeSeizure)$ | 0.0945 |

Table 2.2: Table displaying a simple example of using NB classifier in the context of epilepsy detection

Table 2.2 shows that $0.0945 < 0.117$ meaning that the example EEG signal will be classified as a seizure signal. From this simple example, the NB classifier provides an inspiration for our research. The dominant feature in this simple example is the spikes which is used as the main focus for epilepsy detection in this thesis.

The NB classifier operation has three stages. Firstly, the likelihood $P(x|c)$ is estimated during the training stage with the use of two forms of training sample, i.e a epileptic seizure EEG and a normal EEG. Secondly, an output decision will be made in the testing phase using the posterior probability $P(c|x)$. Thirdly, a high probability will classify the training sample as an epileptic waveform and vice versa. This acts as a basis to develop a more complex NB model.

The equation 2.4 can be simplified with the assumption that the features that are being classified are conditionally independent of each other in order to reduce the required amount of computational power. The parameters in equation 2.4 are $P(c|x)$, the posterior probability of a target given the attribute; $P(c)$, the prior probability of the target; $P(x|c)$, the probability of a attribute given the target; $P(x)$, the prior probability of an attribute.

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)} \tag{2.4}$$

This particular classifier has a very high accuracy when dealing with independent attributes or features. However, there are two main disadvantages. Firstly, the classification is made on a strong assumption that the features used are independent of each other which will affect the results if the features are not independent. Secondly, information might be lost in the process of making continuous features discrete.

The intended research outcome is the design of a simple and low cost hardware, this classifier model would not be a suitable choice as equation 2.4 is a rather complex equation to be implemented in hardware. This can be attributed to the need for multiple multipliers and dividers. Furthermore, the variables of the equation 2.4 require pre-processing when extracted from the original EEG waveform. However, this thesis requires the use of features that can be extracted easily from the EEG signals.

In machine learning, there is the decision tree classifier (DTC) which uses search heuristic model for prediction. For an example, we shall use a simple problem of deciding whether to play or not to play based on the weather situation. The classification shall be conducted using DTC.

The main difference between a decision tree and the NB classifier is that decision tree can classify directly using tabular data but NB classifier require the need of manual feature selection. This type of model uses recursive partitioning algorithms to distinguish subsets of specific data from the original data set. By increasing the number of splits between the partitions, there will be an increase information gain [34].

The DTC has a lot of potential when used as an efficient way of classifying different sets of data. It has a shorter training time than a multilayer perceptron (MLP) solution. This thesis train the proposed ANN design off-line similar to that of the DTC. As compared to other classifiers, there will be no need to conduct the training in real time. It is also specified in a recent work [37] that neural networks are used in the design of a DTC, which, proves that the basic design of this high level classifier would still need the hardware proposed in this research. The general structure of a decision tree is illustrated in Figure 2.1

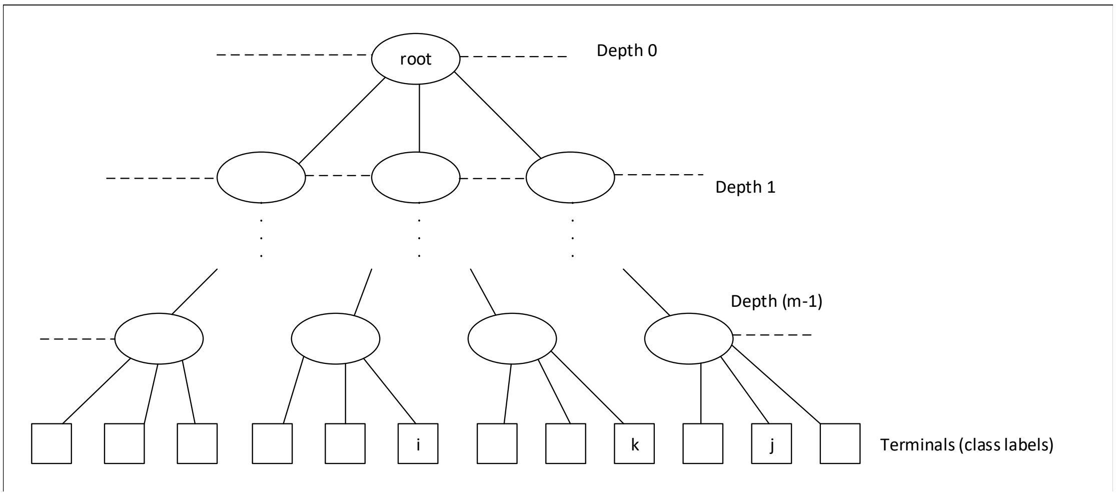


Figure 2.1: General Design of a Decision Tree (Reproduced from paper [37]).

It is well known [38] that, for decades, that timing information has been used for most medical implantations. The Current Implantable Cardioverter Defibrillators (ICDs) applied a time based decision tree for cardiac arrhythmia classification. The network used in this work is a 10:6:3 multilayer perceptron. A perceptron is a computer model that can simulate the brain recognition capabilities. There are a few disadvantages when using a decision tree. It will not be as accurate as the other classifiers as a very small change in the training datasets might result in a huge change in the output prediction. Furthermore, the performance of the DTC is linked to the effectiveness of the particular DTC design.

Next, the review examines the k-NN classifier [35, 39, 40], a non-parametric, non-linear classifier. This type of classifier is more effective when dealing with large datasets. This classifier functions with class assignment based on a nearby dataset. The similarity between the samples used are measured with a distance function. There are two different distance functions that are commonly used, the Euclidean distance function (equation 2.5) and the Manhattan distance function (equation 2.6). The parameters in equation 2.5 and 2.6 are: $d(x_j, x_k)$ is the distance between point $x_j$ and $x_k$; $x_j$ in this context is a point of $x$ on the $j$ axis and $x_k$ is the point $x$ on the $k$ axis. In the context of these two equations, $i$ represents the number of points to be calculated from $i = 0$ to $i = n - 1$ where $n$ is the maximum number of points.

$$d(x_j, x_k) = \sqrt{\sum_{i=0}^{i=n-1} (x_{j,i} - x_{k,i})^2} \tag{2.5}$$

$$d(x_j, x_k) = \sum_{i=0}^{i=n-1} (|x_{j,i} - x_{k,i})| \tag{2.6}$$

Figure 2.2 presents a generic example of the k-NN algorithm where the threshold of the two different classes are shown clearly with different circles. The red star can be interpreted as the input that requires classification, and k are the thresholds (k = $\sqrt{n}$, $n$ is categorized by the data sets, A and B [35]). The two different x axes, x1 and x2 are used as reference for the distance equations 2.5 and 2.6. In this example, the class of red star would be B if k = 3 and A if k = 6. The number of nearest neighbours for class A if k = 3 is higher than that of A, thus the red star is classified as class B. If the circle is expanded to k = 6, it will be classified as class A as the number of nearest neighbours of class A now exceeds that of B.

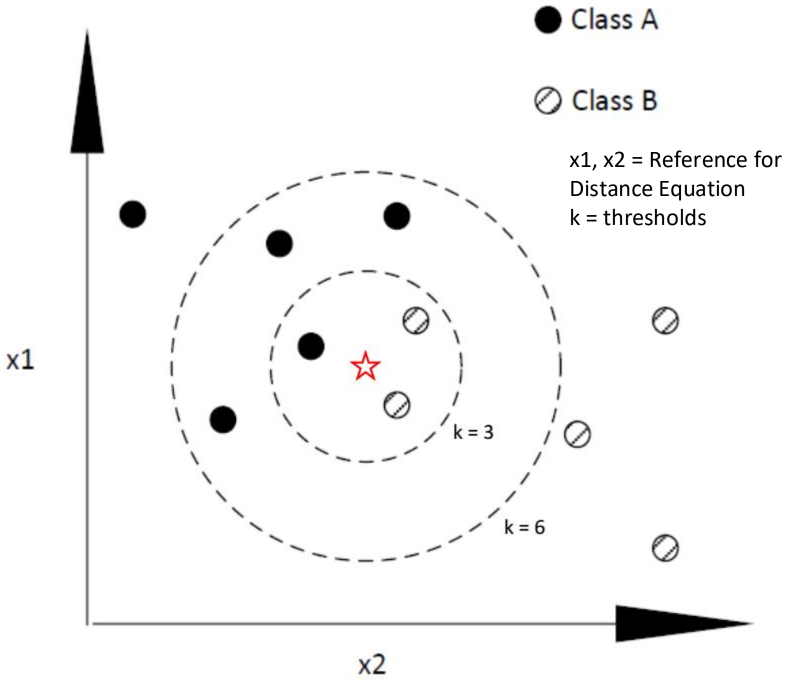


Figure 2.2: General Idea of a k-NN classifier (reproduced from [40]).

In conclusion, we can see that the red star is classified as class B if k = 3.

In general, the classifier takes a majority vote from the k-nearest neighbour where k is the number of neighbour. In A. Sharmilla *et al.* work, the number of k chosen is 2, which provides a minimum error rate [35]. A paper by L. Arbach *et al.* [39] that uses three different classification methodologies to determine which is the best. These methods include a back propagation neural network (BPN), the k-NN classifier and a human reader. It was found that k-NN could not produce a 100% sensitivity as compared to a BPN that prove to have an acceptable performance for the mammographic masses classification application. The recent work by L. Arbach *et al.* in 2003 [39] demonstrated that k-NN is applicable for medical classification problems. Unfortunately, the k-NN classifier still produce a false negative value of 2 while the BPN have no false negative, which inspired the thesis to use a neural network solution rather than the k-NN classifier.

The basic algorithm for a k-NN classifier is relatively similar to that of a neural network classifier. Both have a training stage and a prediction stage. The training stage of the k-NN classifier involves the entirety of the samples used. These samples are stored in a form of memory. A neural network uses the training stage to calculate the weights with the highest accuracy to predict a target output. When taking the similarity between the two different algorithms, it can be beneficial to consider using both algorithms together to compare the output performance.

There are a few advantages to using this particular classifier. The learning process is conducted off-line using simulation software which coincides with our thesis specification (Chapter 1.2). However, the evaluation time for a k-NN classifier is longer compared to an ANN solution. The ANN solution would be more suitable when a hardware implementation is needed as it produces the best results even with huge datasets, i.e. multiple EEG waveforms [34].

## 2.3 Parallel Learning System

There are three type of parallel learning systems: totally dependent; partially independent; totally independent. A totally independent system have nodes that are not affected by any other node in the same level. The partially independent system have nodes which are dependant on the same father nodes in the same level. However, nodes with different father nodes are independent of each other. Lastly, the totally dependant system nodes rely on each other in the same level [41]. The thesis uses a bottom-up learning approach.

Block Based Neural Network (BBNN) is one type of evolutionary algorithm (EA) neural network which consists of two-dimensional arrays that are used to support integer weights [42]. The proposed FPGA based ECG signal classification uses a parallel genetic algorithm and BBNN. The proposed design in this research is suitable for hardware implementation as it has a cellular like structure. This device will be used for long term patient monitoring and the design is shown in Figure 2.3. It also proves to be a form of device that is a great inspiration to the research done in this thesis. However, a different approach is taken here; the implementation uses on-line learning, whereas this thesis research uses off-line learning.

The conventional formulation of fundamental neural algorithms have made neural implementation on state of the art parallel hardware very difficult. Therefore, an effective network parallelisation solution has been formulated by Liberios *et al.* [43]. It is in the form of an algorithmic mapping of multilayer feed forward neural network, and includes

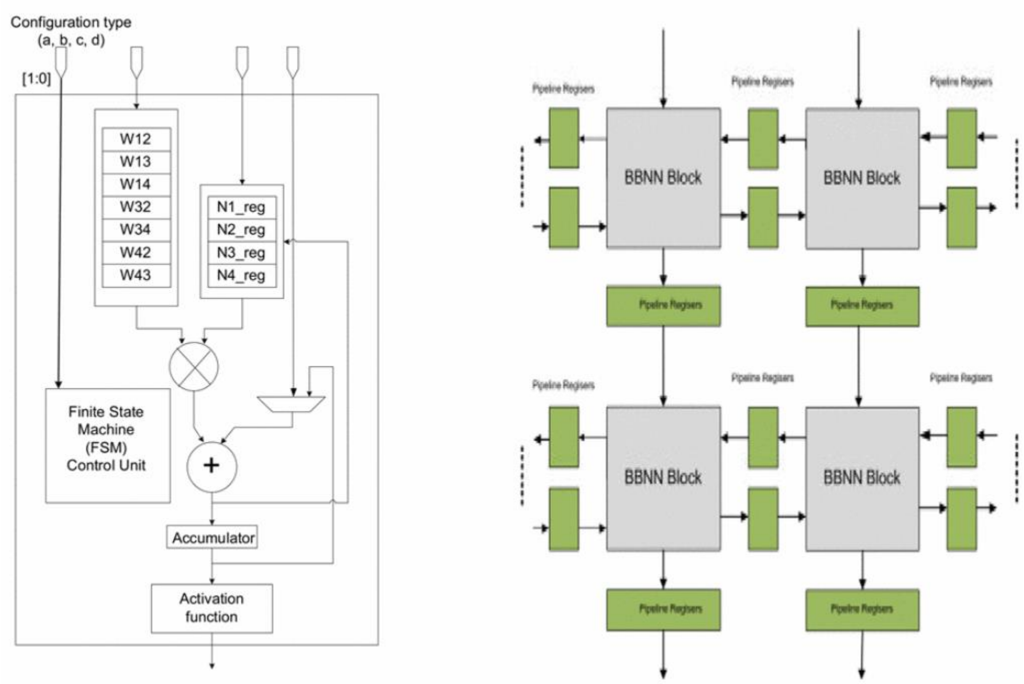


Figure 2.3: The Block Based Neural Network (BBNN) design block diagram (extracted from paper [42]).

a back-propagation learning (FFBP) system. This solution is developed on a massively parallel system framework known as the Neural DF KPI architecture. A recent work [43] used 5 layers for their network topologies and it can be seen their network used over 800 neurons. They have chosen to simulate the network using a Intel P4; 2.4 GHz 1GB RAM workstation.

Grey neural networks [44] are another feasible alternative for predictive applications. There are four types of traditional grey neural network model. These models can be distinguished as serial, parallel, inlaid and blending. The traditional grey model has a rather poor learning ability and a long-term prediction accuracy which is not ideal for the research in this thesis. This literature used a BP neural network to construct a serial grey neural network model, and thus observe an increase in the prediction accuracy [44]. Market prediction would be a suitable application when using this technique. A recent work [45] proposes a improved model to optimize the market prediction model.

The more recent Multi-Views Multi-Learners (MVML) model is a novel neural network designed to solve complex pattern recognition problems [46]. The multi-views single-learner (MVSL) approach may fail to provide a high detection accuracy because MVSL approaches utilizes a single learner to approximate multiple views which might not be enough to converge on all the views. They use two criteria, recognition rate (RR) and normalized root mean square error (NRMSE) to assess the quality of the proposed ANN-based speech recognizer.

## 2.4 Epilepsy Detection Using Software

It is possible that the prediction of the onset of a seizure occurrence can be achieved with the assumption that normal EEG signals are complex but low energy spike waveforms, the spikes will become more repetitive and increase in energy potential during epileptic seizures. Example of EEG waveforms are shown in the figure 2.4, 2.5. By analysing the power/energy spectrum of those waveforms, it is also feasible to continue the analysis by employing a linear approach [8].

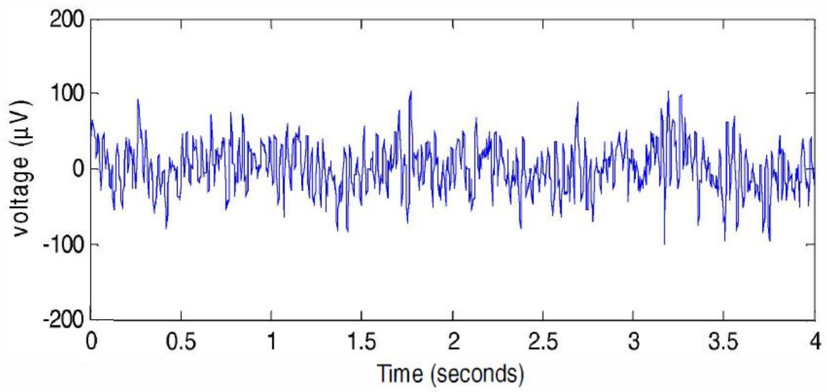


Figure 2.4: Sample of normal EEG signal.

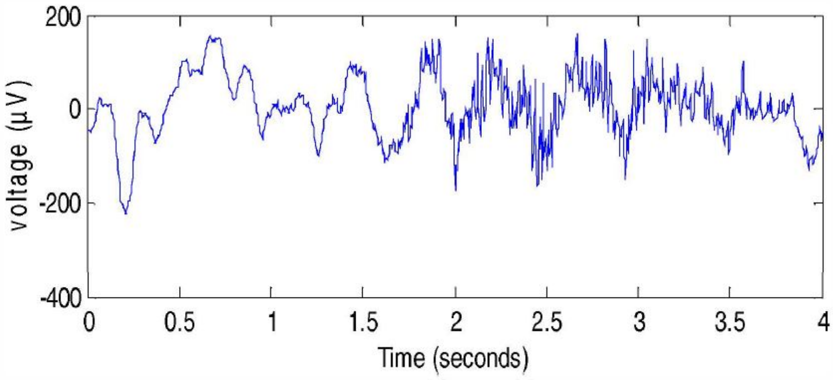


Figure 2.5: Sample of a Seizure EEG signal.

Kiranmayi *et al.* [8] in 2013 states that a three layer BPN is commonly used (Figure 2.6) and in this work, the data used for training and testing purposes is taken from a hospital in India [8]. The paper also suggests the use of a bispectrum for feature extraction in the form of a third order fourier transform. Furthermore, the learning process for this work uses a training set to determine the real values of the EEG signals [47]. The result shows that with the bispectrum feature in neural network analysis have an accuracy of 82.66%, when differentiating an epileptic EEG from a normal EEG. The power spectrum features classification yields an accuracy of 53.33%. The results indicate that the bispectrum feature have a high resistance to noise in EEG recordings. This analysis method can detect phase couplings even with the presence of Gaussian noise in the testing environment. Furthermore, certain processing elements in the hidden layer of BPN are independent of any input. The paper claims the method has an accuracy of 97% when detecting epileptic cases.

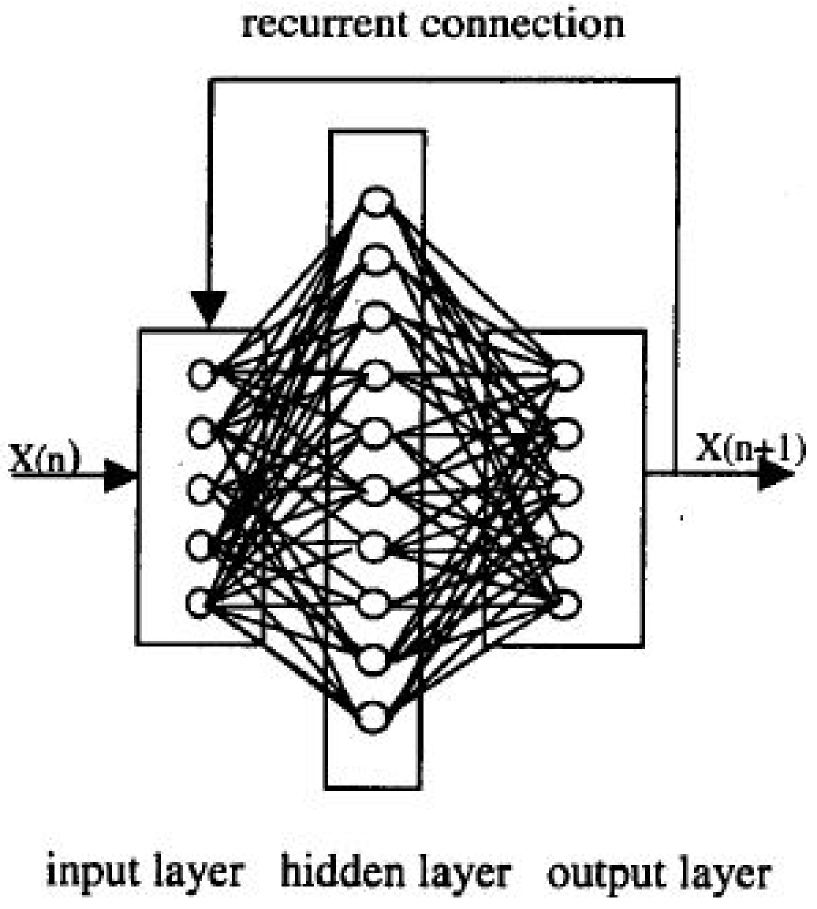


Figure 2.6: Network Structure of recurrent BPN Design (extracted from paper Kiranmayi *et al.*'s work [8])

Another form of automatic detection algorithm, Gabor's BPN method [14], is also used in experimental procedures. In this case, 16 different data channels consisting of Epileptiform Discharges (ED) pattern are used as inputs. The method proposed in this paper has some advantages, as it trains the BPN for each patient to include individual ED pattern. Thus, it has the capability of self-recognition (to recognize ED pattern of each patient). In addition, it has the capability of eliminating artifacts which are activities that are not of cerebral origin, i.e. (eye blinks, electrode and movement artifacts and EMG).

The sliding window technique is also plausible with the use of a feed forward ANN. It is used to differentiate between a normal and an epileptic EEG [48]. The datasets used for the method was obtained from Bonn, Germany [49]. This method has four stages (Figure 2.7) where the neural network classification will be executed once the features are extracted. Soft computing techniques are mainly used in chaotic systems [50] which include ANN amongst other complex algorithms. These algorithms ease the burden on medical experts in the field of medical diagnosis. This paper discuss the use of two different systems, ANN and Neuro-Fuzzy Systems. There are some contributions of using ANN in the medical field as it can model biological systems. This can solve very complex and non-linear problems. The paper proposed a design with a general ANN structure, but it uses a gradient descent approach. This minimizes error at the output stage within the feed-forward network.

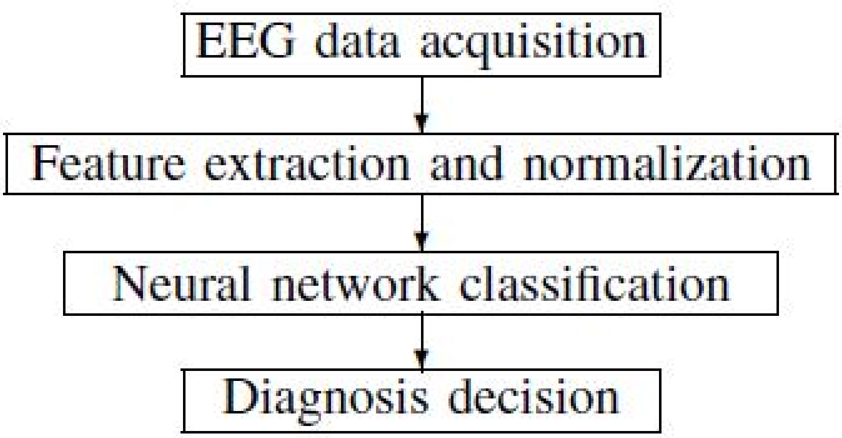

EEG data acquisition
Feature extraction and normalization
Neural network classification
Diagnosis decision

Figure 2.7: Flow diagram of EEG classification scheme incorporating ANN (reproduced from work [51])

The block diagram (Figure 2.8) shows the flow of methodology used here [50]. After the training is complete, it was found that the ANFIS system surpassed the neural network system with an accuracy of 95.38%. However, the use of an ANFIS model will depend on the specific application.

There are other ANN being researched, such as the MLP solution [52] which can be described as a feed forward network with one hidden layer and uses the sigmoid activation function to obtain the output. Furthermore, there is always a target output with a corresponding training set.

Additionally, EEG records have been found to localize within the epileptogenic foci [53]. Researchers have analysed EEG pre-seizure components for seizure activities using a recurrent neural network. The structure is a three layer back propagation 5-10-5 architecture neural network. With incremental training and the gradient-descent algorithm (steepest descent method, which is a form of non-linear optimization method), the BPN can be used to locate the epileptogenic foci within the EEG records of each patient.

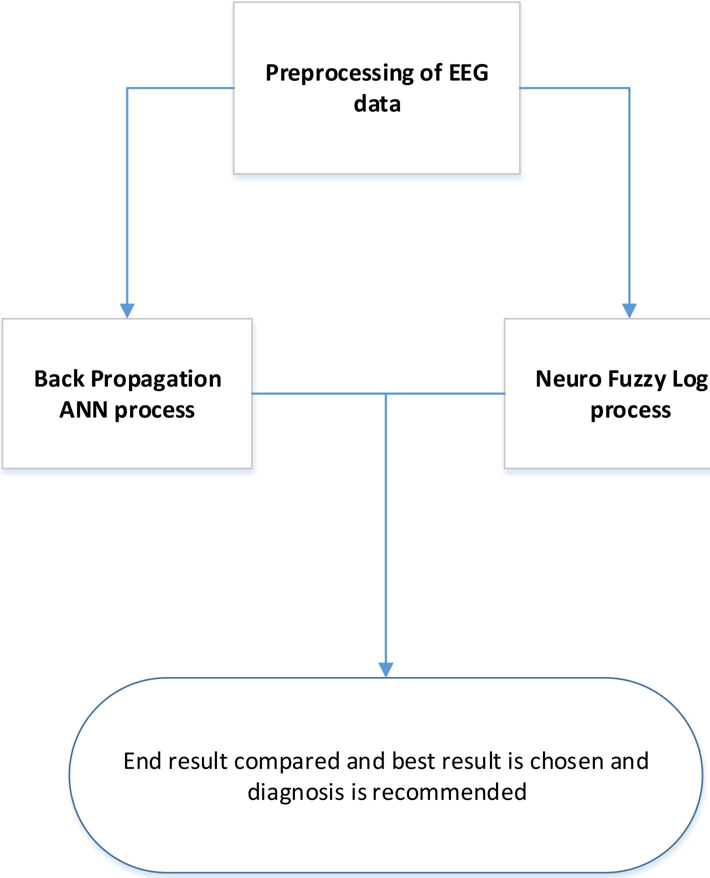

Preprocessing of EEG data
Back Propagation ANN process
Neuro Fuzzy Logic process
End result compared and best result is chosen and diagnosis is recommended

Figure 2.8: Block diagram showing a whole system comparing BPN and a fuzzy logic system [51].

The focus of the work by Bao *et al.* [51] in 2008 is to use pre-seizure EEG and EEG data between seizure/convulsions as an alternative to develop an automatic detection system for epilepsy diagnosis through a probabilistic neural network (PNN). For the purpose of making medical decisions with PNN, the paper suggests the best solution is to use Bayesian strategies [51]. It is very difficult to use EEG data as the input to the ANN as it has to be filtered and analysed. In the report, it was proposed to develop an ANN with distance-based functions for computation and a bell shape activation function when making non-linear decisions. Furthermore, the decision can be changed instantly in real-time as new data is added. The layout of the network design generally consists of 3 layers: an input layer; a radial basis layer; a competitive layer. There is no bias in the competitive layer, but it is present in the radial basis layer. There are two important parameters within the Radial Basis Layer that need to be addressed which are the Radial Basis Layer Weights and Radial Basis Layer Biases.

There are many references in the paper reviewing computerised methods for diagnosing

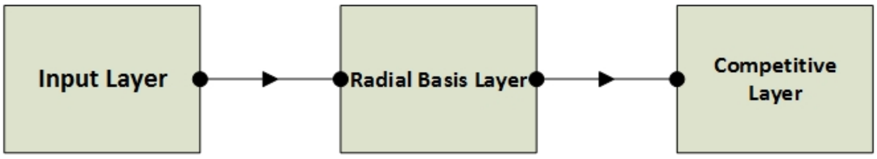


Figure 2.9: Network Structure of PNN Design [51].

epilepsy [54]. The paper focuses on a combination of Elman Network and PNN with inputs from the time domain feature of an EEG signal, approximate entropy (ApEn). ApEn is a statistic parameter that is used to quantify the regularity of a time series in a physiological signal [55]. It has been proposed and used in many other areas [54]. The datasets were obtained in a similar manner to other studies and it contains 100 single-channel EEG segments, each segment has a duration of 23 seconds [54]. Moreover, the datasets are obtained from healthy and epileptic subjects. The epileptic data is recorded during a seizure event with the use of intracranial electrode. The PNN target values, 1 for a normal EEG and 2 for an epileptic EEG [56]. Furthermore, the overall performance ($OA$) of the PNN can be calculated with equation 2.7.

$$OA = N_{CDP}/N_{APP} * 100 \tag{2.7}$$

where $N_{CDP}$ is the total number of correctly detected patterns, and $N_{APP}$ is the total number of applied patterns.

Bao *et al.* [57] extracted three unique features from the EEG signal, Power Spectral Features, an energy distribution description in the frequency domain; Fractal Dimensions which outlines the signal's fractal property; and Hjorth Parameters which models the signal's chaotic behaviour. They fed these features into their PNN. These features were later optimized to fully develop a classification network which is comprised of several PNN-based classifiers. This network has an accuracy of 94.07%. This approach does not require the occurrence of any seizure activity. This reduces the difficulty in data acquisition and thus removes the need for more sophisticated medical devices in places where medical resources are limited. In Bao *et al.* 2008 work [51], they repeated all their experiments using a 10-fold cross-validation and achieved an overall accuracy over 80%, the required specification of this research. Therefore, this shows that their approach has good generalizability. It has a 99.3% overall accuracy during cross validation when employing the interictal EEG based approach (n.b. normal EEG is still included with the interictal data during the learning process) and 96.7% during the ictal EEG based approach. This shows different ANN systems have been developed in the labs and have their own advantages and disadvantages. It should be noted that there is no automated EEG epilepsy diagnostic system using only interictal scalp EEG data at this time. The PNN proves to be fast, highly accurate and the network structure can be updated easily

[58]. A simple voting scheme is used (Figure 2.10) to improve classification accuracy [57] using the method of Leave-One-Out Cross-Validation (LOOCV).

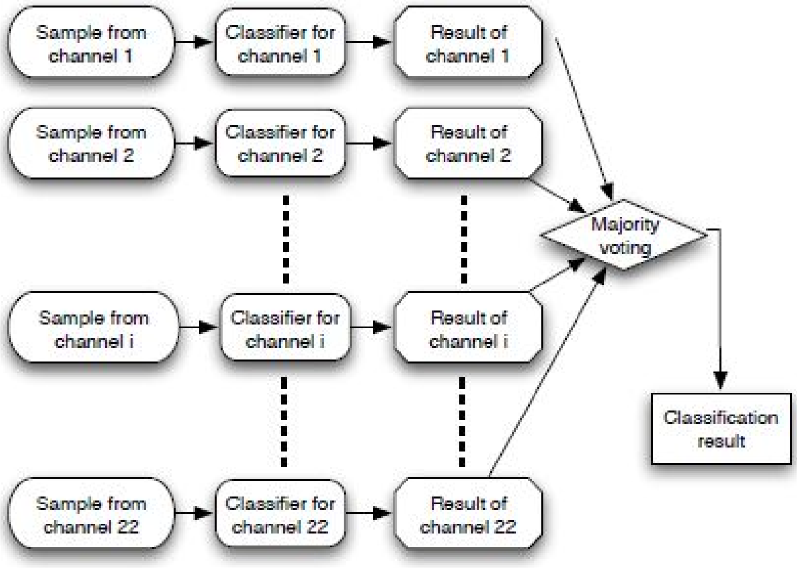


Figure 2.10: Voting scheme used in the classification process [57],

Elman Network (EN) has some contributions that have been reviewed in this thesis. EN has a feedback connection that allows this particular network to recognise and generate temporal and spatial patterns [57]. This network is made from a two layered BPN which includes a feedback loop that connects the output of the hidden layer to the input of the EN [57, 59]. It employs a recency gradient approach. Furthermore, the gradient descent algorithm is used with adaptive learning rate when training the network. Evidence from one of the references, focuses on how Jordan / EN network is used in a recent work [49]. This network allows modification to be made to the multilayer perceptron and it consists of two layers of BPN. The test data was taken from the University of Bonn, and the database is available for public use [60]. The data set consists of 100 single channel EEG segments (23.6 second duration for each segment) with no artificial artifacts. The simulations are conducted using this Jordan/EN with different hidden layers. The input is fed with random data from the database previously obtained. The processing element within the hidden layer during the experiment was also varied. The data used in this work is broken down into three different sections: training data, cross validation (CV) data; testing data. The network is trained repeatedly with random weights to minimize any form of bias. Certain learning rules are also applied in the training procedure. Step momentum is the optimum learning rule in this instance, however other learning rules including conjugate gradient, quick propagation and delta bar delta may also be used. Their proposed clinical epilepsy diagnosis system obtained relatively high overall accuracy of 99.83% for training data and 99.92% for cross validation (CV) data and testing data [49]. Thus, it is a system worth considering for this thesis.

EN recurrent network [52] is another form of EN network that is used in the automatic detection system. It takes results from the previous hidden layer output of the ANN and it feeds it to the input. Furthermore, it uses the sigmoid activation function to introduce the non-linearity property into the EN recurrent network.

SNN is a third generation that has been researched in recent years [61]. SNN is different from other forms of ANN as each individual spiking neuron propagates information by using the timing of the neuron, rather than the rate of the spikes. The supervised rule, known as SpikeProp [62], is also used for training purposes with the assumption that the internal state of the neuron increases linearly within a small enough region for neuronal firing. HuiJuan Fang *et al.* [63] propose a few methods to improve the learning rate adaptability. The proposed methods are then tested with four different experiments : XOR problems; Iris data-set classification problems; fault diagnosis in Tenesse Eastman (TE); decoding information from a Poisson spike train. It was also found that SNN are mainly used in brain modelling [64, 65]. The SNN is very efficient as it only requires a single spiking neuron for pattern recognition. The hardware implementation of a SNN was performed using NVIDIA CUDA, a graphic processing unit (GPU) which can implement the SNN.

There are advantages and disadvantages of this hardware implementation. The constant read-only memory (ROM) is proved to have higher access speed than global memory. However, more GPU memory is required. Additionally, SNN requires more time to access the parameters of individual neurons when ROM are used. The data preprocessing stage for SNN involves the removal of signal noises from the original EEG signal. This process is completed by passing the EEG signal through a low pass filter. Next, the wavelet analysis breaks down the entire EEG input into various sub-band waveforms allowing individual channels to be used as inputs. Furthermore, the chaos analysis stage extracts the important features from the input waveforms. The features are then used as the neural network inputs (Figure 2.11 [21]).

## 2.5 Deep Learning Neural Networks

Si Jin Lie [66] explores the feasibility of using deep learning neural networks (DNN) for human pose detection. In this study, Si Jin Lie stated that a human pose can be estimated quite accurately from a 2D image using depth maps. However, there are some issues that need to be addressed, such as the ambiguities of appearances and self occlusion. In many practices, this type of estimation is completed using two different methods [66]. The methods are part-based graphical models, and both methods rely on regression modelling. In a recent paper [66], certain considerations were made for the proposed neural network such as low level feature sharing, preservation of location information

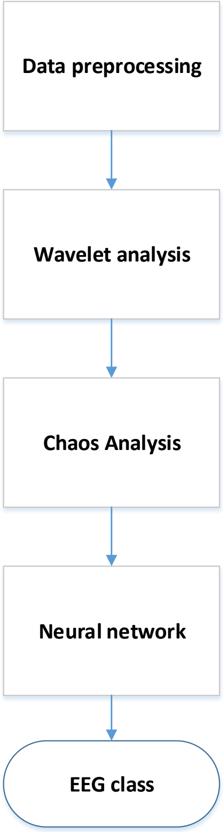


Figure 2.11: The different processes leading to the neural network in classification problem [21],

and integration of context information. The assumptions used by the research group in the proposed deep neural network will be useful in the thesis.

In the case of the speech processing application, the first step is to provide the necessary audio features for processing. Different techniques have been developed over the past decade that are variants of the decision rules based on features of an audio signal [67]. It is stated [67] that convolutional neural networks (CNN) are similar to convolutional deep neural networks (CDNN), with the exception of the additional CNN feature extraction layer.

Other relevant studies include the optimization of DNN. S. Zhang *et al.* [68] in 2014 stated that the need to pre-train the network as an alternative to reduce delay in processing was not required. With dropout based training, the network would only require an approximate of 20 iterations. In order to accomplish this, the network uses dropout as a form of pre-conditioner. Dropout is a useful technology [68] as it increases the generalization capabilities of the neural network. State of the art DNN primarily uses feed-forward neural networks with hidden layers, comprising of the same number of hidden nodes. Figure 2.12 illustrates a sample of DNN.

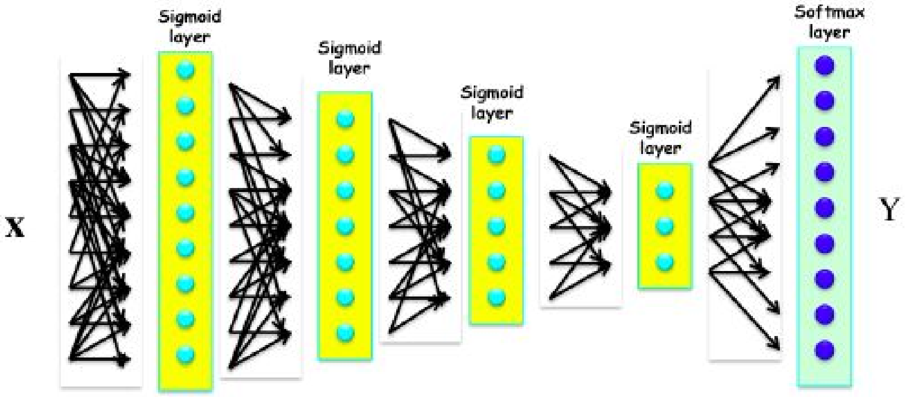


Figure 2.12: A DNN with four hidden layers [68],

## 2.6 Neural Network Processors

### 2.6.1 Radial Basis Function Network

Radial Basis Function (RBF) networks have a fast learning capability and special architecture which is useful for efficient digital hardware implementation. Recent research [69] developed a RBF type network with three inputs to be implemented on the FPGA. This design can be modified easily to include more inputs. Each processing element of the RBF design is analogous to a biological neural element. This may be suitable for the home-based design desired for this thesis. The research design [69] uses the available resources provided by the FPGA development board. Furthermore, it is interesting to compare the performance and logic elements required of each neuron with the proposed bit-serial design.

Equation (2.8) is the end result of the design where : $y$ is the output of the RBF network; $x$ is the input vector; $c_i$ is the centre of the $i$-th basis function; $w_i$ is the weight of the basis function of centre $i$; $N$ -is the number of basis functions [69].

$$y = \sum_{i=1}^{N} g(x, c_i) * w_i \tag{2.8}$$

A full combinatorial approach for hardware implementation is capable of reducing propagation delays. However, there will be an increase in hardware cost. The pipelined sequential approach was proposed by the research team [70] as an alternative, this is a compromise between the rapidity and occupied silicon area. Additionally, the work addresses various problems of hardware implementation in neural networks using a FPGA. The performance issue relating to the activation function has been reviewed. It states that the activation functions commonly used are the sigmoid function in BFNN and the radial function in RBFNN. In order to reduce the design solution complexity, the precision of the activation function can be truncated using a wise linear approximation, or by placing the look-up table within a Read Only Memory (ROM).

Another work by T. Wang *et al.* [71] use a combination of RBF and GA to develop a networked synchronous control model. In this research, the proportional integral derivative (PID) model is compared with the RBF neural network controller. By using the response curve of the phase step, it is proven that the RBF-GA controller is more efficient than the PID controller Figure 2.13.

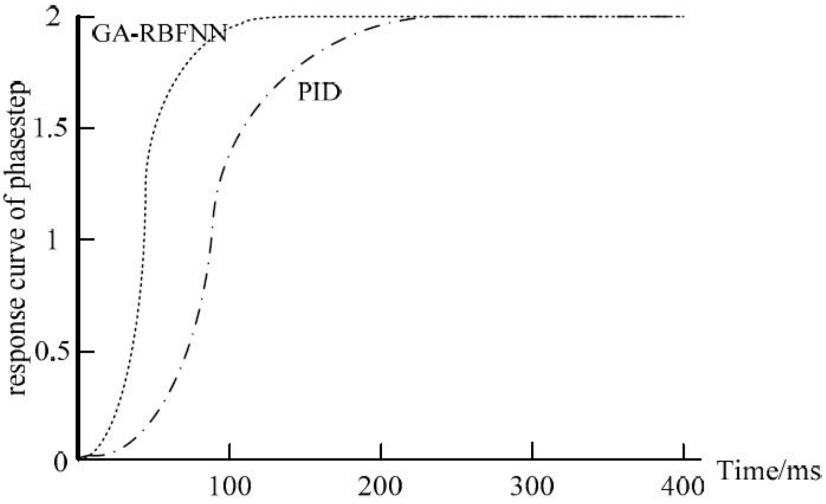


Figure 2.13: The response curve of the phasestep between two different approaches [71],

### 2.6.2 Stochastic Neural Networks

A simplified version of the original ANN hardware architecture was proposed by a different team of researchers [72] to be used in a wind turbine generator system. This ANN hardware system provides control for a sensorless wind speed control system. The stochastic neural network provides a basic, yet effective approach for hardware development. In this section, the basic principles and proposed stochastic feed-forward network will be explained in detail.

The key principles [72] of stochastic arithmetic are:

1. The randomization process converts the inputs into binary stochastic pulse streams. The real number entered into the system will be coded as a binary bit sequence where information is contained using a probability mechanism. The accuracy and the threshold characteristics of the activation function for the stochastic neural

network are paramount. In order to meet these requirements, equation (2.9) is used. This relation applies when $X$ is in the range of [-1,1].

$$X = 2 * p - 1 \tag{2.9}$$

where $X$ is the real number, and $p$ is the probability of any bit with logic '1'. Normalization is enforced to solve the range limitation problem.

2. A bus of random binary bits stream is generated after the randomization process. Any arithmetic operation can be carried out with a simple digital circuit.
3. At the end of the calculation, the random stream will be converted back to a normal numerical value through the de-randomization process.

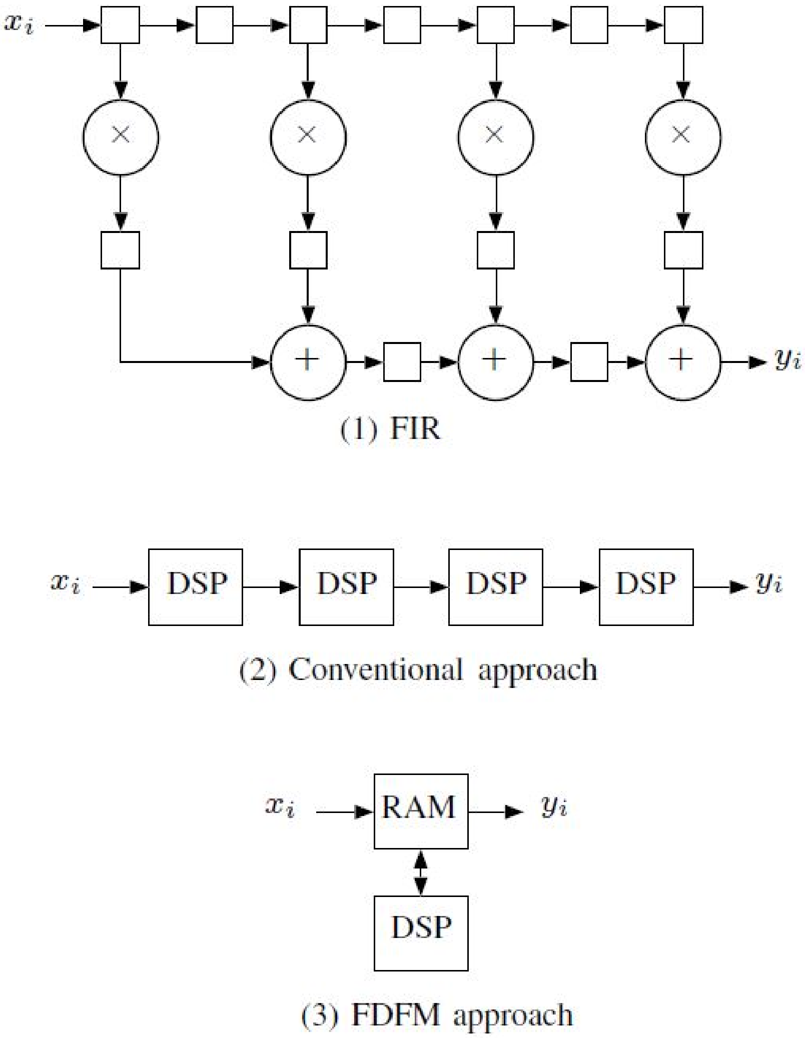
In summary, the mathematical operations used in the proposed feed-forward neural network [72] are : signed multiplication; signed addition; non-linear activation function. Basic mathematical operations do not require a large amount of logic resources. The stochastic multiplier and activation function provide a trade off in terms of accuracy as a method to reduce the number of digital logic elements required for the proposed neural network. The fall in accuracy is associated with the noise introduced by the proposed system.

### 2.6.3 Parallel FDFM Processor Core for Neural Networks

The Few Digital Signal Processor (DSP) slices and Few block RAM (FDFM) design performs routine computation more efficiently [73]. Figure 2.14 includes the conventional approach and a novel FDFM approach proposed by the research group [73]. The conventional approach requires an increase in circuitry.

The paper published by Ago *et al.* [73] reviewed and implemented a 3-layer multilayer perceptron (MLP) using the proposed FDFM approach. Figure 2.16 illustrates the design of this MLP where $N_x$,$N_h$,and $N_o$ clearly denotes the number of nodes in each layer (input, hidden, output layers). The real number $x_i$ which is within the range of [0,1] is fed into the inputs. The target output range ($h'_j$) also lies within the range of [0,1]. The weight of each connection and the sum can be calculated using equation 2.10. In equation 2.10, the $v_{i,j}$ is the weight of each connection. To use this equation, it is assumed that with each hidden node a real number, $c_j$ is assigned. It was decided that a similar neural network approach, a form of MLP would be used as a the basis for the proposed bit-serial neural network (BSNN) in this thesis. Equation 2.10 provides an inspiration for the BSNN in this thesis.

$$h'_j = c_j + \sum_{i=0}^{N_x-1} (v_{i,j} * x_i) \tag{2.10}$$



(1) FIR

(2) Conventional approach

(3) FDFM approach

Figure 2.14: The different Processor Core approach [73]

### 2.6.4 Restricted Boltzmann Machine (RBM)

Restricted Boltzmann Machine (RBM) is a fully connected two layer undirected graphical model with an observed layer and a layer with hidden stochastic variables [74]. With Gibbs distribution, the probability, $p$ of the observed variables in the RBM with a parameter set $\theta$ can be defined using equation 2.6 using a joint energy of visible and hidden units $E(v, h; \theta)$. $v$ and $h$ are the visible and hidden variables respectively. The $Z(\theta)$ denotes the normalisation constant.

$$p(v; \theta) = \frac{1}{Z(\theta)} \sum_h (e^{-E(v,h;\theta)}) \tag{2.11}$$

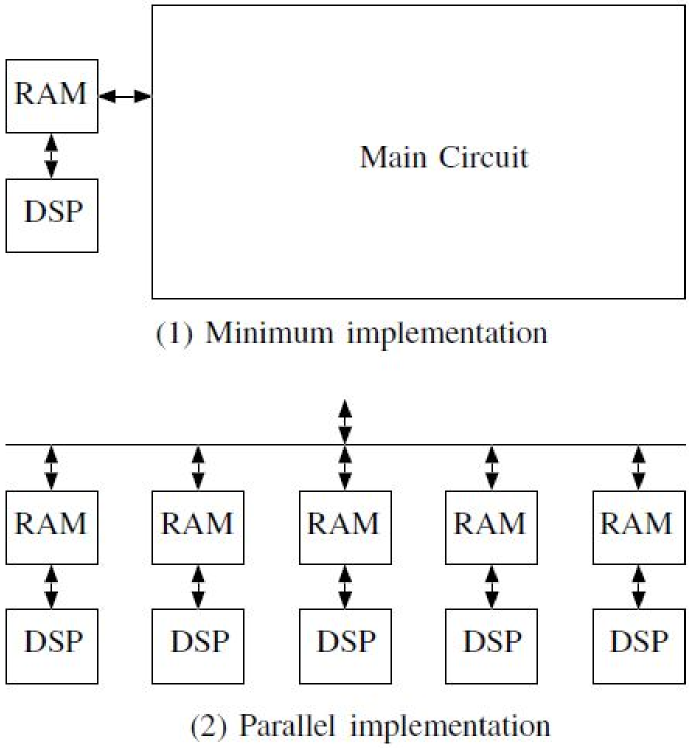


Figure 2.15: The Advantage of FDFM Processor Core approach [73].

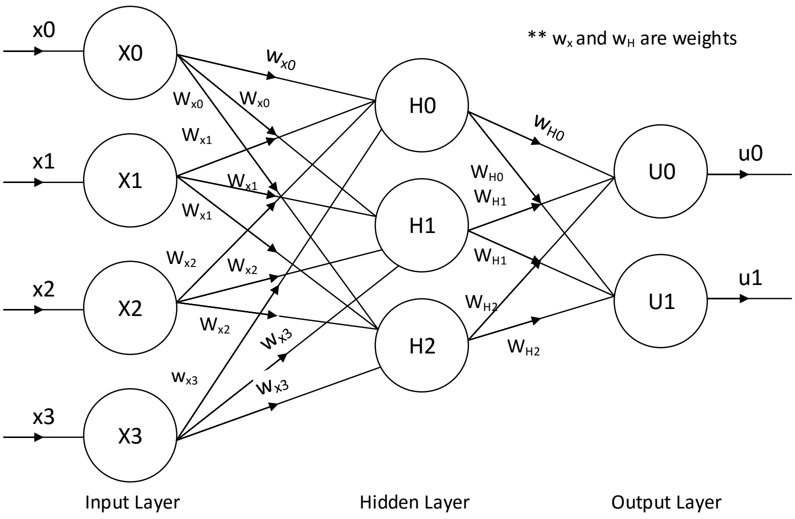


Figure 2.16: 3 Layer MLP Design [73].

The research team [74] proposed and developed a new algorithm using learning class-specific features. This convolutional RBM is a probabilistic model for the density over different observed variables. In contrast with common RBM, this model involves the research of convolutional neural network from a literature written by Lecun Y. et.al [75]. Stacks of these Convolutional RBM (C-RBM) are trained to extract large scale features tuned to any particular object.

### 2.6.5 FPGA-based co-processors

Compact bit vector (CBV) is used to execute core correlation matrix memory (CMM) operations within this type of processor [76]. There are many advantages when using such a representation, including an increase in system storage capacity. However, this representation will compromise the processing performance. The architecture used is the Advanced Uncertain Reasoning Architecture (AURA) [76].

The CMM [76] is a form of binary neural network which is useful in approximate search and match operations that involve massive unstructured datasets. CMM operations are required to be conducted at high speed. In addition, this type of memory is also known as weightless neural network which can be used to implement associative memory structures. CMM can be considered as a two dimensional array $M$, where elements can be set to be 1 or 0. The two main functions of a CMM are loading and recalling an object. When loading an object into the CMM, the input, $I$ and output pattern, $O$ need to be expressed. The $y$ variable is the row index and $x$ the column index. The recall process requires a query input pattern $I$ which can be expressed with equation 2.12.

$$O_x = \sum_{p=1}^{x} (M_{py} \& I_y) \tag{2.12}$$

With hardware hashing, the process of searching and retrieving information from large lookup tables can be completed efficiently. The hardware hashing functionality is based on a combination of a few other functions which include bit folding, exclusive OR and a pseudo random number generator based on cellular automata (CA) [77]. Figure 2.17 shows the design of a hardware hashing memory structure which gives three distinct advantages over RAM based designs [76]:

1. The memory would be limited to the number of column IDs in a single query.
2. All the valid IDs and totals are stored in the current stack frame.
3. By resetting the stack pointer, the memory will be cleared.

In comparison, data streaming memory is heavily pipelined and it requires a burst or stream optimised SDRAM controller to efficiently access the selected row data. The main idea of a hardware hashing memory is illustrated in Figure 2.18 which is one of its functional units. This ITS can be combined to perform logical operations which is necessary for computing problem expressed in the same literature [76].

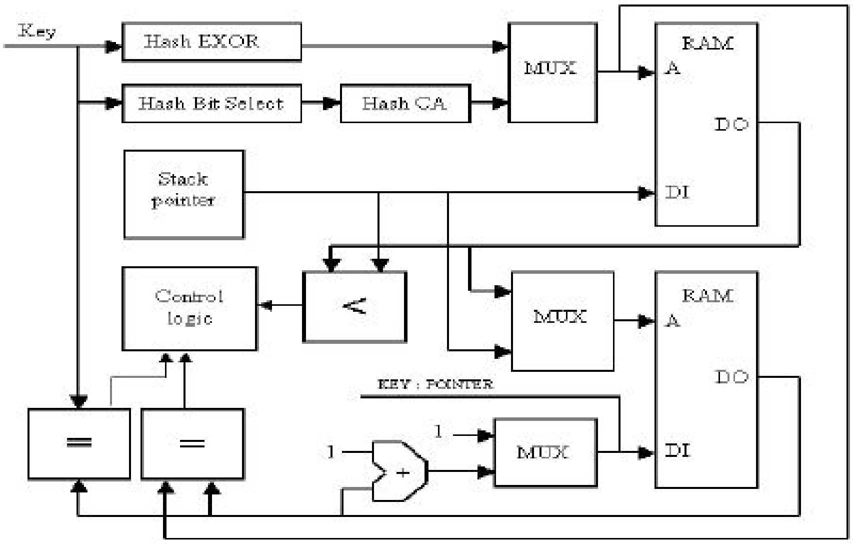


Figure 2.17: The Hardware Hashing Memory [76].

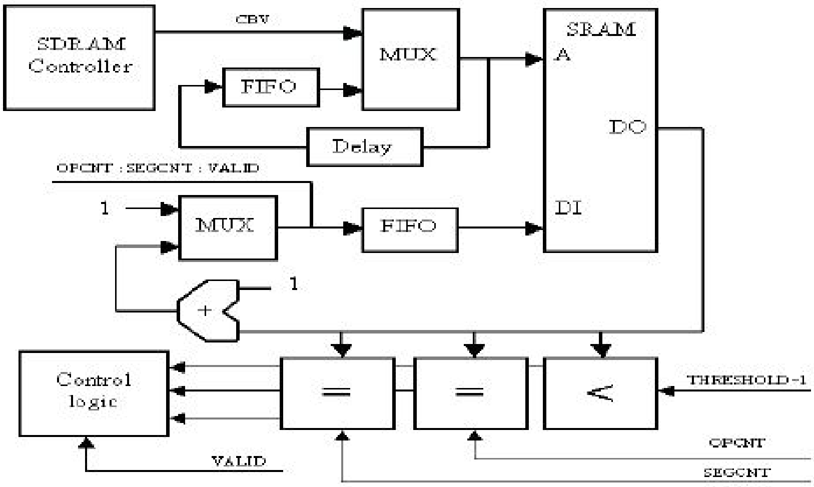


Figure 2.18: The ITS functional unit [76].

### 2.6.6 SNN based Auto-associative based memory

SNN based auto-associative based memory is a type of memory based on the third generation SNN neural network and it is “content-addressable”. Based on a paper by Ang *et al.* [78], the proposed SNN memory is compared with another design proposed by A. van Schaik [79]. The memory developed by A. van Schaik is a Java-based spiking memory model. The proposed model in the literature [79] will store and recall a single item in comparison with other model. The auto-associative memory functionality is clearly illustrated in Figure 2.19. By using some simple form of programmable delays, the training patterns are easily stored and adapted. This idea can be useful when trying to conserve the cost of logic elements.

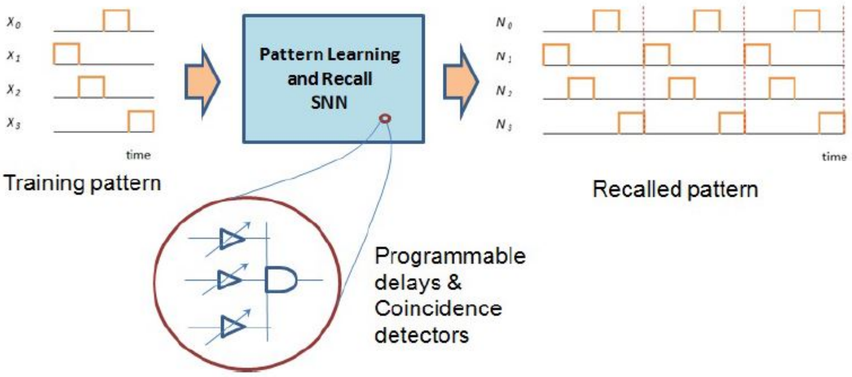


Figure 2.19: The SNN auto-associative memory general functionality [78].

### 2.6.7 Synchronous and Self-timed neuroprocessor

The paper [10] presents a FPGA implementation of a neuroprocessor based on the self-organizing (SOM) architecture. This new novel design is both synchronous and self-timed. Figure 2.20 illustrates the functionality of the SOM architecture and the hardware implementation of this architecture is proposed in the paper.

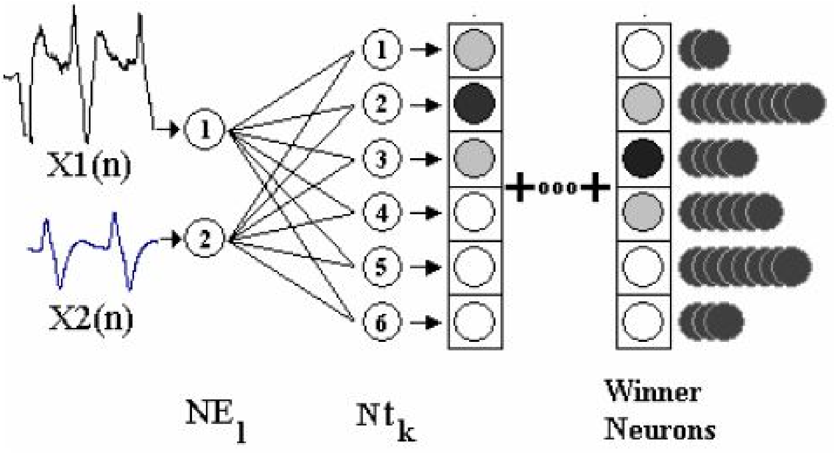


Figure 2.20: The SOM architecture [10].

The SOM network seen in the Figure 2.21 above is composed of 4 main blocks which are listed below [10]:

1. Self-time control block that regulates the data flow in the SOM network.
2. A Somdist Circuit which can be viewed as an ALU of this particular architecture.
3. The Compet circuit can be broken down into two blocks: Compet1 Circuit which compares partial results between active neurons of the output layer; Compet2 Circuit.

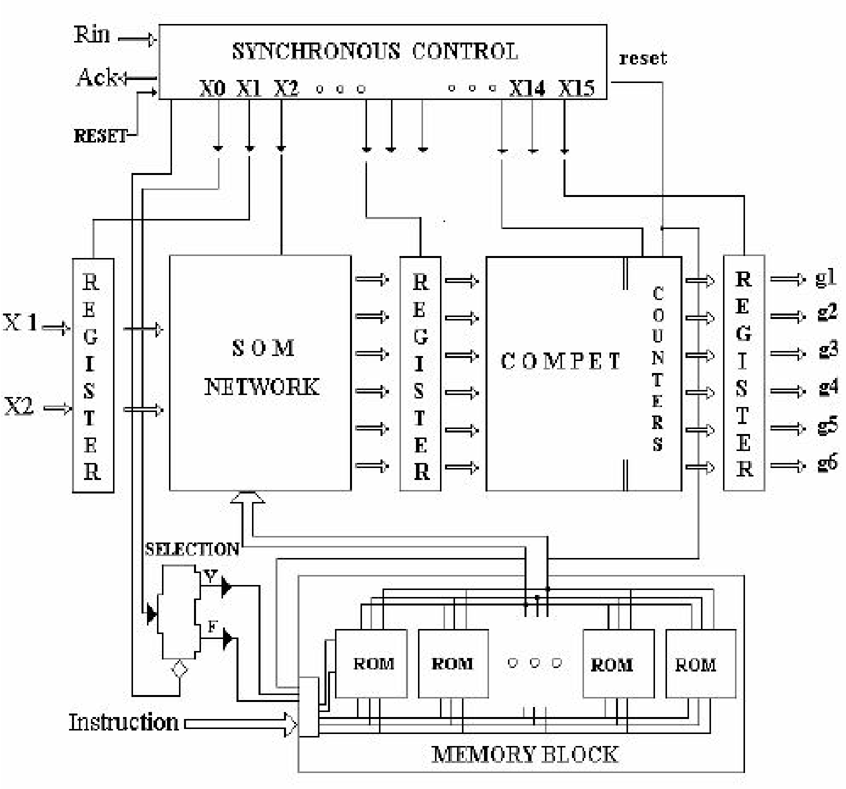


Figure 2.21: The SOM neural network extracted from the source. [10].

4. An array of ROM memory blocks store the weight value corresponding to various input patterns when training the network.

### 2.6.8 Block-based Neural Networks

In 2008, a group [80] came up with a custom FPGA-based implementation that supports dynamic change of the structure and the internal parameters of a certain ANN. This ANN is comprised of 2D regular patterns of locally connected neuron blocks described in the references from a paper [80]. The basic processing elements are connected as a block based neural network (BBNN). BBNN generally are arranged in the form of a grid as illustrated in Figure 2.22.

Equation 2.13 is used for the computation of each individual output based on a weighted input and a bias for the BBNN.

$$y_k = g(b_k + \sum_{j=1}^{J}(w_{j,k} * x_j)), k = 1, 2...K \tag{2.13}$$

where, $y_k$ & $x_j$ are output $k$ and input $j$, $i$ signal of each neuron block. The $w_{j,k}$ & $b_k$ exist as the weight and bias of the output and input node respectively. $g$ is the activation

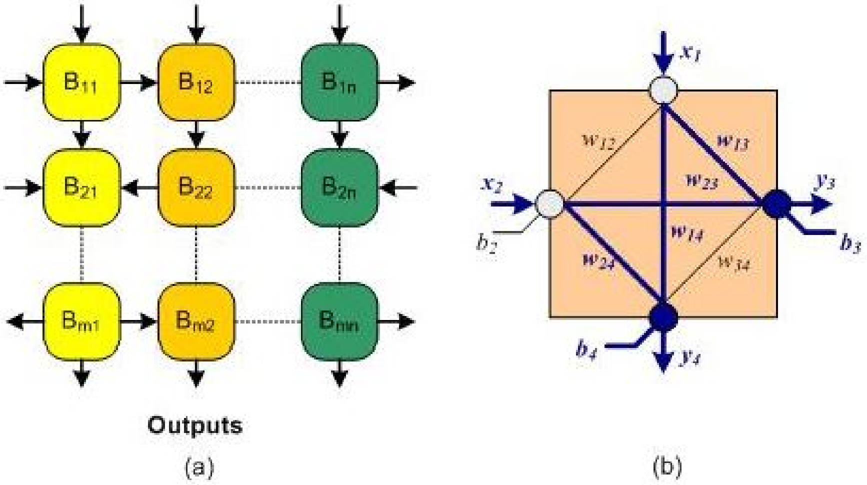


Figure 2.22: (a) The BBNN consisting of basic blocks. (b) A 2/2 internal configuration of the network [80].

function as in many neural network [80]. BBNN was developed over a decade ago but has features suitable for use in this thesis.

## 2.7 Prediction Application Using Different Forms of NN

Many applications involve prediction, one such application involves the use of predictive emission monitoring systems (PEMS). These systems are an alternative method for monitoring air emission condition. A neural network modelling technique was proposed for this application, and certain key issues will impact the performance of the PEMS sensors. These are: reliable and accurate measurement; compliant environmental reporting; integration into existing mill infrastructure; availability of continuing support; cost effective technologies [81].

### 2.7.1 Neural Models Assisted Hardware Implementation Using FPGAs

In 2007, the work conducted by Weinstein, Reid and Lee [82] used different simulation tools such as Matlab / Simulink and LabView. The proposed process utilizes auto generated scripts and runtime interaction tools provided by the software. This can enhance the performance of FPGA as a neural-modelling platform.

Another application involves electromotor control. There are two main types of electromotors: direct current electromotors and alternating current electromotors. Alternating current asynchronous electromotors have more extensive applications as they have a lower manufacturing cost and other favourable characteristics [83]. In order to control such an electromotor, a PID control model is normally used. In this recent research [83], a neuron can be used to simulate the behaviour of the PID model.

### 2.7.2 SpiNNaker: A Massive-Parallel Chip Multiprocessor

The SNN proves to be an inspiration for this research. SNN has many suitable applications in software implementation. SNN has the capability of implementing a massive network which is the basis of SpiNNaker. This architecture is used in various applications such as brain modelling. This form of network requires a huge amount of computational power that fails the specified goals in the thesis research. However, the contributions of SpiNNaker should be addressed.

SpiNNaker is a chip designed with the collaboration of Engineering and Physical Sciences Research Council (EPSRC) and ARM amongst other companies. SpiNNaker is a scalable general purpose platform for massive parallel computing systems which can simulate up to a million neurons with varying degrees of connectivity in real-time. The full detail of the application is explained in many papers [9, 84, 85, 86]. Through the use of the ARM968 processor [87], flexibility and generality is maintained in the neural models simulated by the SpiNNaker chip. The node of the machine consists of 18 such processor cores. Sixteen of these cores will be used for the specified application, one for monitoring and controlling the nodes and the last redundant processor is used in case of any fault. The configuration of the node as well as the best configuration for the whole machine is described in detail in recent works [88, 89]. In another literature [90], a study has been put forward that certain system events have been used by SpiNNaker neural network simulations which include a timer event, packet received event and a Direct Memory Access (DMA) done event. Other than that, the power characteristic of such a system can be characterized as a fixed and a variable power consumption state. The terms that characterize such power consumption of this system are listed below [89]:

1. Reset Power
2. Baseline Power
3. Neural Power
4. Synaptic Power

According to Painkras et.al [89], an assumption was made when the multiprocessor chip was being considered. The main cost behind the computational load is the energy driving the system. A related work [91] proposed that an event driven approach which depends solely on the execution of the data will produce more realistic results in real-life application, compared with the application of a sequential program flow. Another paper

[92] proposed an analog VLSI neural network which is capable of detecting the morphology changes that occur during an ECG. This proposed network has a few advantages such as: a small area as it is an analog design; it can be easily interfaced with an analog signal device; there is no need for temperature compensation.

### 2.7.3 Condition Monitoring Using Different Forms of ANN

The state of the art CMOS technology has led to many new developments in signal electrode technologies such as the high density multiple-electrode arrays (HDMEAs). A large HDME recording system is developed to store the data detected from this array. By recording the electrical activity of a single neuron on multiple electrodes, it is proven to increase the performance of spike sorting. Spike sorting is performed in real time. In this literature, a medium sized Virtex 6 is proven to simultaneously process 650 neurons[93].

When trying to prevent any form of machine fault, condition monitoring is necessary to observe the power electronics components, converters and the condition of the systems in the industrial field. In this research [94], the group attempts to introduce a new method to conduct condition monitoring based on the use of Artificial Neural Network (ANN). It is known that there are two different kinds of electromyogram (EMG) nerve monitoring method: spontaneous and triggered [95]. The custom design presents a flexible framework and good expandability for different applications.

An additional work [96] proposed an on-line streaming kernel that can mitigate the issue of large memory requirement needed by neural signal processing . The Multiple Electrode Array (MEAs) records high resolution neuronal signals which is used in the research. The method proposed is still able to maintain an effective accuracy of the algorithm. The on-line kernel's efficiency is compared to batch processing using a range of BMI benchmark algorithm in the paper.

Back propagation neural network (BPN) is used by some applications. The basis of a BPN includes two distinct stages, a forward propagation stage and a back propagation stage. The normal neural operation uses the forward propagation to pass along the sample provided along the input layer to the hidden layer where calculations are being made which in turn is passed to the output layer to produce the output sample of the neural network. This stage is the basis of our proposed system. The back propagation stage includes a learning process to reduce the error between the calculated output sample and the target output. This process is performed by adjusting the weights of the neural network in real time [8].

BPN neural operation is implemented using equation 2.14.

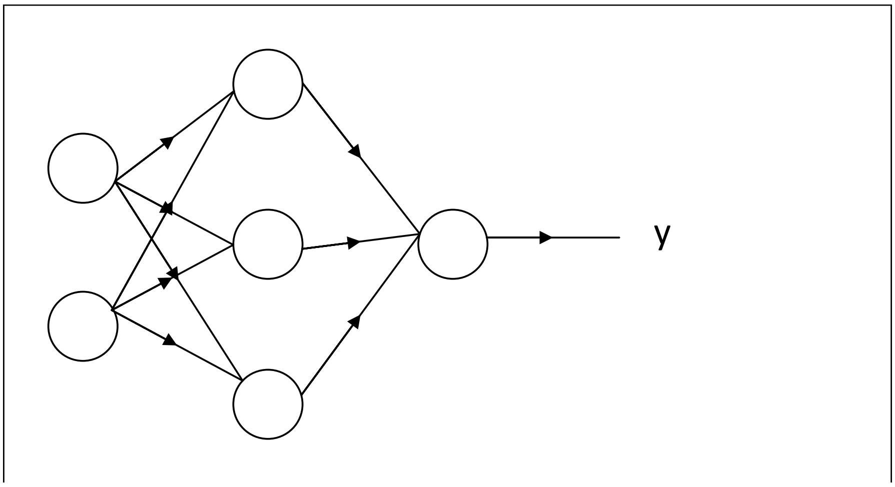


Figure 2.23: General Back Propagation Neural Network (BPN) (Reproduced from paper [8]).

$$u = \sum_{i=1}^{I} w_i x_i \tag{2.14}$$

The equation symbols are identified here. $u$ is the product of the neural calculation, $i$ is the bit number, $w$ is the weight, $x$ is the input and $y$ is the neural output. The back propagation stage uses equation 2.15 to provide an average of the error over a variable number of training samples to increase the performance of the BPN. With the average of sum of squared error, $E$, weight adjustment can be made.

$$E = target - output \tag{2.15}$$

$$E = \frac{1}{2n} \sum_{p} (\| d_p - y_p \|)^2 \tag{2.16}$$

where $E$ is the sum of squared error, $d$ is the expected output, $y$ is the calculated output and p is the sample number and n is the number of samples.

The weight adjustment is calculated using the gradient descent method. Equation (equation 2.17) is provided here when there is need to update the weights. The symbol $\alpha$ is the learning rate, the derivative being the gradient and the product of both is multiplied with -1 as to update the weight, $w_{ij}$ in the direction of the minimum of the error function. The variable $i$ is the inner neuron of the weight and $j$ the output neuron.

$$\Delta w_{ij} = -\alpha \frac{dE}{dw_{ij}} \tag{2.17}$$

In this thesis, the research implements the forward propagation stage in hardware. The training and weight adjustment are completed off-line as to reduce the computational power needed. Furthermore, it is unnecessary to include an on-line learning process for an epilepsy detection application.

## 2.8 Real Time Hardware Based Epilepsy Detection and Prediction Research

This section mainly focus on wearable embedded devices which are used in the area of automated seizure detection.

### 2.8.1 Wearable Embedded Seizure Detection Devices

In 2010, Saleheen *et al.* [97] made some major contributions in FPGA implementation for this specific application. This study found a method to automate real-time detection of seizure events and process EEG signals in an embedded hardware design. The evaluation of the prototype uses three different hardware configurations which are (i) sample entropy and ANN, (ii) variance and predetermined threshold value, and (iii) variance and ANN. The evaluation considered different factors: the accuracy of detection and utilization of hardware resources. The evaluation of algorithmic and optimization techniques reduce the hardware overhead and power consumption while maintaining a high detection accuracy [97]. From the evaluation of the prototype hardware [97], certain observations were made from the results.

1. There is a decrease in the precision of the fixed point representation.
2. They reuse different hardware components during hardware synthesis in order to minimize the hardware footprint. An advantage over other researchers in the field is provided by reducing the hardware footprint by a factor of 4.4. The power consumption was also reduced by a factor of 2.7.

One paper [98] states that a FPGA based sensor has been developed based on Nokia research centre projects. This device uses intelligent processors (Advanced Sensor Processor). Figure 2.24 illustrates the architecture of an Advanced Sensor Processor (ASP) which allows minimal power consumption by processing the signal on a compact and power efficient processor unit. During the signal processing phase, the processor will keep the complex application processing environment idle to conserve computational power. Industrial companies have tried to optimize the processor by reducing the leakage current using a bit-serial processor.

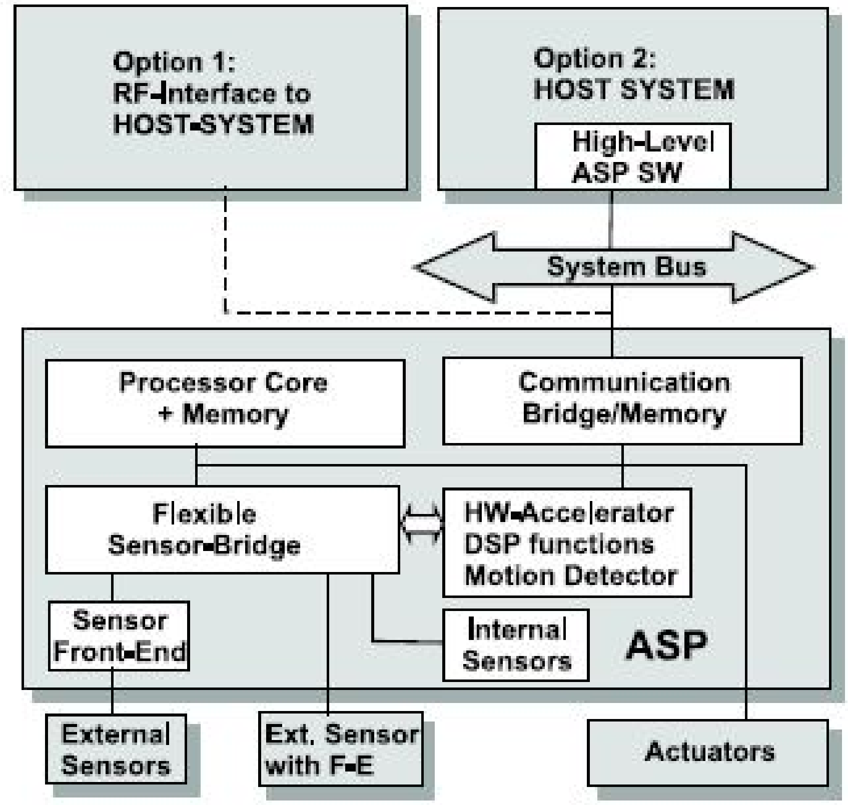


Figure 2.24: The ASP architecture allows two different option of interfacing with the application processor [98].

### 2.8.2 PennBMBI: A General Purpose Wireless BMBI Interface System Design and Developed further for Unrestrained Animals

Neural stimulation is a bidirectional method of communication between the brain and the external hardware. The Brain Machine Brain Interface (BMBI) system proposed integrates four battery-powered wireless devices when implementing the closed-loop sensory motor neural interface, which includes: a neural signal analyzer; a neural stimulator; a body-area sensor node; a graphic user interface. The computer interface is designed by the research team to monitor, control and configure the whole closed-loop system which is connected via a wireless link. This interface uses a custom designed communication protocol [99, 100].

## 2.9 Bit-Serial Architecture with Relation to Neural Network Processors

### 2.9.1 Basics of Bit-Serial Architecture and Advantages over State of the Art Technology

The bit-serial architecture transfers data bit by bit along a wire, during a single clock cycle, while state of the art uses bit-parallel word architecture transfer input bits along a bus during a single clock cycle. This will allow faster processing time. However, it is far better to use bit-serial architecture when the designer requires a lower power design for a low power design as the hardware cost is far lower.

### 2.9.2 COLUMNUS & Bit-serial CORDIC

The design proposed [101] is a SIMD array of bit-sequential processors which provides an extended set of boolean operations. The design is mainly built from a large column of n bit-sequential processors which is then connected directly to a column of dynamic random access memory (DRAM) memory chips using a large data bus (n lines). The SIMD was implemented as a CMOS chip that integrates 32 bit-sequential processors. It was designed using 1.5 $\mu$m technology. The proof of concept was evaluated using a small system consisting of one sequencer board and four processor boards (standard Europe format).

Another architecture was also proposed [102] to address the leakage power issue in modern hardware implementations. Other related work has also supported the claim that the bit-serial approach will be more power conservative . Even though bit-serial design is slower, it can be faster by shortening the critical path. The disadvantage of using this design is the need for a large amount of registers and an increase in dynamic power consumption. This will be dependent on the algorithm required by the designer. The paper [102] focuses on the COordinate Rotation DIgital Computer (CORDIC) Algorithm which uses bit-serial implementation. The design was developed using VHDL and synthesised for a UMC 130nm technology.

### 2.9.3 Bit-Serial Architecture For Neural Network and Various Applications

In order to reduce the size of the logic circuit, bit-serial architecture is implemented while considering the hardware complexity. Further minimisation also occurs during the training phase using the implementation of a leap-frog algorithm. This is completed off-line. Once the network is fully trained with its new weight coefficients, it is loaded onto the existing hardware [103].

When a digital filter is designed using a bit-serial architecture, the input samples are processed in a bit-serial manner as a bit-based filter. However, the overall samples included in the window frame are processed in parallel. A word-based architecture processes the samples sequentially and individual words in a parallel form.

There are a few classifications using this architecture for digital filters reviewed by Yamamoto [104]:

1. Systolic arrays architecture associates each window sample with a rank and this rank is then updated when the window is shifted along the signal.
2. Sorting networks is different from array architecture as there is a sorting mechanism before a sample of necessary rank is selected.
3. Stack-based architectures are a form of filter which maps the filtered sample into the binary domain through the use of majority function.

Moldovan and Fortes [105] illustrates how they can partition and map an algorithm into a fixed size systolic array. Their method is essential when it comes to computational problem that are larger than the size of the VLSI array intended for that problem. Through algorithm partitioning, they divide the index sets into bands and map the bands into the processor space. The results that were obtained from their method was quite promising though quite dated. However, it provides certain inspiration for our research.

Recently, Patrick *et al.* [106] presents Stripes (STR), a new hardware accelerator design for DNN computing. This proposed design is an extension to the DaDianNao (DaDN) design proposed by T. Luo *et al.* [107]. There are many parameters that were compared between STR and DaDN. These parameters are area, energy and performance. The STR exploits precision variability by bit-serial inner-product units while using parallelism.

Another recent work by Charles *et al.* [108] can be used in conjunction with Patrick *et al.* [106] work. Charles research presents a new neural cache that performs bit-serial in cache acceleration of DNN. Their work allowed improved efficiency and lower power consumption across the board.

### 2.9.4 Bit-Serial Multiplier Architecture

In 1982, Noel *et al.* [109] presented a serial multiplier that is suitable for VLSI implementation. This particular design uses a carry-save addition technique that accepts bit-serial inputs and produces bit-serial outputs. Their research use canonic signed digit (CSD) [110] encoding to realize their design along with bit-level pattern coincidences. The resulting design fits well in our attempt to develop a FPGA and ASIC realization for pre-trained neural network. The difference in our approach is our design will be catered mainly for medical applications whereas their approach is more versatile.

The literature [111] states that bit-serial n*n bit multiplication can be split into two different categories: bit-parallel multiplication and bit-serial multiplication. During each clock cycle, the bit-serial multiplier processor will process each input bit individually. Furthermore, bit-serial multipliers can be further classified as serial-serial multipliers or serial-parallel multipliers. Bit-serial multiplier processors are better in terms of power consumption and smaller in size. Another paper [112] supports the decision that array multipliers are not useful in the thesis research as they will consume a large hardware area, thus consuming a large amount of power.

Another work by Shafer *et al.* [113] has also used a fully-serial pipelined multiplier. This research is efficient in certain applications. This type of multiplier includes a quasi-serial multiplier. This multiplier takes in two different inputs operand, one serial and one parallel. However, the output is processed in a serial fashion. This multiplier still requires $2 * n$ clock cycles to perform a multiplication of 2 n-bit numbers. The pipelined design will only require n cycle to return a product, but this would involve a larger hardware cost.

There are two types of multi-bit multiplier: a combinational multiplier and a sequential multiplier which was introduced in a paper [114]. 16-bit representation is used to implement this design which is easy to expand to a 32-bit design. This multiplier is used as an interface for microcomputers to perform specific operations. The current research of this thesis implements a similar form of multiplier that can be compared for performance.

Discrete Wavelet Transform (DWT) have useful features such as multi-resolution time-frequency behaviour, low-aliasing distortion, inherent scalability for VLSI realization and most importantly of all lower computational-complexity. The previous work [115] uses two dimensional DWT (2-D DWT) for image processing. 2-D DWT is highly complex and it involves real-time processing. In order to implement 2-D DWT, there are two approaches: separable (indirect) or non-separable (direct). The separable approach would require two modules of 1-D DWT devices (1 for row transformation and the other for column transformation) and another transposition unit. This transposition unit is not favourable for VLSI implementation [115]. Non-separable devices have small latency in terms of a few clock cycles. This is acceptable in real time applications.

## 2.10 Support Vector Machine Contribution to Epilepsy Detection

EEG signals provide a great deal of detail about the brain activity. Most of the details are redundant, or irrelevant where an epileptic patient is concerned. Thus, this leads to high power overheads for wireless transmission. A smart sensor IC was designed by Sukumaran *et al.* [116] using a CMOS chip that has an area of 0.35um for scalp EEG acquisition. This chip is integrated with the local processing of the sensor node. Furthermore, they include ultra low power electronic circuits to increase the processing power on the chip.

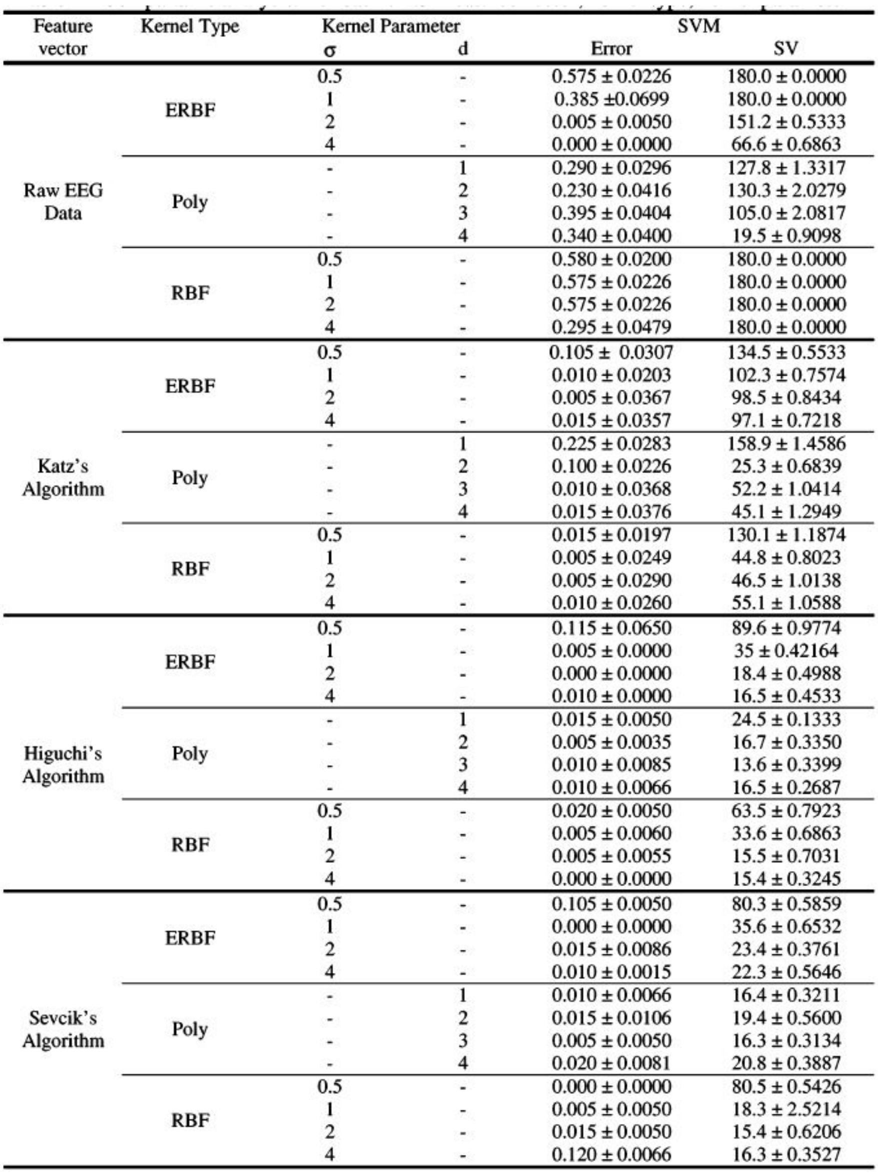
The smart sensor has a low-noise amplifier (LNA) for signal acquisition that is acquired through a single electrode. The feature vectors of the signal are extracted and classified through machine learning. A spiking neuron based ELM pattern classification hardware was used [116] for the classification process. ELM functions similar to a SVM but it requires less nodes and random weights. In order to produce a whole functional system for epilepsy detection, the number of sensors to be worn would increase to achieve spatial correlation. Each individual output of the classifier could then be combined to detect the onset of an epileptic seizure.

### 2.10.1 Support Vector Machine Used in Medical Technologies

In this particular paper [117], the application of fuzzy SVM is being utilized in credit risk analysis. Standard SVM is not suitable here as there is a certain drawback due to their sensitivity to outliers or noises in the training of the sample data. When used for lung cancer diagnosis [118], SVM is suitable as it possesses the advantage of high generalization and an assurance of global optimization as it has been successfully used in many other fields which require classification.

### 2.10.2 Vapnik's Statistical Learning Theory

SVM is based on the Vapnik's Statistical Learning Theory [23]. It is proposed in the research to use the SVM and fractal dimensions (FD) as a mean of EEG signal classification. Figure 2.25 was taken from the paper [23] which provides a comparative analysis table of different algorithms when using support vector machine (SVM).



| Feature vector | Kernel Type | Kernel Parameter | | SVM | |
|---|---|---|---|---|---|
| | | σ | d | Error | SV |
| Raw EEG Data | ERBF | 0.5 | - | 0.575 ± 0.0226 | 180.0 ± 0.0000 |
| | | 1 | - | 0.385 ±0.0699 | 180.0 ± 0.0000 |
| | | 2 | - | 0.005 ± 0.0050 | 151.2 ± 0.5333 |
| | | 4 | - | 0.000 ± 0.0000 | 66.6 ± 0.6863 |
| | Poly | - | 1 | 0.290 ± 0.0296 | 127.8 ± 1.3317 |
| | | - | 2 | 0.230 ± 0.0416 | 130.3 ± 2.0279 |
| | | - | 3 | 0.395 ± 0.0404 | 105.0 ± 2.0817 |
| | | - | 4 | 0.340 ± 0.0400 | 19.5 ± 0.9098 |
| | RBF | 0.5 | - | 0.580 ± 0.0200 | 180.0 ± 0.0000 |
| | | 1 | - | 0.575 ± 0.0226 | 180.0 ± 0.0000 |
| | | 2 | - | 0.575 ± 0.0226 | 180.0 ± 0.0000 |
| | | 4 | - | 0.295 ± 0.0479 | 180.0 ± 0.0000 |
| Katz's Algorithm | ERBF | 0.5 | - | 0.105 ± 0.0307 | 134.5 ± 0.5533 |
| | | 1 | - | 0.010 ± 0.0203 | 102.3 ± 0.7574 |
| | | 2 | - | 0.005 ± 0.0367 | 98.5 ± 0.8434 |
| | | 4 | - | 0.015 ± 0.0357 | 97.1 ± 0.7218 |
| | Poly | - | 1 | 0.225 ± 0.0283 | 158.9 ± 1.4586 |
| | | - | 2 | 0.100 ± 0.0226 | 25.3 ± 0.6839 |
| | | - | 3 | 0.010 ± 0.0368 | 52.2 ± 1.0414 |
| | | - | 4 | 0.015 ± 0.0376 | 45.1 ± 1.2949 |
| | RBF | 0.5 | - | 0.015 ± 0.0197 | 130.1 ± 1.1874 |
| | | 1 | - | 0.005 ± 0.0249 | 44.8 ± 0.8023 |
| | | 2 | - | 0.005 ± 0.0290 | 46.5 ± 1.0138 |
| | | 4 | - | 0.010 ± 0.0260 | 55.1 ± 1.0588 |
| Higuchi's Algorithm | ERBF | 0.5 | - | 0.115 ± 0.0650 | 89.6 ± 0.9774 |
| | | 1 | - | 0.005 ± 0.0000 | 35 ± 0.42164 |
| | | 2 | - | 0.000 ± 0.0000 | 18.4 ± 0.4988 |
| | | 4 | - | 0.010 ± 0.0000 | 16.5 ± 0.4533 |
| | Poly | - | 1 | 0.015 ± 0.0050 | 24.5 ± 0.1333 |
| | | - | 2 | 0.005 ± 0.0035 | 16.7 ± 0.3350 |
| | | - | 3 | 0.010 ± 0.0085 | 13.6 ± 0.3399 |
| | | - | 4 | 0.010 ± 0.0066 | 16.5 ± 0.2687 |
| | RBF | 0.5 | - | 0.020 ± 0.0050 | 63.5 ± 0.7923 |
| | | 1 | - | 0.005 ± 0.0060 | 33.6 ± 0.6863 |
| | | 2 | - | 0.005 ± 0.0055 | 15.5 ± 0.7031 |
| | | 4 | - | 0.000 ± 0.0000 | 15.4 ± 0.3245 |
| Sevcik's Algorithm | ERBF | 0.5 | - | 0.105 ± 0.0050 | 80.3 ± 0.5859 |
| | | 1 | - | 0.000 ± 0.0000 | 35.6 ± 0.6532 |
| | | 2 | - | 0.015 ± 0.0086 | 23.4 ± 0.3761 |
| | | 4 | - | 0.010 ± 0.0015 | 22.3 ± 0.5646 |
| | Poly | - | 1 | 0.010 ± 0.0066 | 16.4 ± 0.3211 |
| | | - | 2 | 0.015 ± 0.0106 | 19.4 ± 0.5600 |
| | | - | 3 | 0.005 ± 0.0050 | 16.3 ± 0.3134 |
| | | - | 4 | 0.020 ± 0.0081 | 20.8 ± 0.3887 |
| | RBF | 0.5 | - | 0.000 ± 0.0000 | 80.5 ± 0.5426 |
| | | 1 | - | 0.005 ± 0.0050 | 18.3 ± 2.5214 |
| | | 2 | - | 0.015 ± 0.0050 | 15.4 ± 0.6206 |
| | | 4 | - | 0.120 ± 0.0066 | 16.3 ± 0.3527 |

Figure 2.25: A Comparative Analysis Table Extracted From The Literature [23] Detailing Different Algorithms Using Support Vector Machine (SVM).

## 2.11 Other Related Work

### 2.11.1 Energy Efficient VLSI Neural Network Design

ASLAN (Automatic methodology for Sequential Logic ApproximatioN) is proposed in a related work [119] to create an approximate version of the sequential circuit which consumes less energy while meeting the necessary requirements of the circuit .

This proposed methodology was used to synthesise a few approximations of well known sequential benchmarks such as 16-bit FIR filter and a 8-bit neuron. These synthesised approximations are then evaluated using two different quality metrics which are maximum error magnitude and relative error.

### 2.11.2 Dedicated Neural Hardware for Medical Technologies

The group [120] studied the Health-Related Quality of Life (HRQOL) and came up with a classification method that can be broken down into three different components including physical health, mental health and social health. One main factor that perceives the HRQOL in Temporal Lobe Epilepsy(TLE) patients would be depression.

The literature [121] outlined the report, through the development of the Quality of Life of Epilepsy questionnaire(QOLIE-89). There is an association between depression and HRQOL. A mood factor has been developed by using the Profile of Mood State (POMS) which was the best predictor of HRQOL. This explains 47% of the variance in the predicted value.

With the wireless sensor networks (WSNs) [122], the system is capable of monitoring patient vital signs in hospitals, and enhancing the performance of emergency responders in large disasters by using automatic electronic triage. Further, WSN also improves the quality of life of the elderly in many situations and enables large field studies of human behaviour and chronic diseases.

There are various technical challenges that will need to be solved when trying to apply WSNs in the patients' daily activities. These challenges can be categorized as follows: trustworthiness; privacy and security; resource scarcity.

BMI approaches [123] have recently been extended to be a recurrent brain-machine interface. This new interface will introduce an artificial neural pathway that allows the adaptive brain to learn to adapt and incorporate the normal desired function . The design proposed involves the use of various forms of antenna to find the optimal design such as monopole antenna; micro-strip patch antenna; substrate integrated waveguide antennas. Next, it employs the use of multi channel spike sorting. The work uses FPGA to evaluate their design.

## 2.12 Critical Analysis

This background research aims to present an overall understanding of the ongoing research into epilepsy detection. Using this knowledge, this thesis presents an alternative epilepsy detection system design which can be used in a wearable device.

The EEG research in this chapter presents an overview on EEG signals and the traditional method for EEG waveform analysis which involves expensive manpower, a specialist and manual inspection of the EEG waveform recording. This method is very time consuming and prone to human error. Therefore, there is a need for automated EEG waveform analysis. Several methodologies have been used and improved to develop such an automated system. These methods include: STFT, Wavelet Transform, Lyapunov

Exponent, AR modelling etc. The advantages and disadvantages of these methods is summarized in Table 2.3.

| Techniques | Advantages | Disadvantages |
|---|---|---|
| STFT | Better noise sensitivity compared to normal FFT<br>Short process time | Less accurate compared to WT |
| Wavelet Transform (WT) | works on a multi-scale basis,<br>More accurate compared to STFT<br>Low spectral leakage similar to AR | needs selecting a proper mother wavelet |
| Lyapunov Exponent | Robust against noise<br>Can evaluate chaotic behaviour of the EEG signals | Complexity in selecting parameters for analysis |
| AR modelling | High frequency resolution similar to WT | susceptible to heavy biases and large variability |

Table 2.3: Advantages and Disadvantages of the reviewed EEG research technologies [124, 125]

This research does not focus on the issue of EEG signal processing. These methods were reviewed to provide the reader with an understanding of EEG analysis methodology in epilepsy detection.

Conventional software classification methods for epilepsy detection does include NB classifier, DTC, k-NN etc. Table summarises the advantages and disadvantages of these approach.

| Methods | Advantages | Disadvantages |
|---|---|---|
| NB | Limited use of training data | Based on assumption that features used are all independent<br>information might be lost when making continuous features discrete |
| DTC | Do not require manual feature selection | small change in training data lead to possible error in prediction<br>performance is linked to the particular DTC design |
| k-NN | effective when dealing with large dataset<br>learning conducted offline | long evaluation time |
| ANN | learning conducted off line<br>able to deal with large EEG datasets<br>feature selection is optional though recommended<br>amount of training data can be limited if accuracy has been obtained | accuracy will depend on particular design |

Table 2.4: Advantages and Disadvantages of different conventional epilepsy detection classification methods

Epilepsy Detection using ANN software implementation include BPN; PNN; EN and a more recent SNN. With a software implementation, high accuracy up to 97% can be obtained. The different types of ANN structure are compared here to provide some idea for the hardware design presented in this thesis.

In summary, every form of ANN in Table 2.5 involves some form of feed-forward neural network (FNN). Therefore, it can be summarised that designing a better FNN is useful for a wearable epilepsy detection system. This research propose to design a hardware FNN which can be used as the main processor for such a system.

Table 2.6 gives a summary of different processors that are included in this literature review. This table shows that different approaches have been used to improve the results

| Type of ANN | Advantages | Disadvantages |
|---|---|---|
| BPN | fast, flexible, easy to program,<br>does not require extensive parameter tuning | performance depend on input data,<br>sensitive to noisy data and outliers |
| PNN | faster than MLP, more accurate than MLP,<br>easily updated network structure | slower than MLP, require large memory space |
| EN | an extension of the BPN design, can be used recurrently | requires specific learning rules |
| SNN | closely models a biological brain<br>efficient in pattern recognition | require more GPU memory and time to access parameters |

Table 2.5: Advantages and Disadvantages of different bit-serial architecture ANN

| Neural Processors | Advantages | Disadvantages |
|---|---|---|
| FPGA based processors | Reprogrammable<br>Suitable for proof of concept and prototyping | Higher unit cost compared to micro-processor |
| SNN based Auto-associative based memory | simple circuity | not reprogrammable |
| Synchronous and self time neuroprocessor | 80% accurate<br>for pattern recognition<br>easy programming | Large hardware cost |
| Block-based | optimized configuration | restricted to 2D arrays<br>and integer weights (easy implementation) |

Table 2.6: Advantages and Disadvantages of different type of processors

for pattern recognition. This research use the information gathered from these papers to further improve the pattern recognition of EEG signals in the context of an epilepsy detection system.

### 2.12.1 Why not deep learning neural networks in this research?

The deep learning neural network is a network that has the capability to compute and update the required weights in real time. This requirement is not a necessary specification for a long term epilepsy detection and monitoring device. It is possible for the device to be trained and updated off-line as a way of reducing computational power required by a deep neural network.

### 2.12.2 Why FPGA for prototyping?

FPGA is a probable choice for prototyping as it is more flexible, reusable, quicker to acquire not pre-designed to perform a certain task. A ready made FPGA is purchasable which cost more upfront. However, re-programmability prevents recurring expenses but it is better at handling parallelized task. The simpler design cycle with integrated software which makes it easier to manage much of the routing, placement, and timing to match the programmed specification.

In order to meet the re-programmability specification of this research, there are a few technologies that has been considered, such as the FPGA and available microcontrollers. There is a reason why the FPGA was chosen over the microcontroller. If we have chosen a microcontroller, we need to have a sequencer and a program to implement our algorithm. If we use an FPGA, there is another option of using only FSM to implement our algorithm. Therefore, the design decision was made to use an FPGA instead of a microcontroller. Furthermore, a microcontroller is not suitable for an application that involves large number of parallel operations. In the case of our research, we need to operate multiple DPUs in the same neural network layer in parallel. Thus, an FPGA is a far better choice than a microcontroller. As this is the first stage of the research, an FPGA provides more flexibility when compared with a microcontroller. At a later stage in the research, the prototype can be implemented on a microcontroller.

### 2.12.3 Why not SVM?

By analysing the work of Yongqiao Wang *et al.* [117], we can see that SVM have made some contribution towards epilepsy detection research. However, there is the issue of sensitivity to noise or outliers in the sample data when going through the training phase as mentioned in section 2.10.1. In the case of an ANN, this issue can be dealt with feature extraction and EEG signal processing.

### 2.12.4 Why bit-serial architecture?

| **Bit-serial technology** | **Advantages** | **Disadvantages** |
|---|---|---|
| COLUMNUS | More powerful bit-serial operation in a single clock cycle<br>Application not limited by memory size | Dedicated array processor |
| Bit-serial CORDIC | reduced leakage power | slower logic |
| STR | Highly configurable | Traded area for precision |

Table 2.7: Advantages and Disadvantages of different bit-serial architecture ANN

The dedicated array processor COLUMNUS inspires the thesis research using dynamic memory bank and bit-serial architecture. However, it is designed with a specific purpose for pattern recognition in statistical physics. Therefore, COLUMNUS cannot be directly implemented in the thesis research. The COLUMNUS design does indicate that a dedicated processor is better than a general processor when solving a specific problem such as epilepsy detection.

The CORDIC algorithm is a low level design with the intention for reducing leakage power and different constraints have been applied in this design to achieve the smallest and fastest solution. This design provides inspiration for the thesis to use a bit-serial approach as to find the smallest solution for the epilepsy detection system.

With bit-serial data path, it is possible that throughput will decrease however this will reduce the area as well. This is presented in the STR design which is similar to our proposed design.

## 2.13 Conclusion

In summary, this literature review has covered a range of areas in the ongoing epilepsy detection research. There are four different EEG waveform analysis that can be considered when analysing EEG waveforms. These are STFT; Lyapunov Exponent; Wavelet Transform; autoregressive modelling. The literature review has focused on software implementation of neural networks (i.e. BPN, PNN, EN and SNN), thus a major section of the literature review has been used to focus on different neural network processors. Many forms of ANN are suitable for software implementation when a simulation is needed to resolve certain problems. In this research, a hardware solution is required that needs to perform at a fast pace yet not reducing the accuracy of the solution. The first stage of the BPN has been used in this research which essentially forms an Multi Layered Perceptron (MLP) and the SNN has been used as a inspiration for the research.

State of the art neural processors such as SpiNNaker requires high computational power which is not conducive for a mobile hardware solution. Such parallel systems is efficient in terms of speed but have larger hardware cost. However, the research needs a low cost approach with acceptable efficiency. Therefore, bit-serial architecture is used in the new approach. The proposed network focuses low cost and power. The speed is considered to be a second priority in the specification of this research.

The proposed system would be an open-loop detection system. This is a form of device that has no form of intelligent control mechanism. It is used to monitor the brain state and improve seizure control. In comparison, close loop seizure control is a three part control system: an EEG recording system, seizure detection signal processor and a programmable neurostimulator. This system however analyses EEG recording in real-time [126]. Such a close loop system will be far more accurate yet larger in size and cost. The new approach uses a variable bit-precision approach to reduce the size and cost while still maintaining an acceptable accuracy.

In conclusion, a neural network solution proves to be a better solution as compared with other forms of classifier that can meet the research specifications. The neural network can be implemented in hardware without affecting much of the device performance when processing huge amounts of data which is one of the main goals of this thesis. The next chapter will present the work of the author in this research that incorporates the choice of using a neural network. Chapter 3 presents the research team proposal to develop a bit-serial neural network for epilepsy detection. In this chapter, the author addresses the advantages of using the ANN in the proposed hardware through various experiments.

With positive experimental results, the thesis hopes to provide a convincing proof of concept for the use of the newly proposed hardware. The main reason to have a working proof of concept for a wearable / home based bit-serial neural network for epilepsy detection is because there are still many epileptic patients across the globe who are always at risk of accidents. Furthermore, this proof of concept would be able to act as the first step in assisting the patients in completing their daily routine safely.

# Chapter 3

# Bit-serial Dedicated Neural Data Processing Unit (DPU)

In epilepsy detection research, the dedicated neural hardware approach is one of the main reasons this research was conducted. The bit-serial architecture was chosen to model our design as bit-serial computing has been the interest of ultra low energy consumption processor designers in recent years. Bit-serial architectures which process data bit by bit during each clock cycle are largely historic. Most modern processors use bit-parallel data processing for performance. However, when high performance is not a priority but instead the emphasis is on very low-power and low-cost, bit-serial computing has its advantages. In modern applications, bit-serial processing is still used in digital filters where input samples are processed in a bit-serial manner. [102].

This chapter is organized as follows: Section 3.1 presents the fundamental equations of the proposed data processing unit (DPU); Section 3.2 then introduces the proposed DPU design; Section 3.3 illustrates the basic experiments conducted to verify the DPU's functionality and the results of synthesising the proposed DPU on various FPGAs; Section 3.4 discusses the proposed model and compares it with related work; Section 3.5 finalizes the chapter.

## 3.1 Neural Processor Model

The neural model of the proposed dedicated processor is implemented using the linear equation 3.1. The equation is based on the McCulloch Pitts neuron model. This proposed Data Processing Unit (DPU) is the first contribution of this research and is used to implement equation 3.1. In this equation, $u$ is the sum of products calculated from the inputs, $x_i$ and weights, $w_i$; where $i$ is the number of inputs.

$$u = \sum_{i=0}^{I} w_i x_i \tag{3.1}$$

$$y = \Phi(u) \tag{3.2}$$

Equation 3.2 obtains the value of $y$ when the neural output $u$ is used in an activation function, $\Phi$. $\Phi$ can be any type of activation function where a sigmoid function is commonly used. In this research, a unit step (threshold) function 5.1 where $th$ is the threshold value has been used to ascertain the correct y value which is either low '0' or high '1'. This function is chosen for it can be implemented without the need of a ROM.

$$\Phi(u) = \begin{cases} 0 & u \leq th \\ 1 & u > th \end{cases}$$

When the DPUs are used in a vector arrangement, there will be no need of multiple control units for each DPUs. With a single control unit, multiple number of DPUs can be controlled simultaneously as each DPU in the same layer will perform the same operation. The vector approach of using multiple DPUs in a single layer will be presented in a later section.

## 3.2 Bit-Serial DPU Design

In this research, a novel dedicated neural hardware is proposed. This chapter will focus on describing the DPU design and its work flow. The control unit, a finite state machine (FSM) and counters used to complete the neural operation are also presented. The proposed design is compared with the Stripe design [106] in section .

A modular design is used as this enables each block to be tested and integrated easily using a simple encapsulating module which simplifies the debugging process. The SystemVerilog Hardware Design Language (SystemVerilog HDL) is used in this research. The hardware modules are assembled according to the block diagram 3.1 to implement a single neuron (equation 3.1). The proposed DPU control signals in the data path, $S_H$ and $S_A$ are received from the FSM to complete the add and shift algorithm.

The DPU design includes a synchronous RAM, a single bit full adder, a few general purpose registers, a single AND gate and a number of multiplexers. This design ensures the DPU is a low-cost component in the ANN that is proposed in Chapter 4. A synchronous RAM memory was chosen over a ROM for the purpose of storing the weight

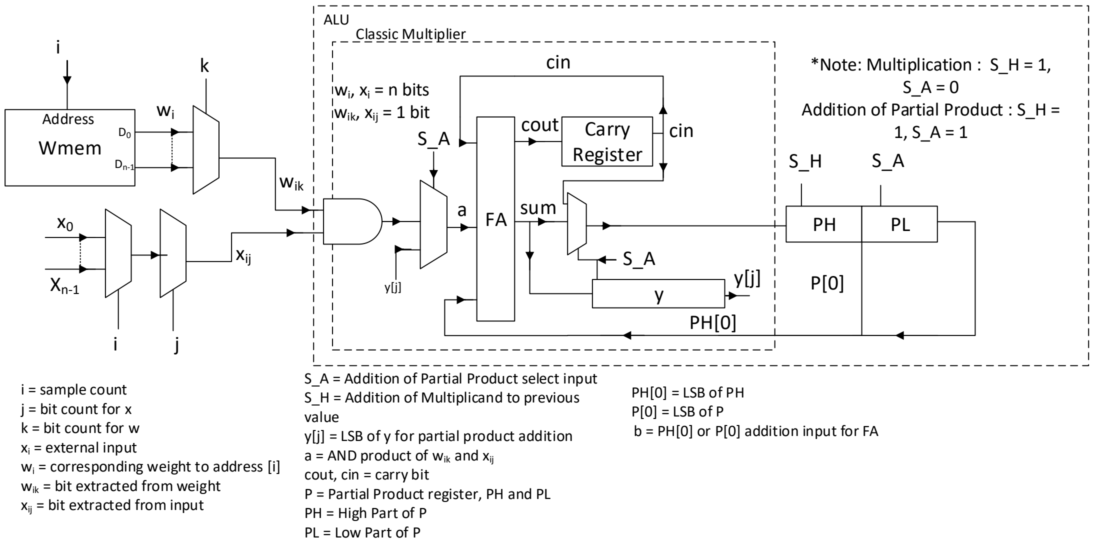

ALU
Classic Multiplier
i
k
Address
Wmem
$D_0$
$D_{n-1}$
$w_i$
$w_{ik}$
$x_0$
$X_{n-1}$
$x_{ij}$
i
j
$w_i$, $x_i$ = n bits
$w_{ik}$, $x_{ij}$ = 1 bit
S_A
a
y[j]
FA
cin
cout
Carry Register
cin
sum
S_A
y
y[j]
PH[0]
*Note: Multiplication : S_H = 1, S_A = 0
Addition of Partial Product : S_H = 1, S_A = 1
S_H
S_A
PH
PL
P[0]
i = sample count
j = bit count for x
k = bit count for w
$x_i$ = external input
$w_i$ = corresponding weight to address [i]
$w_{ik}$ = bit extracted from weight
$x_{ij}$ = bit extracted from input
S_A = Addition of Partial Product select input
S_H = Addition of Multiplicand to previous value
y[j] = LSB of y for partial product addition
a = AND product of $w_{ik}$ and $x_{ij}$
cout, cin = carry bit
P = Partial Product register, PH and PL
PH = High Part of P
PL = Low Part of P
PH[0] = LSB of PH
P[0] = LSB of P
b = PH[0] or P[0] addition input for FA

Figure 3.1: Proposed DPU Design.

(w) values to limit any use of logic elements resulting in a more efficient DPU design. A ROM would be much more costly in terms of hardware logic. The complete ALU is represented as the novelty of the DPU design in this research with the main accumulator $y$ acting as a memory storage unit for the end result of the neural operation.

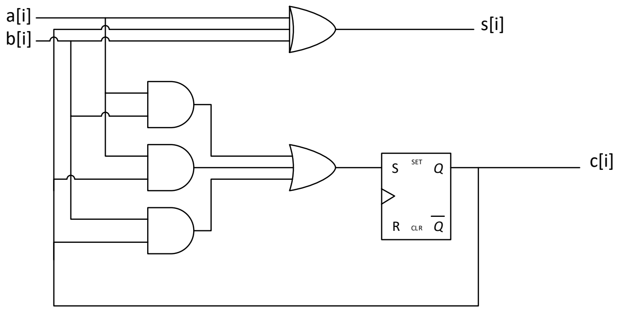


Figure 3.2: Single Bit-adder Circuit in the proposed Bit-Serial Processor.

The sum s[i] is fed into the accumulator (PH, PL) using a serial load operation. The accumulator is a double length general purpose register of 2*n bits used where n is the number of bit representation needed in the neural operation. The carry register input is taken from the full adder (cout) and output (cin) to be processed by the full adder (FA) in the next stage of the bit-serial addition process.

The DPU design uses a classic single bit full adder, Figure 3.2 to perform the addition process in the neural processor. Multiple multiplexers will be needed to complete the DPU circuitry, as shown in Figure 3.1 to accommodate the use of only a single bit full adder for the whole DPU circuit. This DPU design is slower but smaller as compared to a general ALU design which uses parallel loading. The neural output is then stored in the y register before the next operation starts. In order to use the value stored in the y register as input to the next network node, a truncation process takes place by selecting the most significant bit (MSB) of the value stored. This process is completed by right shifting the register value. This novel DPU design will provide an alternative approach in the ongoing research for an efficient home based system needed for the epilepsy detection application.

The Table 3.1 below illustrates an example that performs the bit-serial operation by using this proposed DPU.

The idea that is used in this design is an add and shift multiplier. The classical bit-serial multiplication is performed by the single bit adder (FA), a carry register and a product register (y). The bit's from the multiplicand and multiplier selected using multiplexers. The novelty in this design comes from the use of a single bit adder instead of a k-bit adder in order to save cost and adding a partial product register (P). The connections between each component can be altered at any given time using multiplexers.

### 3.2.1 Hardware Counters for Neural Operation

The FSM includes a basic counter system in order to complete the neural operation (Figure 3.3). The multiple loops represented in the counter system (Figure 3.4) are simple while loops.

where:

- i = number of samples
- j = bit number in input(x), j = 0,1,... n-1
- k = bit number in weight(w), k = 0,1,... n-1
- p = bit number in output (u), p = 0,1, ... n-1
- add = signal to begin counter addition
- $S_A$ = signal from FSM to undergo addition of partial product to end product

Conditions associated with the counters are:

- When j reaches n, j resets to 0 and k increments by 1
- Once k reaches n, p increments till it reaches 2*n - 1
- When p reaches 2*n - 1, reset j, k, p and increment i by 1 till i reach maximum number of samples.

| **Description** | i | j | k | w[ik] | x[ij] | a | sum | PH | PL | y |
|---|---|---|---|---|---|---|---|---|---|---|
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Add a to PH[0] and shift sum to PH[msb] | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| Add a to PH[0] and shift sum to PH[msb] |  | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Add a to PH[0] and shift sum to PH[msb] |  | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Shift P by 1 bit to the right |  |  |  |  |  | 0 | 0 | 0 | 0 | 0 |
| Add a to PH[0] and shift sum to PH[msb] |  | 0 | 1 | 1 | 1 | 1 | 1 | 100 | 0 | 0 |
| Add a to PH[0] and shift sum to PH[msb] |  | 1 | 1 | 1 | 0 | 0 | 0 | 10 | 0 | 0 |
| Add a to PH[0] and shift sum to PH[msb] |  | 2 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| Shift P by 1 bit to the right |  |  |  |  |  | 0 | 0 | 0 | 100 | 0 |
| Add a to PH[0] and shift sum to PH[msb] |  | 0 | 2 | 0 | 1 | 0 | 0 | 100 | 0 |  |
| Add a to PH[0] and shift sum to PH[msb] |  | 1 | 2 | 0 | 0 | 0 | 0 | 100 | 0 |  |
| Add a to PH[0] and shift sum to PH[msb] |  | 2 | 2 | 0 | 0 | 0 | 0 | 100 | 0 |  |
| Shift P by 1 bit to the right |  |  |  |  |  | 0 | 0 | 0 | 10 | 0 |
| Add y[j] to P[j] and shift sum into y[msb] |  | 0 |  |  |  |  |  | 0 | 1 | 0 |
| Add y[j] to P[j] and shift sum into y[msb] |  | 1 |  |  |  |  |  | 0 | 0 | 100000 |
| Add y[j] to P[j] and shift sum into y[msb] |  | 2 |  |  |  |  |  | 0 | 0 | 10000 |
| Add y[j] to P[j] and shift sum into y[msb] |  | 3 |  |  |  |  |  | 0 | 0 | 1000 |
| Add y[j] to P[j] and shift sum into y[msb] |  | 4 |  |  |  |  |  | 0 | 0 | 100 |
| Add y[j] to P[j] and shift sum into y[msb] |  | 5 |  |  |  |  |  | 0 | 0 | 10 |
| Add a to PH[0] and shift sum to PH[msb] | 2 | 0 | 0 | 1 | 1 | 1 | 1 | 100 | 0 | 10 |
| Add a to PH[0] and shift sum to PH[msb] |  | 1 | 0 | 1 | 0 | 0 | 0 | 10 | 0 | 10 |
| Add a to PH[0] and shift sum to PH[msb] |  | 2 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 10 |
| Shift P by 1 bit to the right |  |  |  |  |  | 0 | 0 | 0 | 100 | 10 |
| Add a to PH[0] and shift sum to PH[msb] |  | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 100 | 10 |
| Add a to PH[0] and shift sum to PH[msb] |  | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 100 | 10 |
| Add a to PH[0] and shift sum to PH[msb] |  | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 100 | 10 |
| Shift P by 1 bit to the right |  |  |  |  |  | 0 | 0 | 0 | 10 | 10 |
| Add a to PH[0] and shift sum to PH[msb] |  | 0 | 2 | 0 | 1 | 0 | 0 | 0 | 10 | 10 |
| Add a to PH[0] and shift sum to PH[msb] |  | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 10 | 10 |
| Add a to PH[0] and shift sum to PH[msb] |  | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 10 | 10 |
| Shift P by 1 bit to the right |  |  |  |  |  | 0 | 0 | 0 | 1 | 10 |
| Add y[j] to P[j] and shift sum into y[msb] |  | 0 |  |  |  |  | 1 | 0 | 0 | 100001 |
| Add y[j] to P[j] and shift sum into y[msb] |  | 1 |  |  |  |  | 1 | 0 | 0 | 110000 |
| Add y[j] to P[j] and shift sum into y[msb] |  | 2 |  |  |  |  | 0 | 0 | 0 | 11000 |
| Add y[j] to P[j] and shift sum into y[msb] |  | 3 |  |  |  |  | 0 | 0 | 0 | 1100 |
| Add y[j] to P[j] and shift sum into y[msb] |  | 4 |  |  |  |  | 0 | 0 | 0 | 110 |
| Add y[j] to P[j] and shift sum into y[msb] |  | 5 |  |  |  |  | 0 | 0 | 0 | 11 |
| Final answer in register y |  |  |  |  |  |  |  |  |  |  |

Table 3.1: Table: Example of the bit-serial operation performed using the proposed DPU using 3 bit precision. (P = [PH +PL], w1 = 2, x1 = 1, w2 = 1, x2 = 1).

### 3.2.2 Layer Finite State Machine (FSM)

The DPUs are controlled using a single layer FSM. The FSM has a default START stage where no operations are carried out. When the nreset signal from the central control

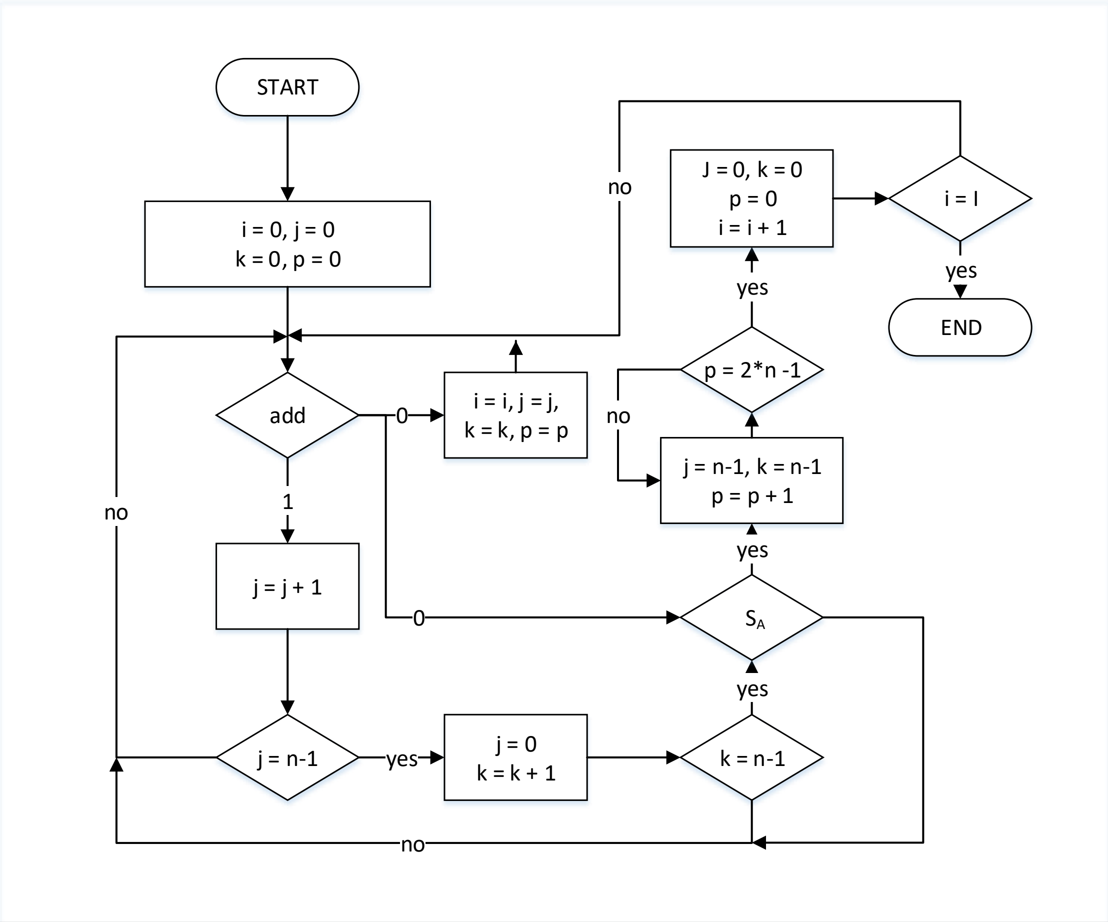
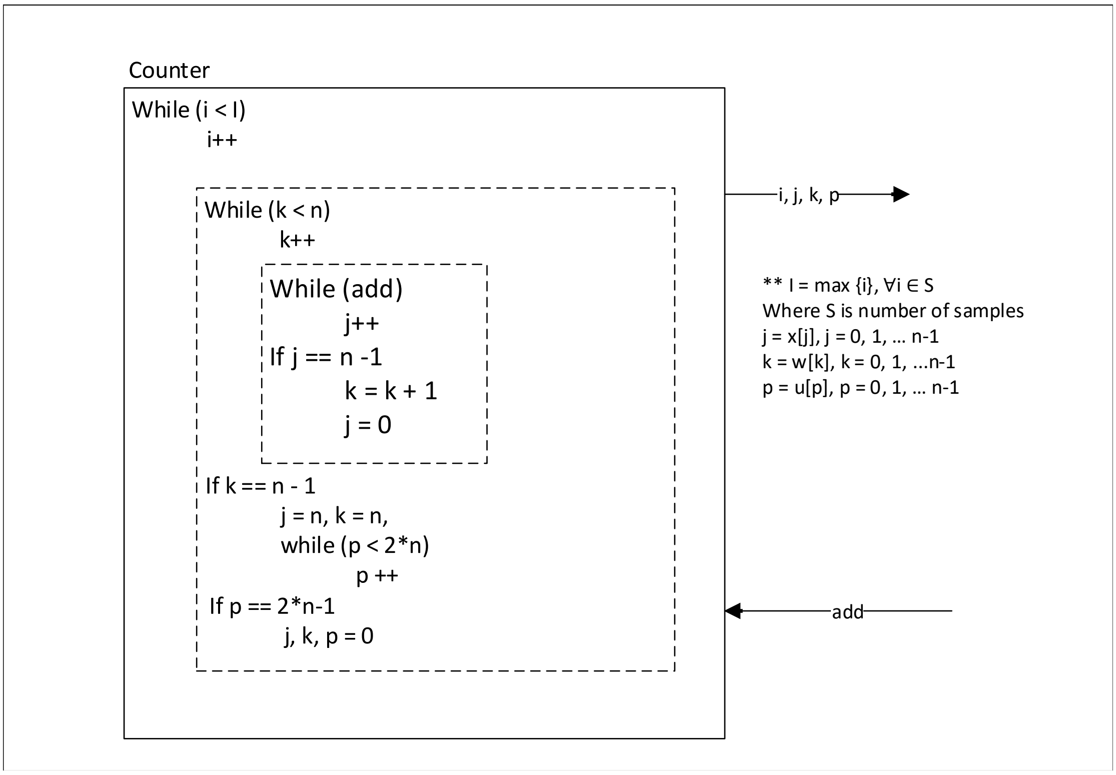


Figure 3.3: Counters algorithm for the hardware neural network

FSM is low ‘0’, the state machine proceeds to the LOAD stage. In the LOAD stage, the inputs, $x_i$ are loaded from the previous neuron / feature extraction hardware, and the weights,$w_i$ from the hex file (extracted from training) into the DPU. The FSM sends the $S_H$ signal to provide an addition function to the data processing unit (DPU) during the ADD stage. As the values of $x$ and $w$ are multiplied in a bit-serial format, the bit-wise operations are controlled by the counter system (Figure 3.3, 3.4).

Once the calculations were completed for each sample (k = n-1), the FSM sends $S_A$ to the DPU which allows the partial product to be added to the final product registers. When the total number of $i$ inputs have been calculated, the hidden layer FSM will send a $DONE_H$ signal to the central control FSM to trigger the output layer to begin operations ($nreset_O$ signal). The hidden layer will then revert to the START stage to await further instructions. The mechanism of the central control FSM will be explained in Chapter 4. Finally, the $DONE_O$ signal will indicate that the output layer has also completed the calculations which will then stop the device.

The timing diagram for the processor interaction between the control unit and the DPU for an individual input (Figure 3.6). The global clock (clk) is used as a reference point with the rising edge and falling edge for different signal activations. When the GO signal



```
While (i < I)
        i++
    While (k < n)
            k++
        While (add)
                j++
        If j == n -1
                k = k + 1
                j = 0
    If k == n - 1
            j = n, k = n,
            while (p < 2*n)
                    p ++
    If p == 2*n-1
            j, k, p = 0
```

** I = max {i}, ∀i ∈ S
Where S is number of samples
j = x[j], j = 0, 1, … n-1
k = w[k], k = 0, 1, …n-1
p = u[p], p = 0, 1, … n-1

Figure 3.4: Multiple counters diagram for the hardware neural network

goes high, the Mreset signal will go low. The output of the central control FSM (nreset) will begin the FSM operations which sends a $S_H$ or a $S_A$ signal to perform the add and shift algorithm using the inputs, $x$ and weights, $w$. The accumulator product will be recorded in $u_0$. The same process will be repeated for any number of samples.

## 3.3 DPU Verification

### 3.3.1 Simple Classification with BSNN

The next stage involves functionality testing for the DPU when used as the basic building block of an artificial neural network. Two simple neural networks (Figure 3.7, 3.8) have been designed to solve some general classification problems. The first problem uses the network (Figure 3.7) to determine three different types of class (van, lorry or car) using two different inputs (mass (M) and length (L)). The output of the network presents the probability that the input has been correctly classified. The network is an input output design which does not include a hidden layer. The output layer consists of 3 DPUs. The weights ($w_M$ and $w_L$) are clearly labelled next to the arrows.

The next test involves logic gate verification. The XOR gate was chosen for this test(Figure 3.8). The design has two layers with a single DPU in both the hidden and output layers. It has two inputs (a,b) and the weights are clearly labelled $w_1$, $w_2$,

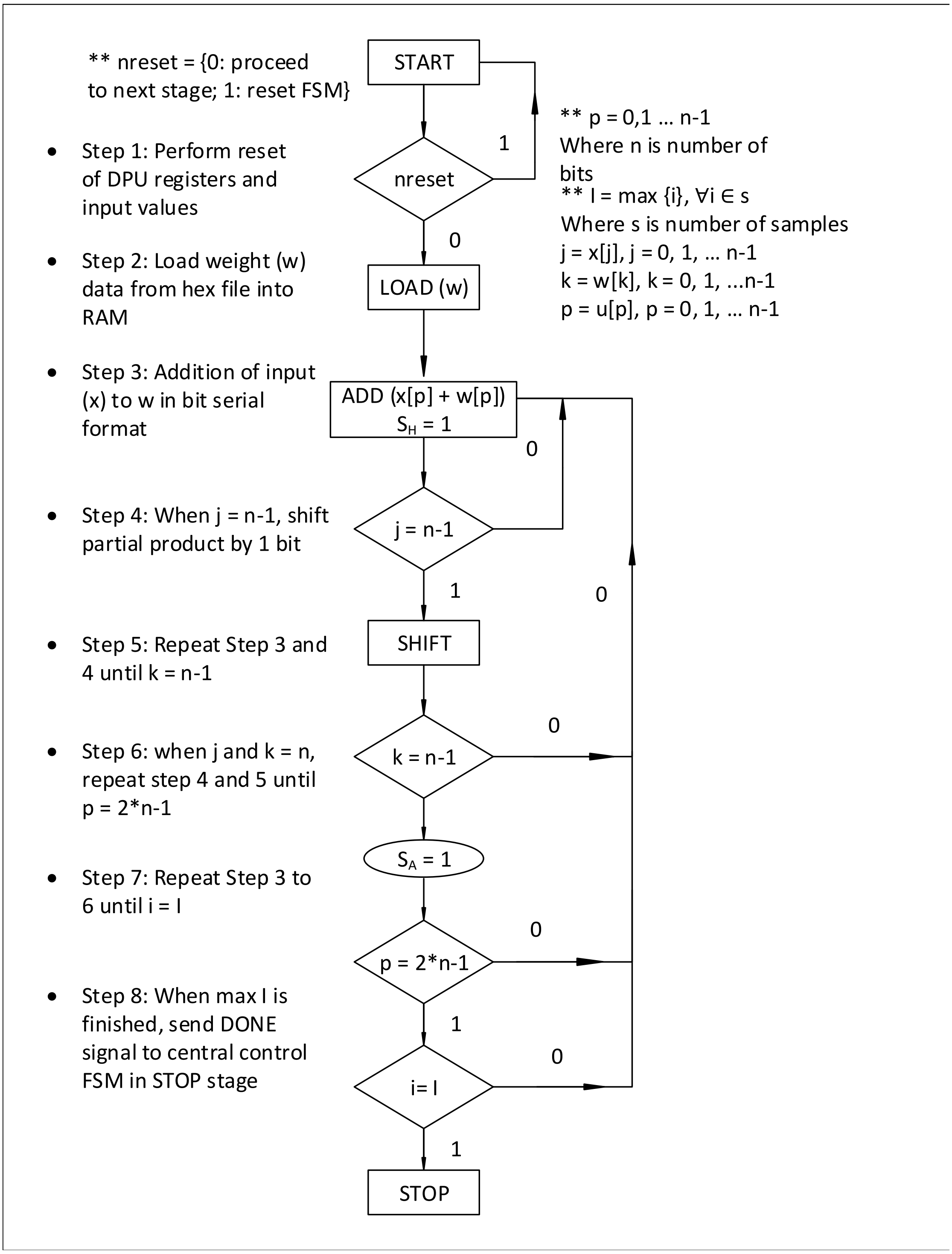

** nreset = {0: proceed to next stage; 1: reset FSM}
Step 1: Perform reset of DPU registers and input values
Step 2: Load weight (w) data from hex file into RAM
Step 3: Addition of input (x) to w in bit serial format
Step 4: When j = n-1, shift partial product by 1 bit
Step 5: Repeat Step 3 and 4 until k = n-1
Step 6: when j and k = n, repeat step 4 and 5 until p = 2*n-1
Step 7: Repeat Step 3 to 6 until i = I
Step 8: When max I is finished, send DONE signal to central control FSM in STOP stage
START
nreset
1
0
LOAD (w)
ADD (x[p] + w[p])
S_H = 1
0
j = n-1
1
SHIFT
0
k = n-1
0
S_A = 1
0
p = 2*n-1
1
0
i= I
1
STOP
** p = 0,1 … n-1
Where n is number of bits
** I = max {i}, ∀i ∈ s
Where s is number of samples
j = x[j], j = 0, 1, … n-1
k = w[k], k = 0, 1, …n-1
p = u[p], p = 0, 1, … n-1

Figure 3.5: A finite state machine for the Layer FSM

$w_3$, $w_4$, $w_5$ throughout the network. The output corresponds exactly to the truth table of a common XOR gate.

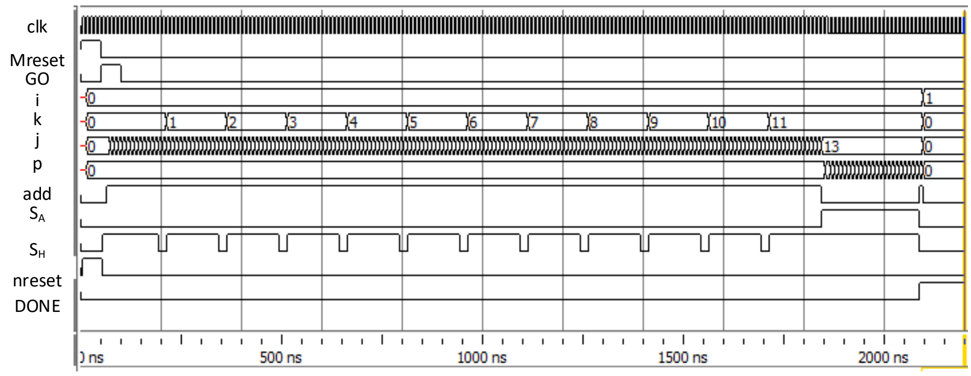


Figure 3.6: Timing diagram of a single sample neural operation (clk = global clock, Mreset = Master reset signal, i = sample number, k = bit in weight, j = bit in input x, p = accumulated producted, u, add = add signal, $S_H$ , $S_A$ = signal from the FSM to perform shift and add algorithm, nreset = nreset signal, DONE = signal in every layer of the neural network to stop operation )

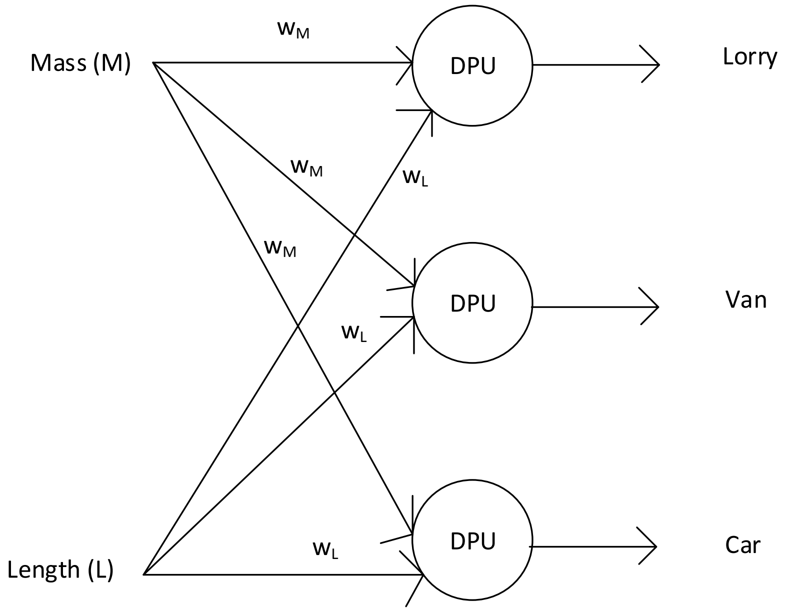


Figure 3.7: Simple Single Layer Neural Network (Mass (M), Length (L) as inputs and $w_i$ = weights)

### 3.3.2 Peak Detection Using the Proposed Vector Processor Design

A two layer neural network was designed for peak detection with ECG inputs to examine the functionality of the proposed vector processor in the field of medicine. For this peak detection problem, a sample was extracted from the ECG input waveform with 1000 consecutive data points. The design has been included in Figure 3.9 where $a$ and $b$

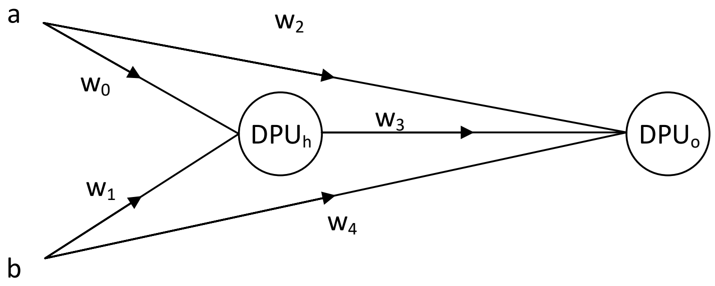


Figure 3.8: XOR gate double Layer Neural Network (a and b as inputs and $w_i$ = weights)

are different thresholds. $a$ and $b$ can be 0,1 in the neural network where 1 indicates it has passed the thresholds and vice versa. These two threshold allow the output to be differentiated into three separate classes, 0,1,2. If the output moves between classes (0 to 1, 0 to 2, 1 to 2), the output is registered as a peak.

Figure 3.10 shows the plot of an ECG input which clearly illustrates seven different peaks. Thus, the different peaks can be confirmed by reviewing Figure 3.11 and the result from the device. All weights and inputs are truncated into the form of integers allowing simpler calculations. However, the truncated values will affect the accuracy of the results. The input and weight data representation can be reviewed in section 4.2.

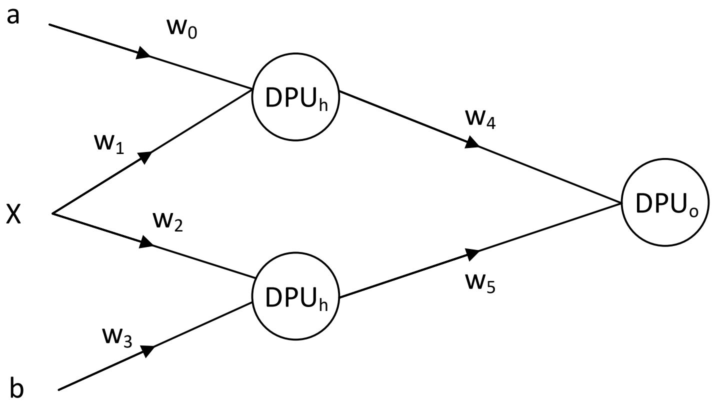


Figure 3.9: Simple Multi Layer Neural Network for ECG Plot Peak Detection (a, X, b as inputs and $w_i$ = weights)

Another experiment was also conducted using a 1-8-1 network topology. This experiment goal is to fit the input onto a target output waveform. This experiment has also been conducted using different bit architecture (6,8,12,16 bit). The figures below shows the results of the software output and the different hardware outputs.

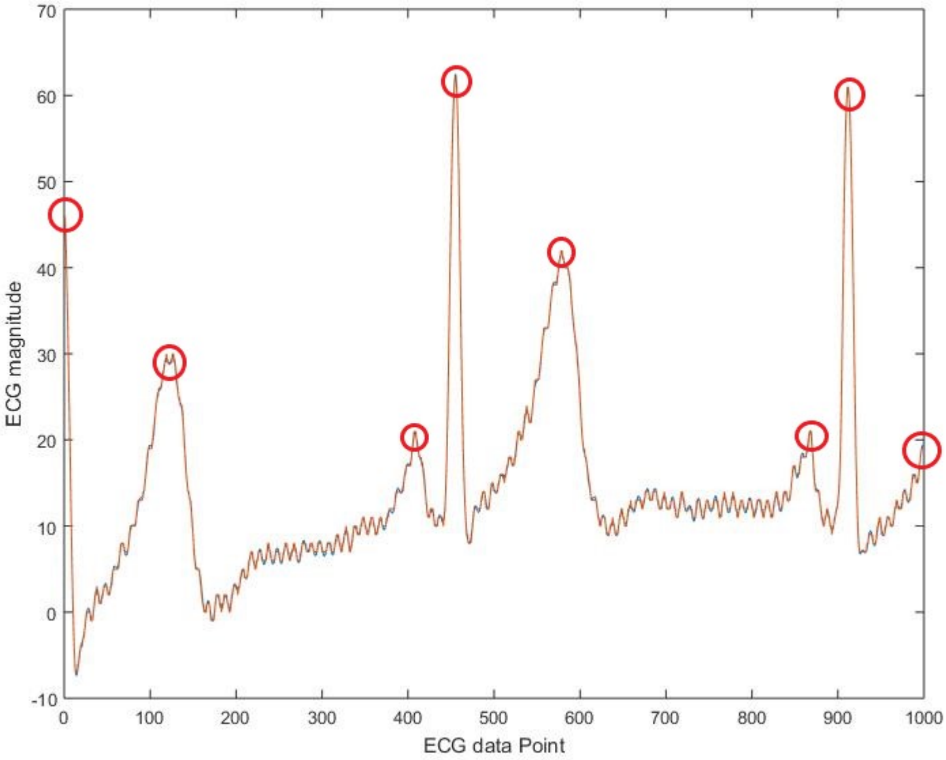


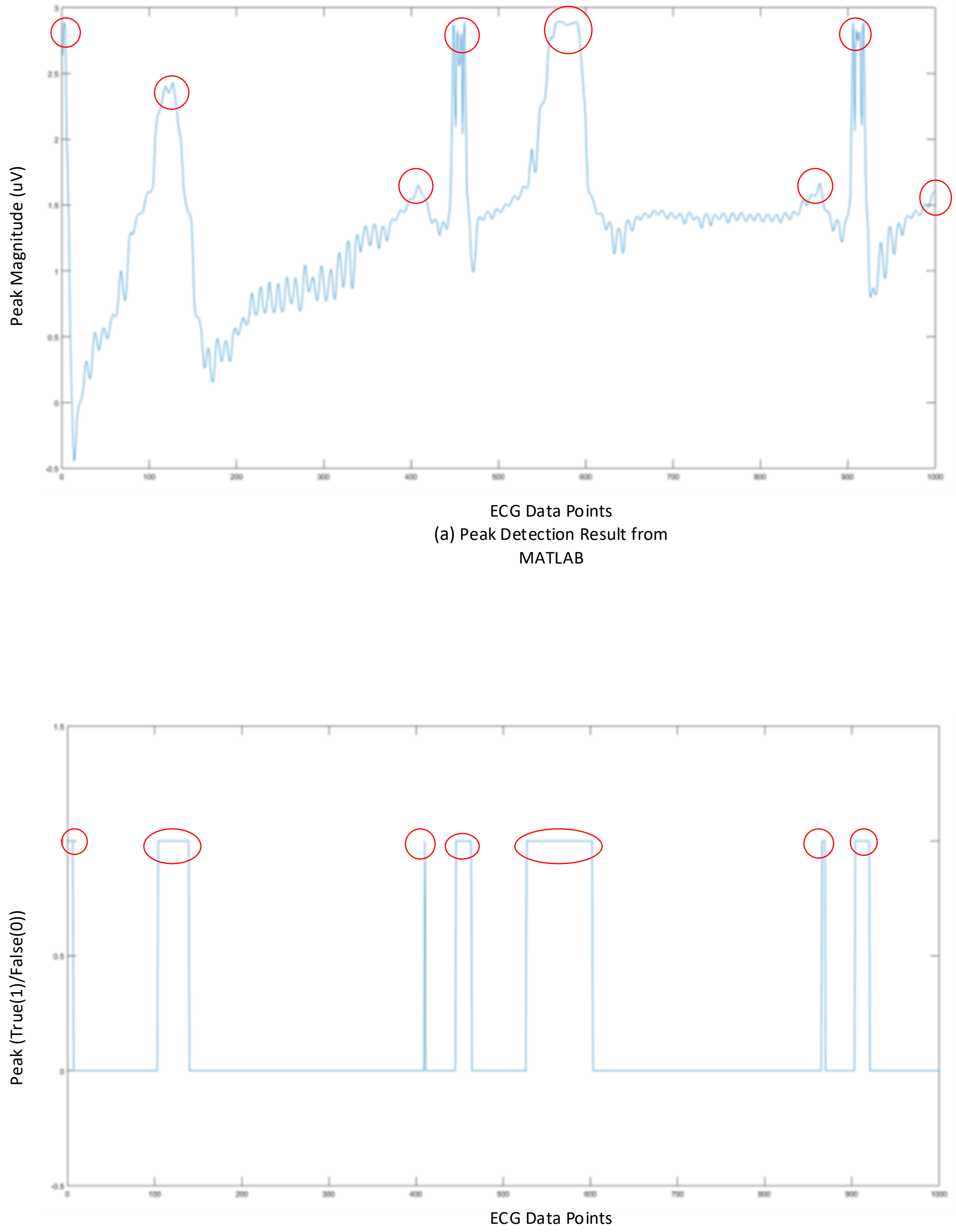
Figure 3.10: ECG Plot For Simple Peak Detection

### 3.3.3 Bit-Serial DPU FPGA Synthesis

With the novel approach, it is possible to develop a specialized neural processor which can provide significant advantages in terms of cost and accessibility. This reduces the need for complex simulation software that requires high computational power. The synthesised hardware can reproduce the results of the simulation software within an acceptable error margin with an accuracy over 85% and complete the necessary operation within a reasonable time frame. The results are presented in Chapter 4. However, there is a trade off between the speed and cost during development. This feature allows the use of low bit resolution yet able to produce reliable results.

The synthesis of any hardware prototype was completed using a hardware synthesis software (Quartus II). The RTL of the device needs to be reviewed to confirm the correct hardware has been synthesised. The cost for the proposed bit-serial designs is vital to this research goal by reducing the number of logic elements. The proposed approach includes variable bit precision. The logic elements needed for each DPU is presented in Table 3.2 using variable bit precision.

In the next chapter, the synthesised DPU model will be used as network nodes in a bit-serial ANN. The network will then be used for epilepsy prediction based on EEG



(a) Peak Detection Result from MATLAB

(b) Peak Detection Result From Hardware

Figure 3.11: Peak Result obtained from (a) MATLAB & (b) Hardware to be compared against Figure 3.10

readings of a single patient.

By analysing Table 3.2, 3.3, 3.4, it is clear that Cyclone IV FPGAs require additional hardware compared to Cyclone V and Stratix IV FPGAs. Next, Figure 3.17, 3.18, 3.19 presents a clear comparison of the hardware cost between the three different FPGAs.

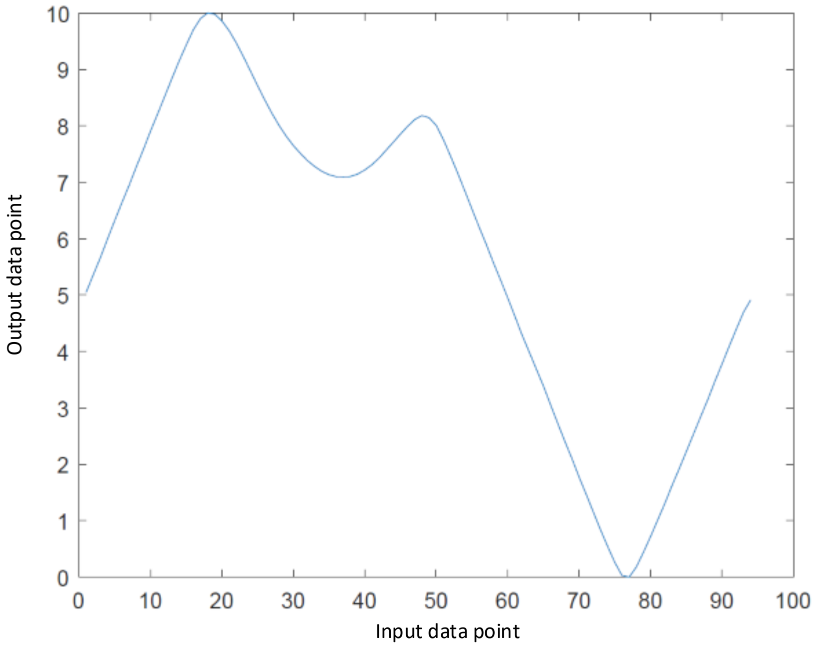


Figure 3.12: Target output waveform to be matched by hardware

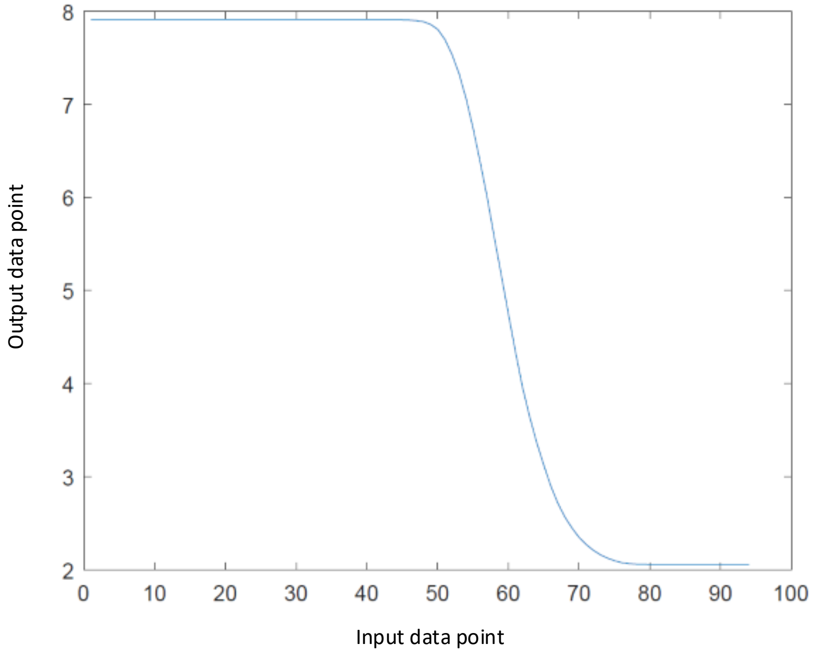


Figure 3.13: Output Waveform obtained from using 6-bit precision 1-8-1 hardware

From the graphs, it can be seen clearly that Stratix IV will require more Altera Logic Elements (LE) if a 16 bit precision or higher is used for a single DPU. Furthermore, the total cost of a single neuron and DPU starts to flattened out once a higher bit precision

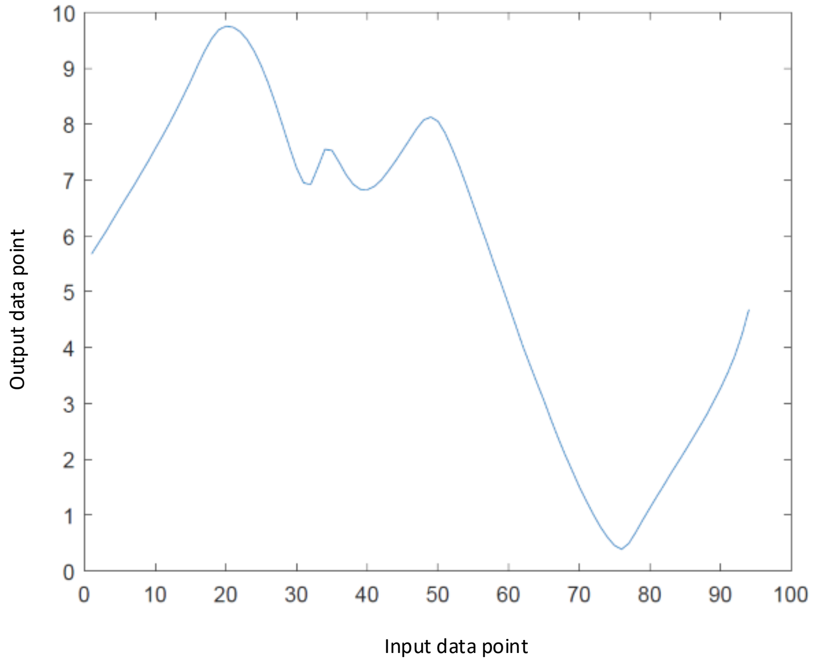


Figure 3.14: Output Waveform obtained from using 8-bit precision 1-8-1 hardware

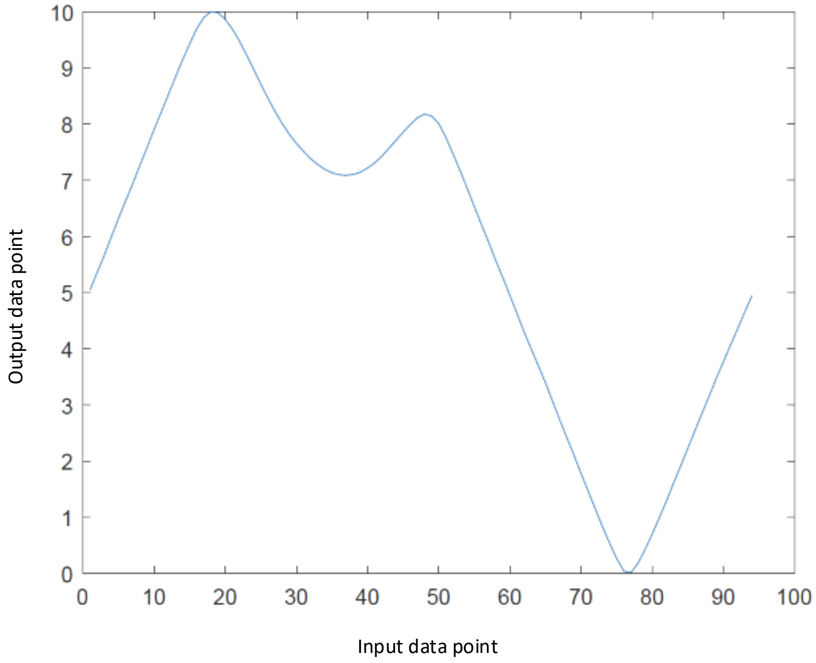


Figure 3.15: SOutput Waveform obtained from using 12-bit precision 1-8-1 hardware

is used on any Cyclone FPGA. This could be attributed to the fact that certain logic

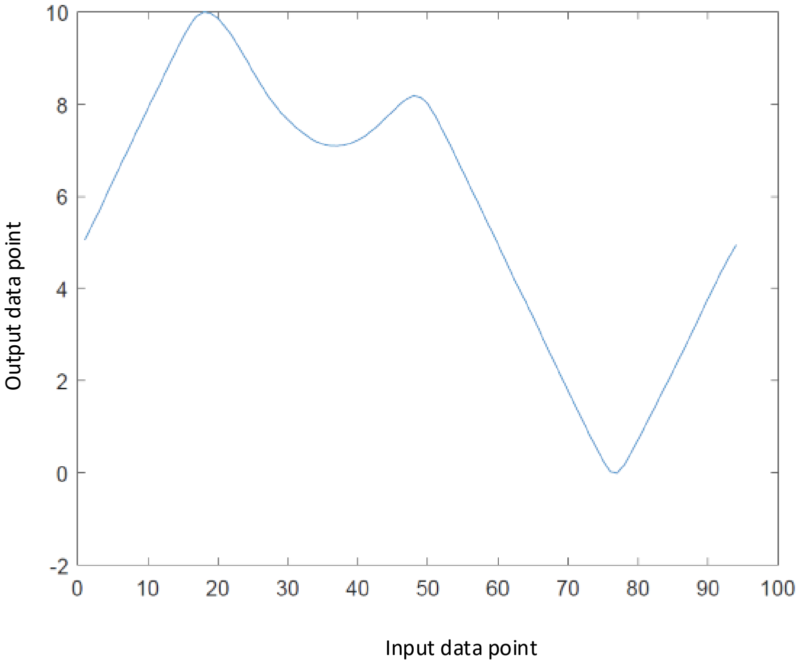


Figure 3.16: Output Waveform obtained from using 16-bit precision 1-8-1 hardware

| bit precision | | Logic Element Cost | |
|---|---|---|---|
| | Cyclone IV | Cyclone V | Stratix IV |
| 4 | 42 | 17 | 20 |
| 6 | 48 | 24 | 20 |
| 8 | 57 | 25 | 28 |
| 12 | 87 | 39 | 33 |
| 16 | 131 | 41 | 48 |
| 32 | 217 | 74 | 79 |
| 64 | 249 | 85 | 142 |

Table 3.2: Logic elements needed for DPU tested with three different FPGA technologies

gates might have been reused across the network.

After analysing the results, Cyclone V FPGA is the better choice for this research. This particular board has sufficient computational power to accommodate the testing of the hardware neuron. The proposed hardware neuron is simulated using MATLAB. Then, the FPGA is used in this research to implement our designs as it has the advantage of reprogramming various designs, unlike a GPU or ASIC chip.

Different hardware implementation methodologies have also been considered. These include an ASIC approach, and a complex programmable logic device (CPLD). The ASIC architecture proved to be more energy efficient and have a lower hardware cost, yet an

| **bit precision** | | **Logic Element Cost** | |
|---|---|---|---|
| | **Cyclone IV** | **Cyclone V** | **Stratix IV** |
| 4 | 21 | 15 | 15 |
| 6 | 25 | 19 | 18 |
| 8 | 29 | 23 | 20 |
| 12 | 35 | 25 | 24 |
| 16 | 42 | 30 | 30 |
| 32 | 68 | 41 | 43 |
| 64 | 123 | 84 | 81 |

Table 3.3: Logic elements needed for FSM tested with three different FPGA technologies

| **bit precision** | | **Logic Element Cost** | |
|---|---|---|---|
| | **Cyclone IV** | **Cyclone V** | **Stratix IV** |
| 4 | 57 | 50 | 48 |
| 6 | 82 | 68 | 66 |
| 8 | 106 | 80 | 81 |
| 12 | 191 | 155 | 153 |
| 16 | 244 | 165 | 166 |
| 32 | 463 | 182 | 296 |
| 64 | 700 | 291 | 559 |

Table 3.4: Logic elements needed for a single neuron tested with three different FPGA technologies

ASIC is fixed thus preventing it from implementing a prototype that needs to be constantly modified. The CPLD architecture has a less flexible design due to its restrictive structure. Therefore, the FPGA is the best choice for implementing the proposed design. In the near future, the ASIC architecture will be considered for implementation. The details of this approach is presented in the future work section of this thesis.

In order to validate the functionality of the bit-serial processor, different tests have been conducted and completed at various stages during the research. The first stage is the testing of a single DPU design using a n-bit architecture. The addition process will require $2^n + 1$ clock cycles ($2^n$ clock cycles to complete the whole addition and 1 clock cycle to save the data). The tests are conducted using simulation software and the results validated with the results from the synthesised hardware.

## 3.4 Discussion and Comparison

There are different methods for neural processor implementation that includes analog and digital techniques. There are also hybrid methods that use the best of the two

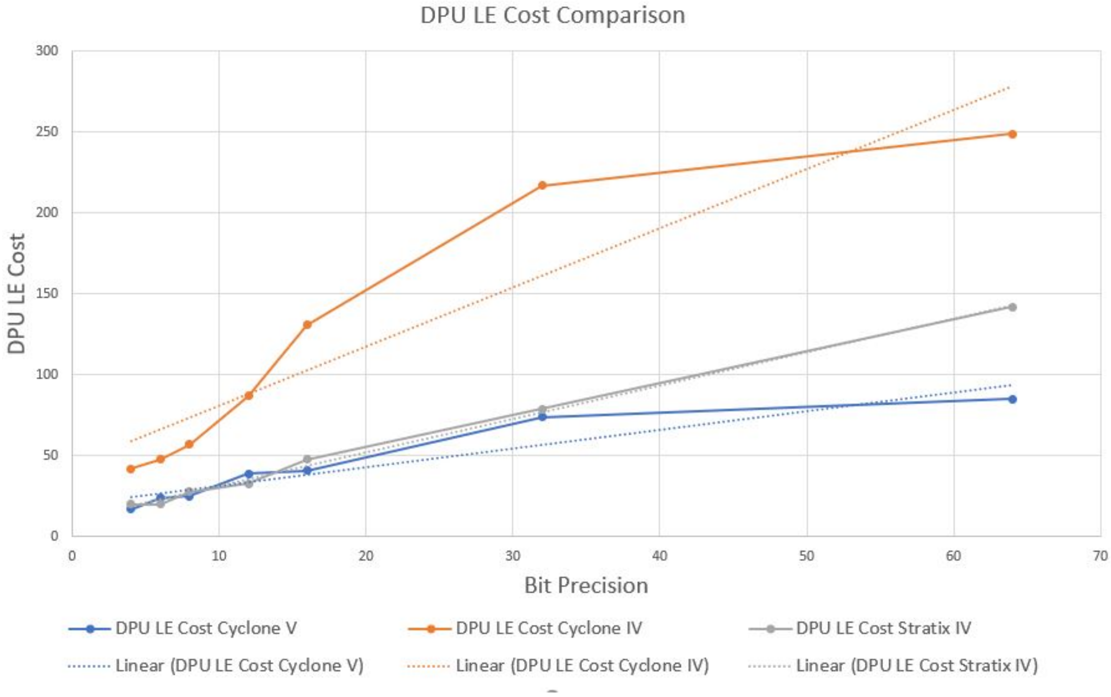


Figure 3.17: DPU LE Cost Comparison Between Cyclone IV, Cyclone V and Stratix IV

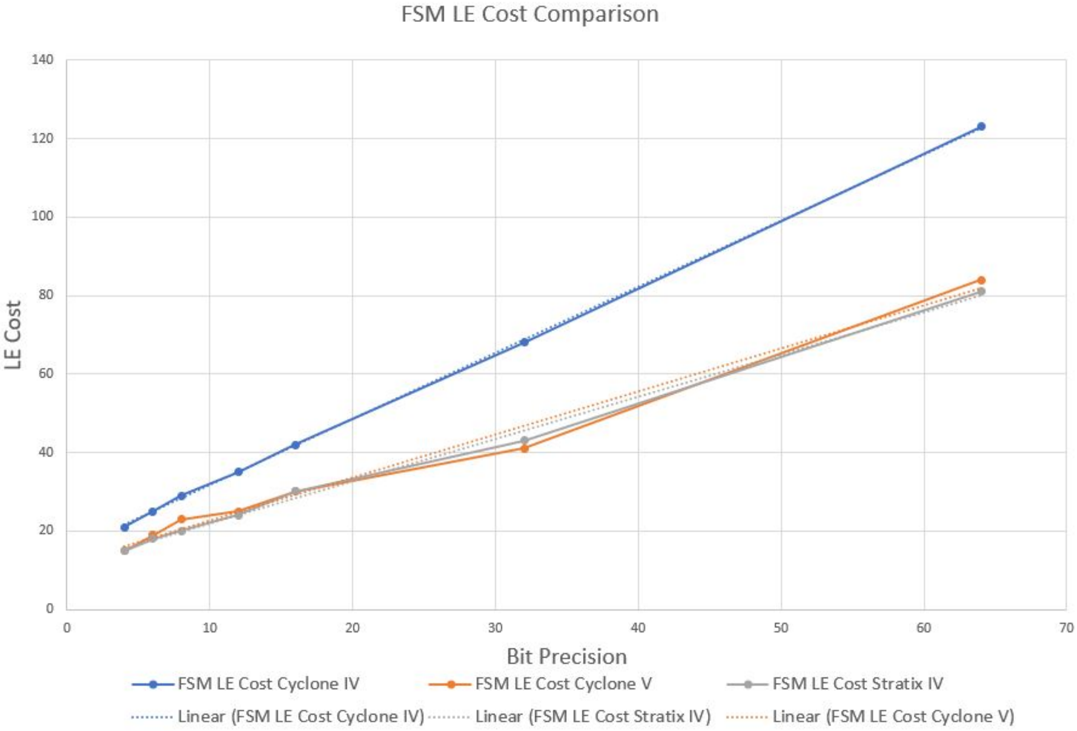


Figure 3.18: FSM LE Cost Comparison Between Cyclone IV, Cyclone V and Stratix IV

approaches. The analog methodology would need certain circuit elements as a candidate for the synapse element. One such candidate is the ferroelectric memristor [127].

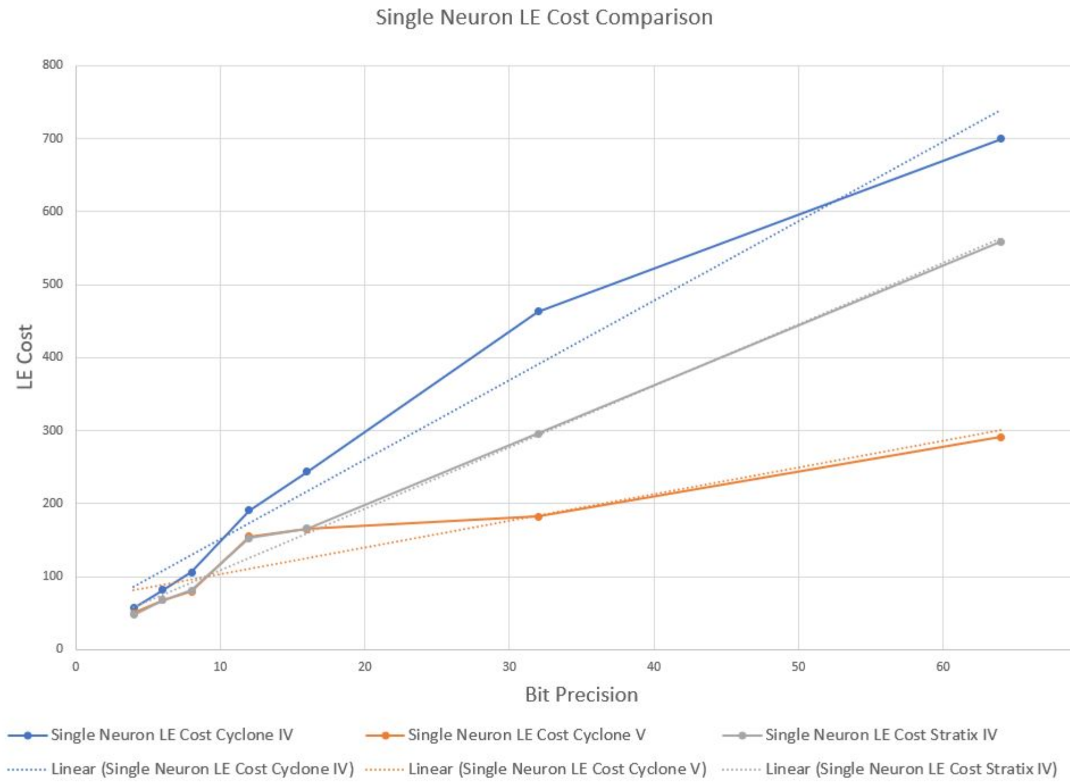


Figure 3.19: Single Layer LE Cost Comparison Between Cyclone IV, Cyclone V and Stratix IV

However, there is difficulty in controlling the accurate conductance for this device. In comparison, the digital approach offers accurate precision and reprogrammability. Such an approach would be more suitable for our research as there is a need to deal with large EEG datasets (6576 sample waveforms) and accurate detection (minimum recognition rate of 80%). Therefore, the digital approach has been chosen to implement the dedicated DPU.

As part of the evaluation process, few experiments have been conducted to test the functionality of this DPU. The first experiment involves the classification of three different vehicles lorry, car or van depending on the inputs mass or length. The weights were relatively straightforward and there were no issues for the hardware network to classify the samples. This also apply when dealing with the case of the XOR logic gate experiment.

For the last two experiment, the results were more interesting. The third experiment involved the use of an ECG plot. It can be seen clearly that from the Figure 3.10 that eight peaks are clearly shown in the ECG plot and in Figure 3.11 (a) MATLAB. The hardware Figure 3.11 (b) could only detect 7 out of 8 peaks which provides a recognition rate of 87% for such a simple neural network. Furthermore, this was achieved with 12-bit precision. This indicates that this DPU can be used in a scenario where peak detection is required such as epilepsy detection.

The last experiment involves the matching of input data points to output data points. Figure 3.13 - 3.16 waveforms are results extracted using the proposed hardware. 12-bit and 16-bit precision matches the target waveform, but 6-bit and 8-bit precision have some mismatching issues. Therefore, the hardware neural network is feasible but minimum bit precision for accepted recognition rate must be maintained.

In this research, few revisions have been made to the bit-serial DPU design. The first design was found to be inefficient in terms of hardware cost when compared with the DPU illustrated in section 3.2. Figure 3.21 below presents our first design which was published in 2016. The design itself includes two ALUs, various registers (Wmem, Res1 and Mreg) to complete the neural operation. Wmem store the weight values and Mreg the input values. Res1 then store the partial product values. These components were simplified in the final design. This design was programmed with simple machine codes which meant more hardware costs for the control path. In Chapter 4, the control path of the final design will be illustrated and explained.

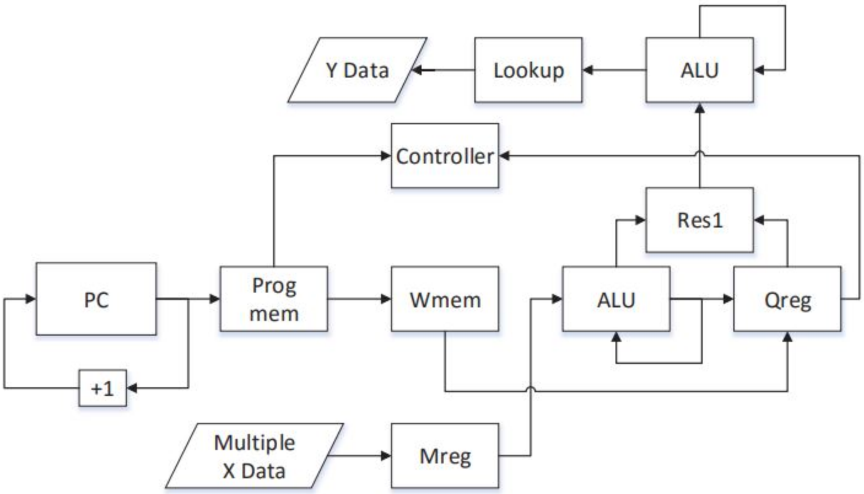


Figure 3.20: First DPU Design Published in WASET Paper [128]

Table 3.5 shows that an 8-bit DPU requires only 24 Logic Elements (LEs) on an inexpensive Altera Cyclone V FPGA, out of over 300,000 LEs available on a Cyclone V chip. This compares favourably with the size of the data paths of typical bit-serial processors mentioned in the Table. Bearing in mind that the control logic of the proposed approach requires only simple state machines, rather than fully-fledged program control paths used in general-purpose processors, the expected overall benefits of an ASIC implementation will include faster operation and lower power consumption.

The proposed design is compared with Stripes [106] proposed by P. Judd and his research team. However, the Stripes design trades precision for energy and performance at the expense of the chip area. The proposed DPU in this thesis trades speed for precision and area instead. With the power analyser tool provided in the Quartus II Altera software, a

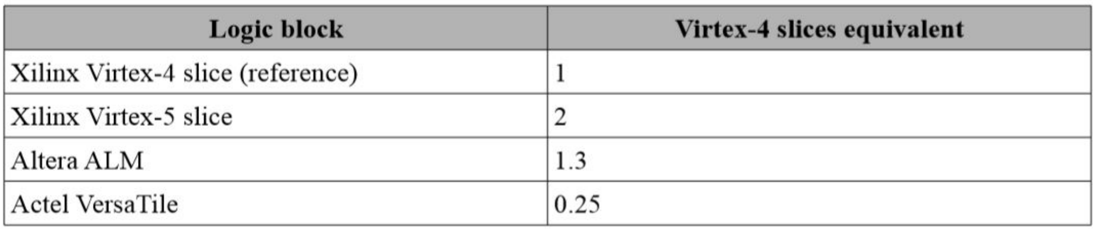
| Hardware | Development Chip | LE Count |
|---|---|---|
| Bit Array [129] Processor | ASIC | 56 Altera Equivalent LEs |
| Cellular Processor [130] (Data Path) | Virtex 5 | 26 Altera equivalent LEs |
| Proposed Neural Processor | Cyclone V | 25 LEs |

Table 3.5: Cost comparison between three different processors.

single proposed DPU is estimated to use 5.81 mW. The FSM consumes 0.1 mW. These estimated values are of a whole system rather than that of a critical path.

The main difference between Stripe and our design is in the design itself. Stripe is a modification of DaDianNao, a DNN accelerator which focus on improving the energy efficiency and speed. The design proposed in this chapter instead focus on reducing the area while maintaining an accuracy over 80%. Next, Stripe has been implemented on a chip while our design is still being implemented on an FPGA as a prototype. Therefore, direct comparison is not suitable at this point in the research. Direct comparison can be made at a later stage when a dedicated chip has been fabricated for our proposed design.

In this section, comparison will be made in the context of different design decisions. The Stripes design uses a dedicated neuron memory while our proposed design utilizes synchronous RAM to store weight values. Next, the Stripes design includes the convolutional layer of the DNN whereas the thesis focusses only on producing a bit-serial FNN. Equivalent LE a term used to try to bridge the world from ASIC  discrete logic to how FPGA function with their slices and lookup tables. One slice could be used to create a single AND gate or a to some extent part of a larger adder. By rationalising the equivalent logical gates required is some pseudo way of marketing their size.



| Logic block | Virtex-4 slices equivalent |
|---|---|
| Xilinx Virtex-4 slice (reference) | 1 |
| Xilinx Virtex-5 slice | 2 |
| Altera ALM | 1.3 |
| Actel VersaTile | 0.25 |

Figure 3.21: Table for logic cell comparison (Extracted from datasheet [131]

The table above is extracted from a data sheet [131] where slices equivalent was made that can be used to compare Xilinx Virtex-4 and Altera ALM. The author used this comparison to make the LE equivalent comparison between his design and other designs found in research literatures. By using this form of comparison, the author hopes to achieve some form of fair comparison across different designs.

## 3.5 Conclusive Remarks

In this chapter, a novel approach of using a dedicated neural processor to detect epilepsy has been presented. The novelty lies within the use of bit-serial architecture when designing the dedicated data processing unit (DPU). The DPU consists of different basic electronic components such as an AND gate, multiple multiplexers and a single bit full adder. This full adder will perform the addition and multiplication process for the neural operation. A few registers have been used to complete the processor. This DPU presents the implementation of the functionality of a single neuron. Multiple identical units are connected to a finite state machine (FSM) that provides the algorithm for the DPUs. Simple experiments were conducted in this stage of the research to fully ascertain the feasibility of the proposed design and various bit precision were also used. The design was synthesised on various FPGAs to find the best suited FPGA for this research.

The author considers that simple verification is needed to show that DPU can perform individually which had led to the decision of using simple verification experiments as a verification platform for the proposed DPU. Once the simulation process is completed, the processor is synthesised and the cost of the processor proves that the basic vector processor is a small device with a minimum of 80 logic elements for a single neuron using 8 bit architecture. In the next chapter, the author will use the proposed DPU as the basic building block to develop various forms of ANNs for epilepsy prediction.

# Chapter 4

# Bit-serial Based Hardware Neural Network for Epilepsy Detection

This chapter presents a novel approach for hardware-based epilepsy detection. As presented in the last chapter, bit-serial architecture was chosen to model this artificial neural network hardware as bit-serial computing has been the interest of ultra low energy consumption processor designers in recent years.

The chapter is organized as follows: Section 4.1 presents the basic understanding of implementing a hardware neural network. Section 4.2 proposes the approach of implementing a hardware neural network with the main focus on the novel data processing unit (DPU) presented in Chapter 3. The section also provide some basic explanation of the bit-serial computation used in the design. This will provide a simple, low cost yet efficient way of classifying a seizure event. Section 4.3 discusses the training and testing of the neural network for epilepsy detection. Section 4.4 examines and compares various software simulation and the hardware results. Finally, Section 4.5 draws some concluding remarks based on the discussion from the work completed in this research. A vector processor approach was chosen from the various state of the art approaches of building a complex neural network, as it is the most suited way of performing the required computations which involves massive matrices.

## 4.1 A Neural Network Model for Hardware

The literature discussed in Chapter 2 illustrates the existing state of the art technology which is used as a reference during the development of the novel neural processor proposed in this thesis. The proposed approach recognises that there is a trade off for using a bit-serial architecture instead of a bit parallel approach. The parallel architecture will complete the neural operation within a few clock cycles. However, the size of the hardware being designed will be much larger than the latter design which is small in size but needs much more time to complete the same operation. In the research specifications, size of the hardware is paramount as it is needed to be a low cost wearable device. Therefore, a compromise was made to use bit-serial architecture for the main data processing unit (DPU) but the other components of the system will use a parallel architecture.

Figure 4.2 represents the hardware implementation of the 4-3-2 network topology shown in Figure 4.1. The input layer of the 4-3-2 topology consist of 4 input neurons (X0, X1, X2, X3), 3 hidden layer neurons (H0, H1, H2) and 2 output neurons (U0 and U1). The weights, $w$ is presented along each arrow leading to the hidden layer. The hidden layer weights ($w_h$) are the arrows leading from the hidden layer to the output layer. This is one of the many examples that can be configured using the proposed approach.

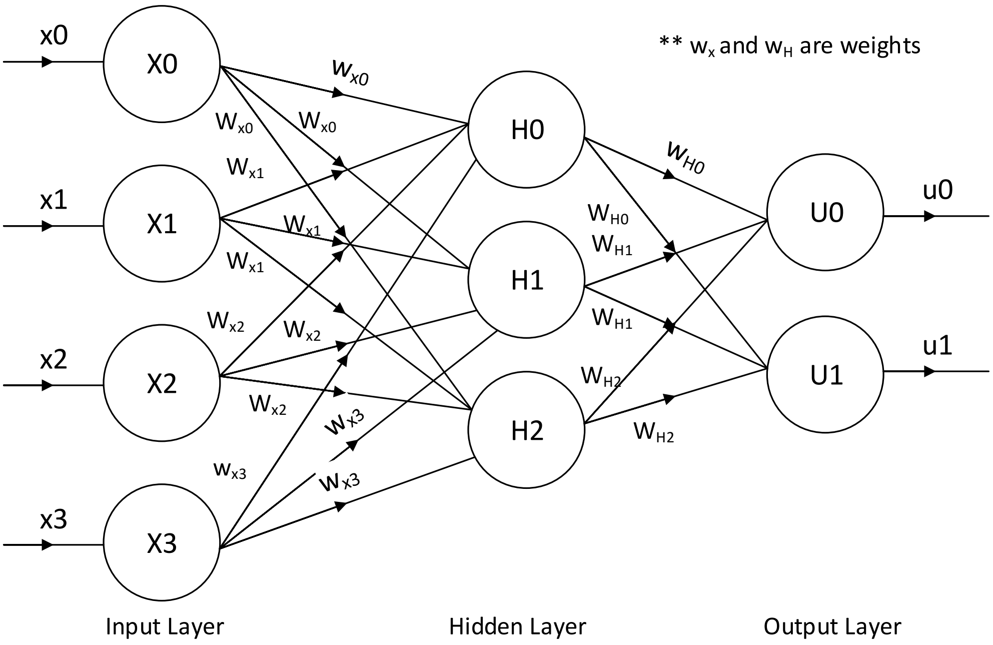


Figure 4.1: 4-3-2 Network Topology

Equation 3.1 and Figure 3.5 illustrate the neural operation of a 4-3-2 network topology when implemented in hardware (Figure 4.2). In Figure 4.2, the range of x0 to x3 indicate the inputs, w indicates the weights with u0 and u1 as separate outputs. The $u$ outputs will later be passed through an activation function to obtain the output $y$ in equation 3.2.

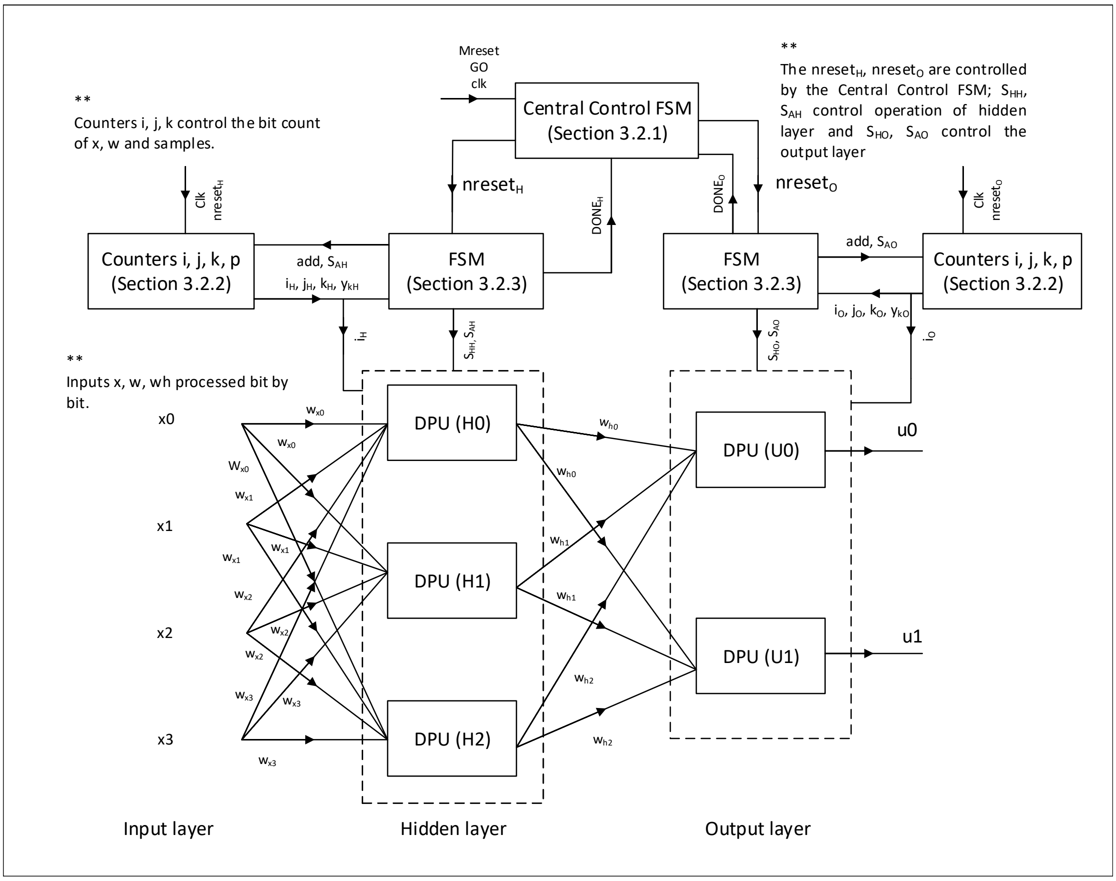

Mreset
GO
clk
Central Control FSM (Section 3.2.1)
**
Counters i, j, k control the bit count of x, w and samples.
**
The nreset$_H$, nreset$_O$ are controlled by the Central Control FSM; S$_{HH}$, S$_{AH}$ control operation of hidden layer and S$_{HO}$, S$_{AO}$ control the output layer
Clk
nreset$_H$
nreset$_H$
DONE$_H$
DONE$_O$
nreset$_O$
Clk
nreset$_O$
Counters i, j, k, p (Section 3.2.2)
add, S$_{AH}$
i$_H$, j$_H$, k$_H$, y$_{kH}$
FSM (Section 3.2.3)
FSM (Section 3.2.3)
add, S$_{AO}$
i$_O$, j$_O$, k$_O$, y$_{kO}$
Counters i, j, k, p (Section 3.2.2)
i$_H$
S$_{HH}$, S$_{AH}$
S$_{HO}$, S$_{AO}$
i$_O$
**
Inputs x, w, wh processed bit by bit.
x0
x1
x2
x3
w$_{x0}$
w$_{x1}$
w$_{x2}$
w$_{x3}$
DPU (H0)
DPU (H1)
DPU (H2)
w$_{h0}$
w$_{h1}$
w$_{h2}$
DPU (U0)
DPU (U1)
u0
u1
Input layer
Hidden layer
Output layer

Figure 4.2: 4-3-2 Network Topology in hardware

## 4.2 Proposed Approach: Novel Hardware Neural Network Implementation Design

In this section, a novel approach was proposed as an alternative method when implementing an artificial neural network (Figure 4.2). The DPU design proposed in Chapter 3 is efficient and has a very low hardware cost. With the historic bit-serial architecture, a biological neuron can be accurately modelled. This provides the basis for the bit-serial neural network (BSNN) developed in this thesis. The learning process of the proposed design is completed off-line by using a simulation software. The basic structure of this processor consists of a control path (Figure 4.3) and a data path explained in Chapter 3. The control path only requires two different FSM in order to keep the design simple and efficient. The FSM in the control path are connected with a simple nreset signal.

### 4.2.1 Central Control FSM

In order to complete the wearable device, a simple algorithm is shown in the form of a simple FSM diagram (Figure 4.4) without the need of a complex MIPs design. The

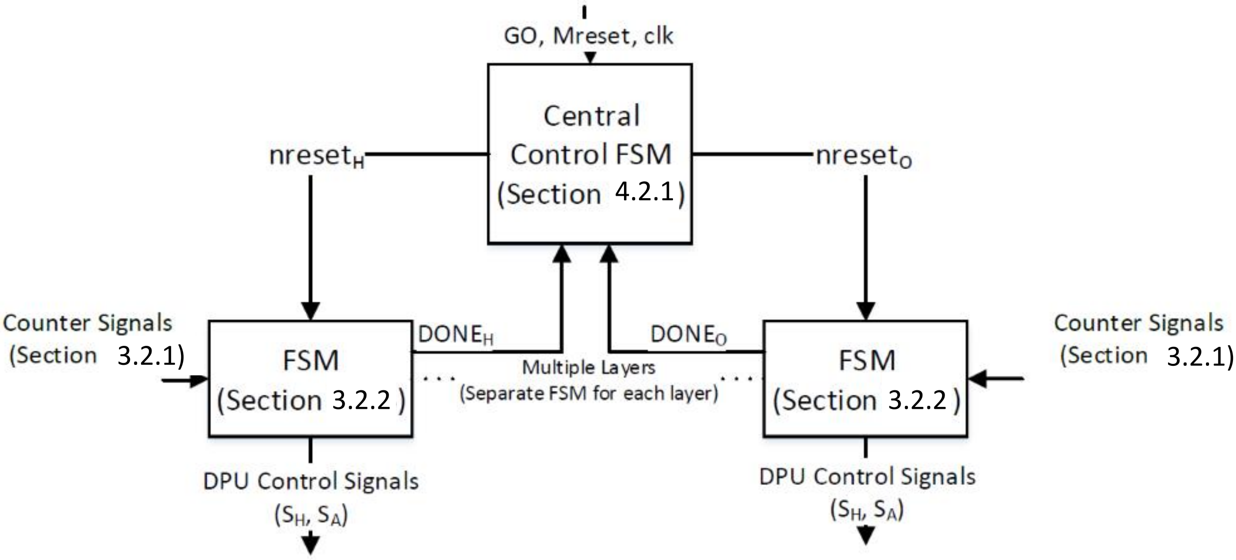

GO, Mreset, clk
Central Control FSM (Section 4.2.1)
nreset$_H$
nreset$_O$
Counter Signals (Section 3.2.1)
FSM (Section 3.2.2 )
DONE$_H$
DONE$_O$
Multiple Layers (Separate FSM for each layer)
FSM (Section 3.2.2 )
Counter Signals (Section 3.2.1)
DPU Control Signals ($S_H$, $S_A$)
DPU Control Signals ($S_H$, $S_A$)

Figure 4.3: Control Path for Vector Processor.

Mreset signal received externally will reset the whole network to default. The GO signal will begin the neural operation.

At the start of the algorithm, there will be a $READY$ state with a high $nreset_H$ and $nreset_O$. The network will proceed with the calculations in the $MUL_H$ state. In order for the hidden layer to operate, $nreset_H$ and $nreset_O$ will need to have opposing signals to prevent the output layer from performing any operation until the hidden layer has completed the calculations. When the layer has finished the calculations, a $DONE$ signal is sent from the layer FSM to the central control FSM to proceed to the next layer.

When the $DONE_H$ is received by the FSM, it proceeds to the $MUL_O$ state starting the operation on the output layer. The end of the algorithm is indicated when a $DONE_O$ is received by the central control FSM. With the power analyser tool provided in Quartus II Altera software, this component is estimated to use 14.93 mW of power.

### 4.2.2 BSNN Data Path

The full data path of the proposed BSNN is based on the proposed neural DPU which is fully explained in Chapter 3. The DPUs are used as network nodes for the BSNN. The DPU are connected as shown in Figure 4.2. The DPU design was presented in Chapter 3 and illustrated clearly in Figure 3.1. As mentioned in Chapter 3, each layer is controlled using a simple FSM. Please refer to Figure 3.5 in Chapter 3 for clarity. In

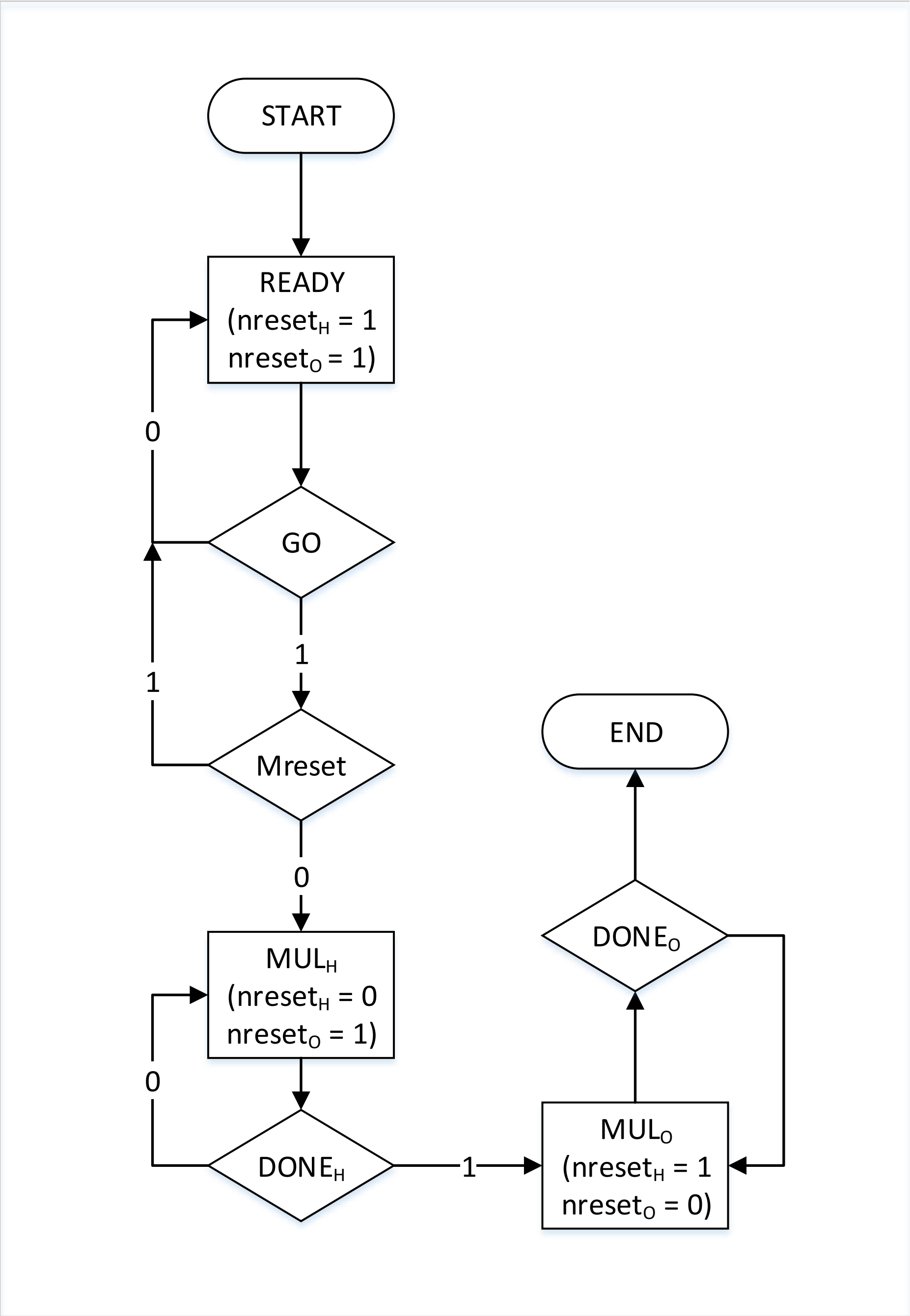


Figure 4.4: Flow chart for the central control FSM

order to fully complete the neural operation, several counters were used and this was explained in Chapter 3 Section 3.2.1.

## 4.3 Case Study: Training and Testing of BSNN For Epilepsy Detection

As described in Chapter 2, there are different types of classifier that can be used for detecting epilepsy including Naive Bayes classifier, decision tree or the k-NN classifier. There are certain advantages when using these different classifiers. However, the computational power needed for these classifiers is higher when dealing with massive datasets. An ANN solution would be more suitable when a hardware implementation is needed. Furthermore, a neural network hardware solution can perform better when dealing with multiple EEG waveforms.

Many BSNN configurations were designed and tested to confirm its full functionality. These designs could include multiple numbers of hidden neurons, as well as multiple inputs. Figure 4.5 show two different series of an EEG input plot clearly demonstrating the different amplitudes of both seizure free and seizure waveforms. The higher amplitude waveform represents a seizure event, and the free seizure event has a peak to peak amplitude of 40 uV. Each wave is plotted with 100 data points. In order to minimize the possibility of any overfitting issues, bias was introduced into the network and the number of tests during training were increased.

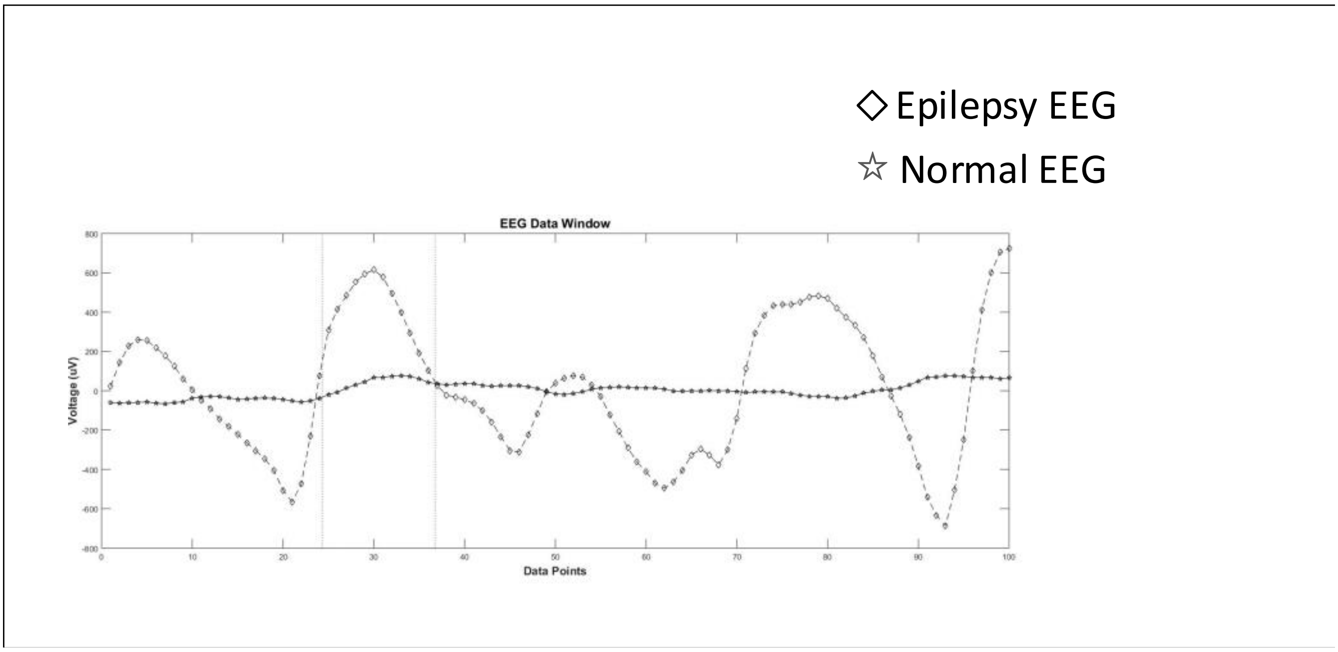


Figure 4.5: EEG data window

### 4.3.1 Epileptic seizure detection in EEG Waveform

In order to analyse the accuracy of the proposed approach, variable number of bit architecture were tested (6 bit, 12 bit). The input EEG digitized signals were taken from an on-line open source EEG database [132]. The proposed network topology (n-1-1) is trained and simulated with the MATLAB software. The weights are then extracted and used in the synthesised hardware design. With a low bit resolution, the accuracy of input waveforms is limited and further tests were made to assess the suitability of the low resolution approach. The first seizure waveform experiment involves a n-1-1 neural network ( Figure 4.6) where n can be any number of input neurons. The next part of the research explores the potential of this proposed approach in various massive parallel neural networks.

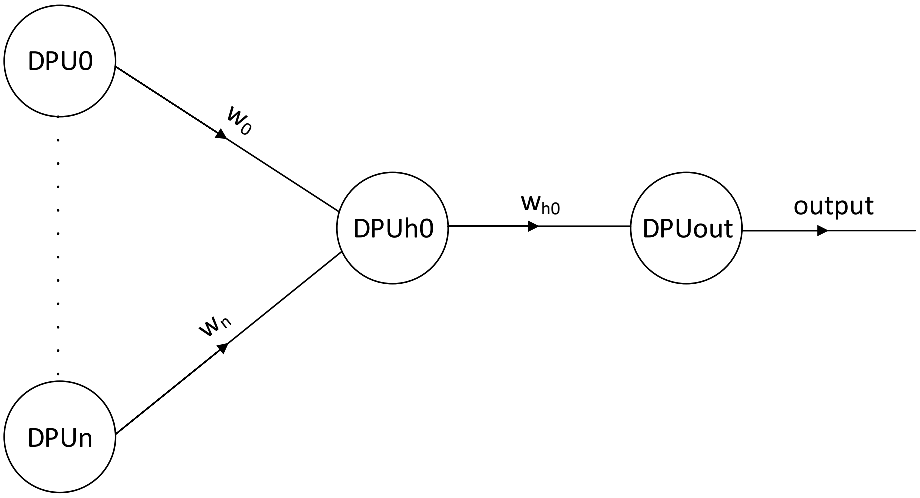
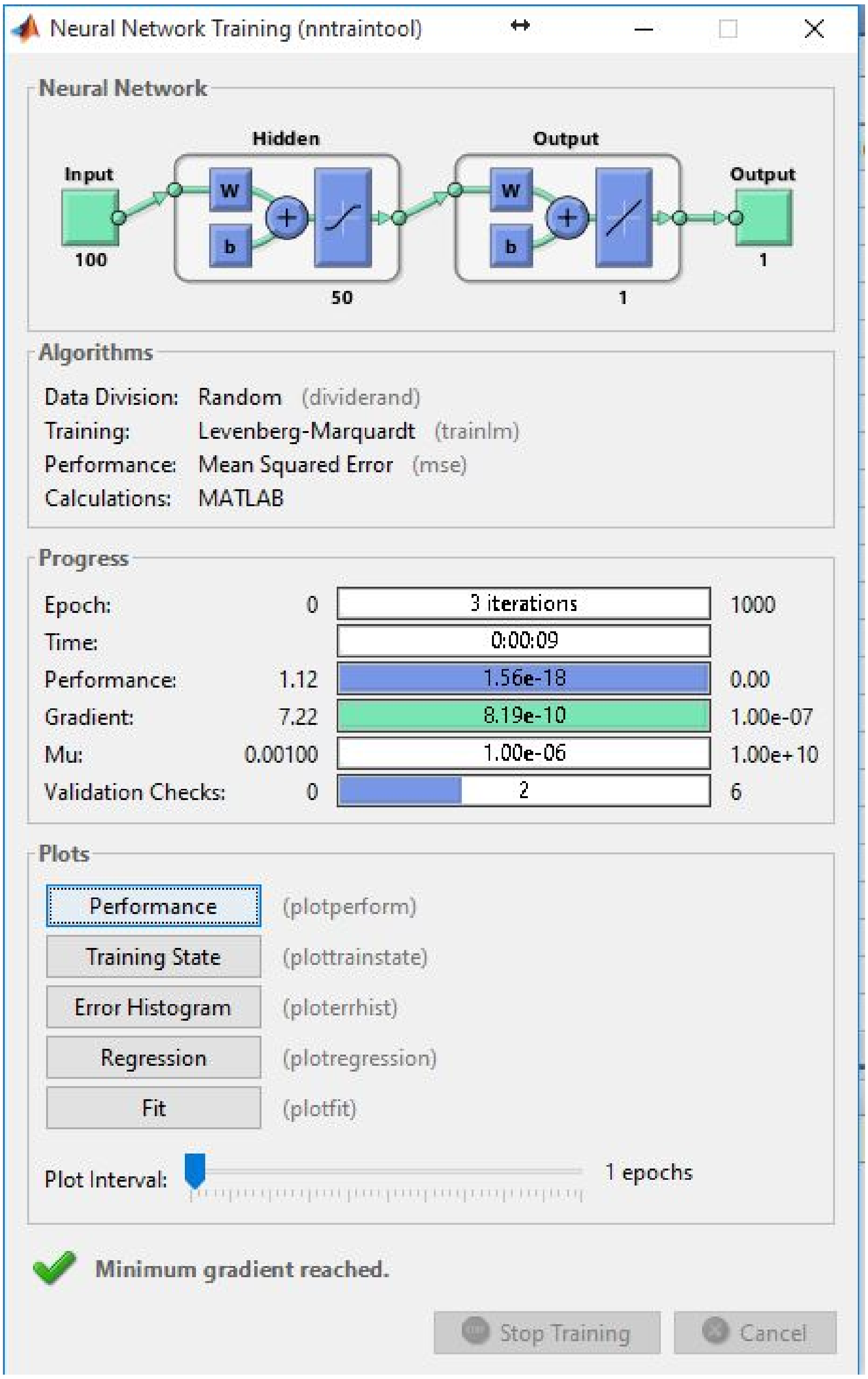


Figure 4.6: Simple Neural Network Testing (e.g. n-1-1)

Multiple numbers of inputs are used as the independent variable (10,20,30,40,50). Each independent variable are tested using 50 trials. The baseline of the output varies with the number of inputs. Any output above the threshold is a seizure event, and any output below the threshold is a seizure free event.

### 4.3.2 Network Architecture Development

With the simulation software, different network designs can be trained and tested easily. MATLAB provides the neural network toolbox that enable the hardware neural networks to be trained more effectively. Figure 4.7 presents a training tool with different training functions. The functions include checking the network performance which assess the mean square error (mse) of the simulated network. Different training functions can be used to train the neural network in the MATLAB software, such as the Levenberg-Marquardt backpropagation (trainlm), BFGS Quasi-Newton (trainbfg), Resilient Backpropagation (trainrb), Scaled Conjugate Gradient (trainscg) and Conjugate Gradient with Powell/Beale Restarts (traincgb). It has been stated that the Levenberg-Marquardt backpropagation function is the fastest training function for most applications. Furthermore, the toolbox gives a clear view of the network design. With the default setting, the training uses 70% of the datasets being tested; 15% are used for validation and the last 15% for testing the network.


Neural Network Training (nntraintool)
Neural Network
Hidden
Output
Input
W
b
100
50
Output
1
1

Algorithms

Data Division: Random (dividerand)
Training: Levenberg-Marquardt (trainlm)
Performance: Mean Squared Error (mse)
Calculations: MATLAB

Progress

| | | | |
|---|---|---|---|
| Epoch: | 0 | 3 iterations | 1000 |
| Time: | | 0:00:09 | |
| Performance: | 1.12 | 1.56e-18 | 0.00 |
| Gradient: | 7.22 | 8.19e-10 | 1.00e-07 |
| Mu: | 0.00100 | 1.00e-06 | 1.00e+10 |
| Validation Checks: | 0 | 2 | 6 |

Plots

Performance (plotperform)
Training State (plottrainstate)
Error Histogram (ploterrhist)
Regression (plotregression)
Fit (plotfit)

Plot Interval: 1 epochs

✓ Minimum gradient reached.

Figure 4.7: The training of the neural network provided in MATLAB

The Levenberg-Marquardt backpropagation training function (LM function) is accurate for linear approximate problems. The LM training function is at least four times faster than the BFG training algorithm. This was tested using a simple sine wave problem provided by open source MATLAB documentation. When dealing with less than hundred weights and result approximation needs to be very accurate, the LM function was proven to be the best choice. Thus, it is suitable for the research when implementing the bit-serial approach effectively. In conclusion, the choice of training algorithms will depend on the specific application.

BFG and CGB can perform better with pattern recognition problems. These problems include the recognition of huge datasets patterns. When the weights and EEG data reach the limit of the capability of the LM training, other types of training functions should be considered. It would be advantageous to have a comparison of the different types of training functions when testing the EEG datasets.

The trainlm function has been proven to be the most successful function in terms of performance by conducting the training process on MATLAB using different training functions. The performance for these networks is recorded: 0.0801 for BFGS Quasi-Newton, 0.1654 for Resilient Backpropagation and 0.0059 for Levenberg-Marquardt algorithm. Figures (4.8, 4.9,4.10) shows clearly that the Levenberg-Marquardt algorithm takes the least amount of time to obtain the best performance (6 epochs / iterations). It has also the least mean square error (mse) giving the best performance for the network.

### 4.3.3 Neural Network Design Validation and Testing

The research input data was obtained from an on-line open source [133]. The online dataset were obtained from different scenarios during an EEG scan. The tests used a combination of SET C (100 EEG waveforms with seizure free instances) and SET E (100 EEG waveforms during seizures). In the form of a controlled experiment, the datasets are taken from the brain (epileptogenic zone) of the same patient.

#### 4.3.3.1 Network Validation

Using MATLAB results as a comparison, Table 4.1, 4.2, 4.3, 4.4 illustrates the correct recognition of the software implementation and the bit-serial neural network hardware. The EEG waveform used in these tests are the same dataset used in the training process as a method of validation. These tests validates the feasibility the designs which were also tested using various bit precision. Two different threshold approaches were used here: mean and median. By increasing the number of tests in the training process, the accuracy of the network increased accordingly. Each trial is a single EEG waveform which contains a certain number of inputs. The figures below (Figure 4.11, 4.12, 4.13, 4.14 and 4.15) illustrate the neural output of the network for each trial in the form of spikes. Any neural output point (circle), above the border (red line in the Figure 4.11, 4.12, 4.13, 4.14, 4.15) is considered as a seizure event; below the border are the seizure free events.

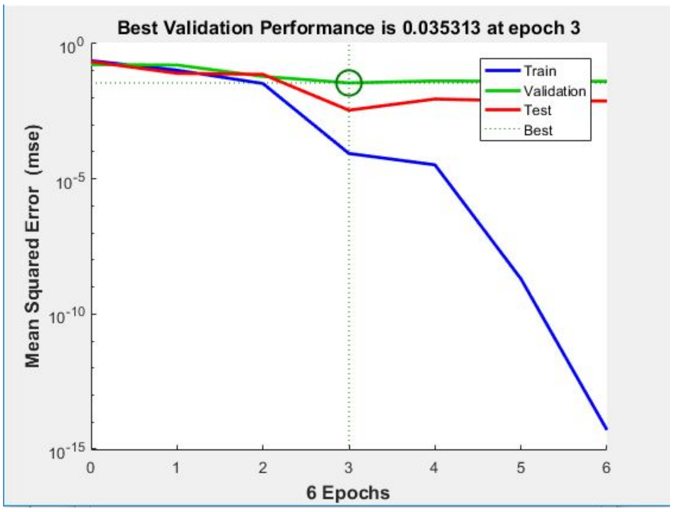


Figure 4.8: (a) training process in MATLAB (Levenberg-Marquardt algorithm)

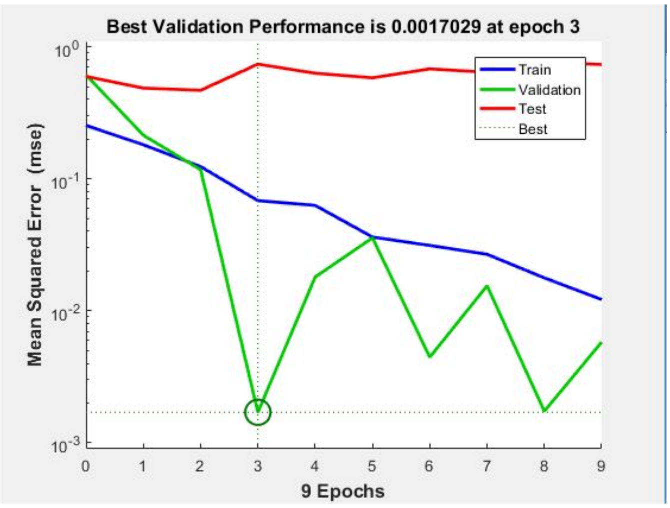


Figure 4.9: (b) training process in MATLAB (BFGS Quasi-Newton algorithm)

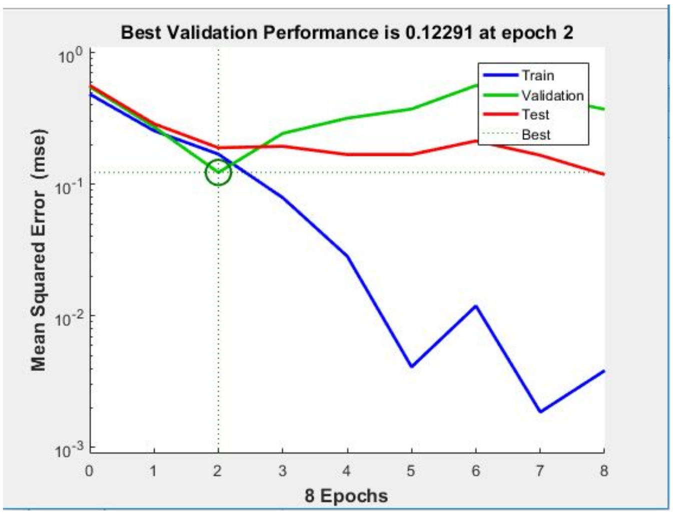


Figure 4.10: (c) training process in MATLAB (Resilient Backpropagation algorithm)

| **No. of inputs** | **Correct Recognition (Software / Hardware)** |
|---|---|
| 10 | 49 / 45 |
| 20 | 48 / 47 |
| 30 | 47 / 48 |
| 40 | 49 / 50 |
| 50 | 46 / 50 |

Table 4.1: Correct Recognition of different inputs for Bit-Serial Vector Processor Using Mean (n-1-1 network,12 bit precision))

| **No. of inputs** | **Correct Responses Out of 50 Trials (Software / Hardware)** |
|---|---|
| 10 | 49 / 45 |
| 20 | 48 / 48 |
| 30 | 47 / 48 |
| 40 | 49 / 50 |
| 50 | 46 / 50 |

Table 4.2: Correct Recognition of different inputs for Bit-Serial Vector Processor Using Median (n-1-1 network, 12 bit precision)

| **No. of inputs** | **Correct Responses Out of 50 Trials (Software / Hardware)** |
|---|---|
| 10 | 49 / 19 |
| 20 | 48 / 20 |
| 30 | 47 / 33 |
| 40 | 49 / 28 |
| 50 | 46 / 34 |

Table 4.3: Correct Recognition of different inputs for Bit-Serial Vector Processor Using Mean(n-1-1 network, 6 bit precision)

| **No. of inputs** | **Correct Responses Out of 50 Trials (Software / Hardware)** |
|---|---|
| 10 | 49 / 18 |
| 20 | 48 / 24 |
| 30 | 47 / 32 |
| 40 | 49 / 32 |
| 50 | 46 / 33 |

Table 4.4: Correct Recognition of different inputs for Bit-Serial Vector Processor Using Median(n-1-1 network, 6 bit precision)

Figure 4.11 presents the tests of a 10-1-1 network; Figure 4.12 a 20-1-1 network; Figure 4.13 a 30-1-1 network; Figure 4.14 a 40-1-1 network and Figure 4.15 a 50-1-1 network. The networks were tested with an increase of inputs but the same number of neurons in the hidden layer. These validation experiments attempt to increase the accuracy before conducting further tests using additional data.

From the results, the 40-1-1 network design is one of the basis of our optimal BSNN

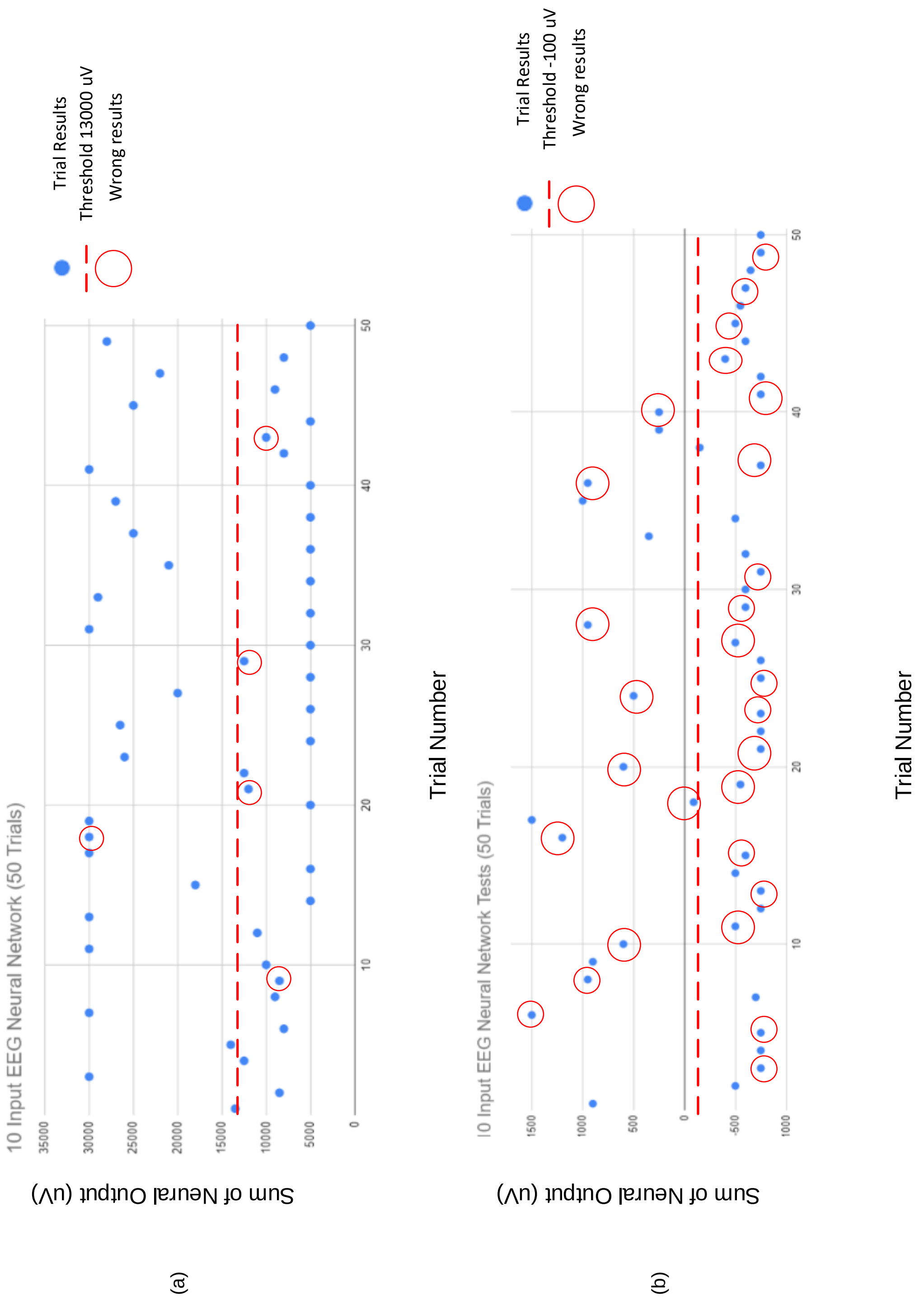
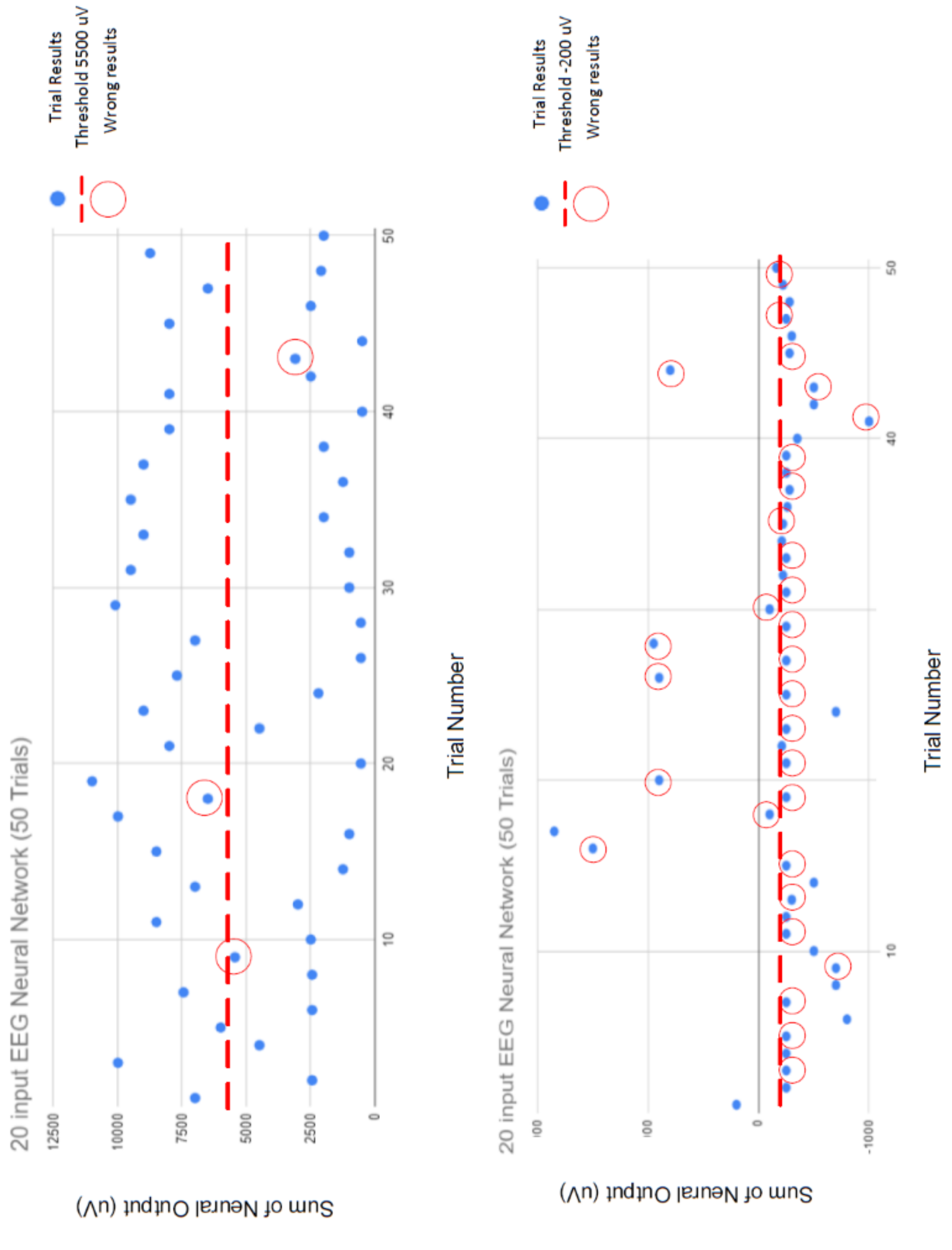


Figure 4.11: 10 Input EEG Neural Network Test ((a) 12 bit representation, (b) 6 bit representation)



(a)

(b)

Figure 4.12: 20 Input EEG Neural Network Test ((a) 12 bit representation, (b) 6 bit representation)

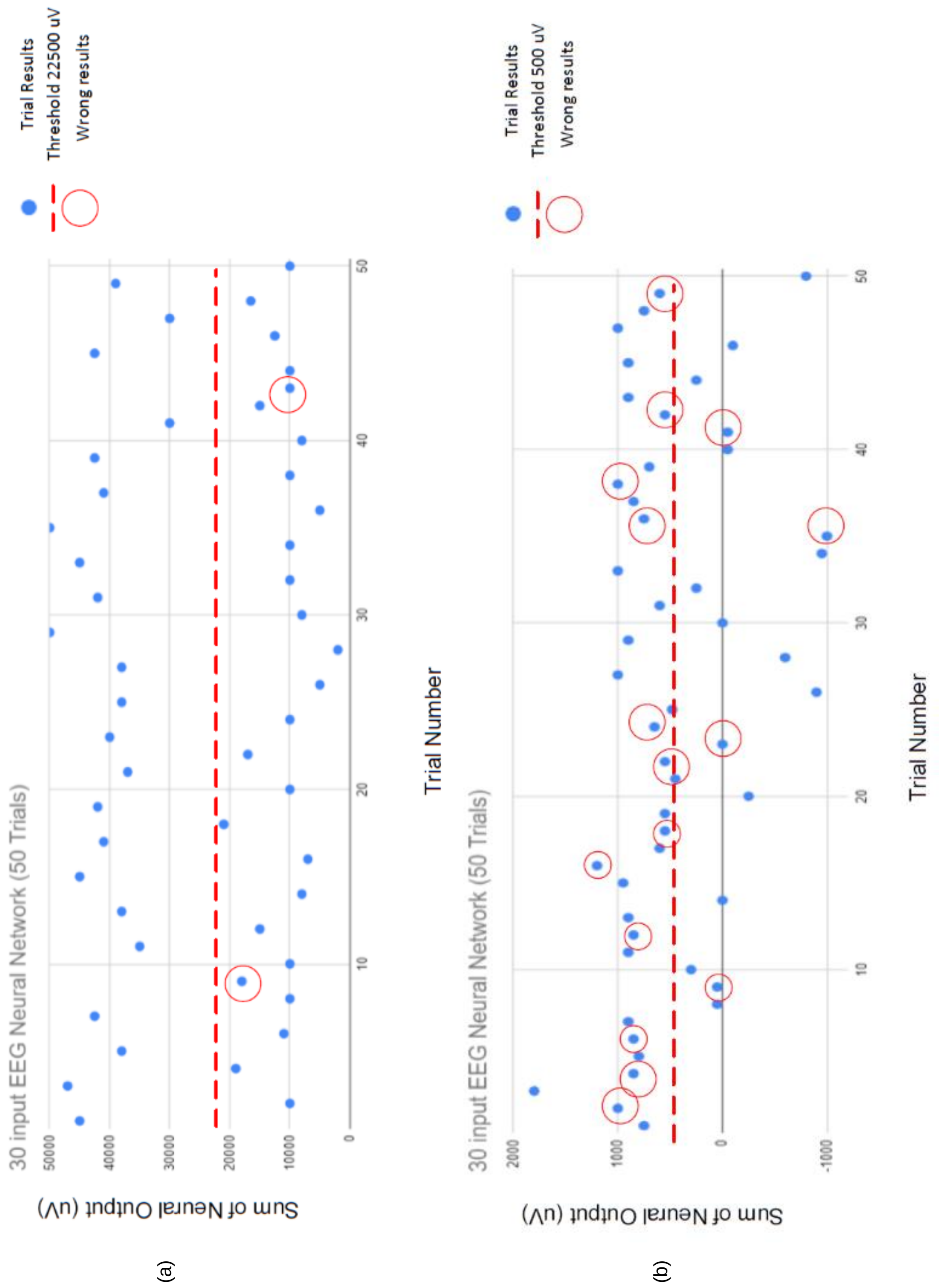


Figure 4.13: 30 Input EEG Neural Network Test ((a) 12 bit representation, (b) 6 bit representation)

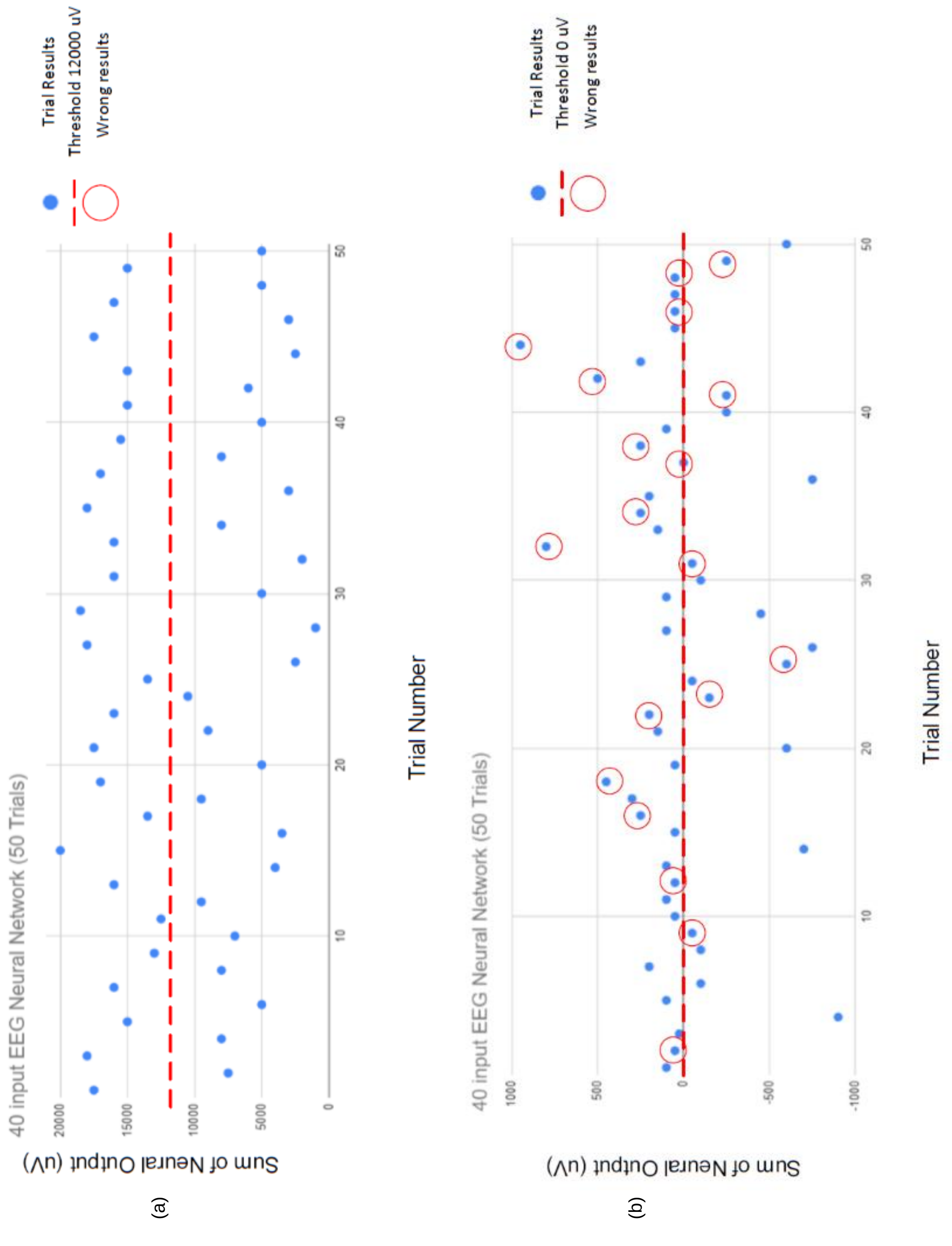


Figure 4.14: 40 Input EEG Neural Network Test ((a) 12 bit representation, (b) 6 bit representation)

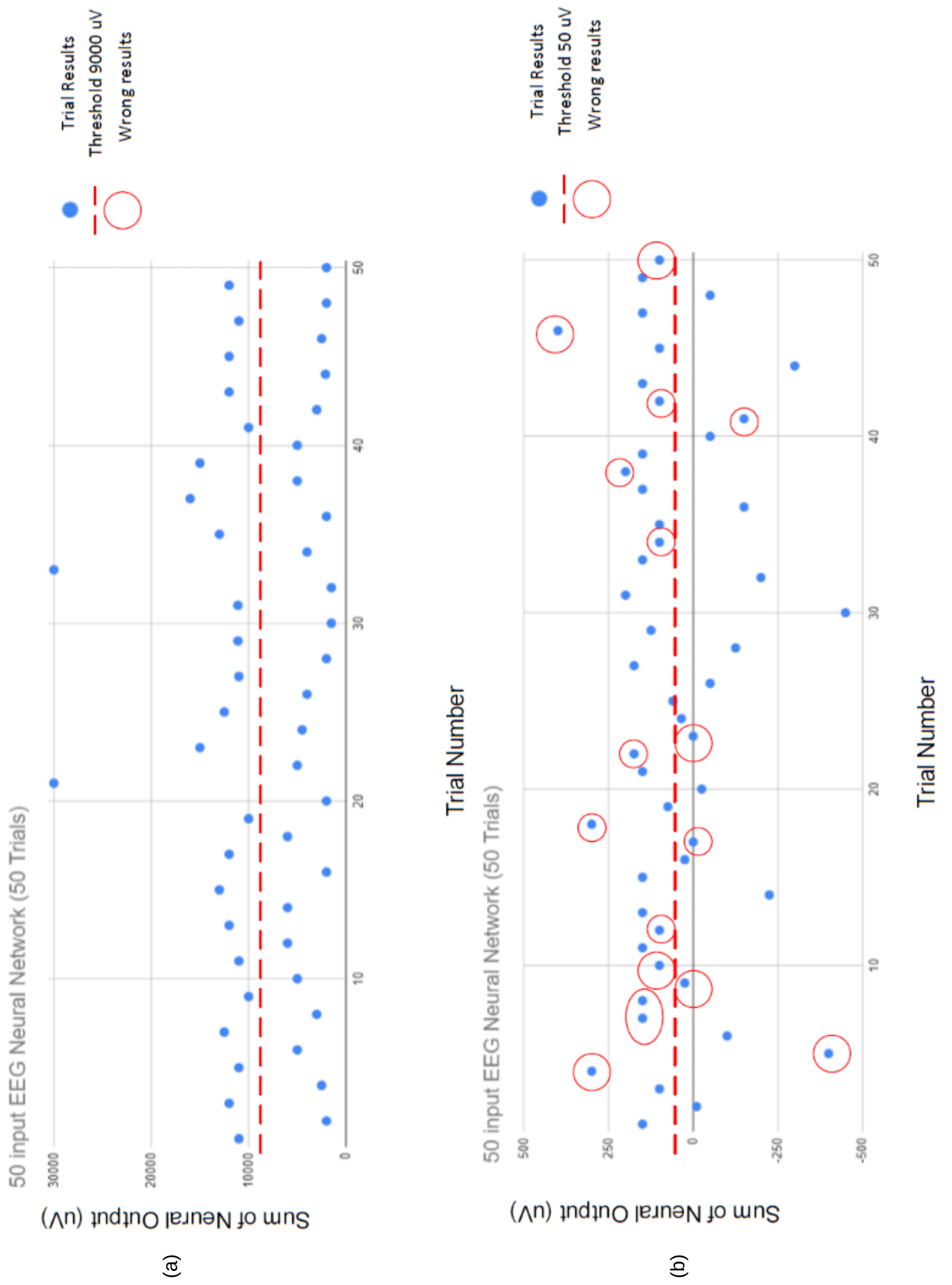


Figure 4.15: 50 Input EEG Neural Network Test ((a) 12 bit representation, (b) 6 bit representation)

design. In this validation process, the tests for epilepsy detection yielded an average validation accuracy of 94% when using a 12 bit architecture. A lower bit resolution (6 bit) have also been tested which yielded a validation accuracy of 54%. However, the time consumed for each operation is reduced by half. Thus, the 12 bit architecture will be used in the testing stage with various configurations to find the optimal network for epilepsy detection.

With the correct recognition results, it is possible to evaluate the potential of the network with several different evaluation metrics such as sensitivity, specificity and precision/-positive predictive value (PPV). The equations and definition for these metrics are listed below.

1. Sensitivity:
$$TPR = TP/P = TP/(TP + FN) \tag{4.1}$$

2. Specificity
$$SPC = TN/N = TN/(TN + FP) \tag{4.2}$$

3. Precision/PPV
$$PPV = TP/(TP + FP) \tag{4.3}$$

4. Negative Predictive Value/ NPV
$$NPV = TN/(TN + FN) \tag{4.4}$$

Where:

1. $P$ = Positive Sample: Epileptic Patient EEG
2. $N$ = Negative Sample: Normal Patient EEG
3. $TP$ = True Positive: Correctly Identified Epileptic EEG
4. $TN$ = True Negative: Correctly Identified Normal EEG
5. $FP$ = False Positive: Incorrectly Identified Epileptic EEG
6. $FN$ = False Negative: Incorrectly Identified Normal EEG

Table 4.5 demonstrates the suitability of using a single neuron to detect epilepsy. Using a 40 input design, it is possible to achieve the maximum accuracy of detecting the epileptic spikes. Table 4.6 illustrates the possibility of using a single neuron in epilepsy detection. The detailed comparison between the two architecture is illustrated clearly in Figure 4.16.

| No. of Inputs | Sensitivity | Specificity | PPV | NPV |
|---|---|---|---|---|
| 10 | 95.4% | 85.7% | 84% | 96% |
| 20 | 95.9% | 94.1% | 94% | 96% |
| 30 | 95.8% | 92.3% | 92% | 96% |
| 40 | 100% | 100% | 100% | 100% |
| 50 | 95.8% | 92.3% | 92% | 96% |

Table 4.5: Evaluation of different number input for single neuron design with 12 bit architecture

| No. of Inputs | Sensitivity | Specificity | PPV | NPV |
|---|---|---|---|---|
| 10 | 33.3% | 42.8% | 20% | 50% |
| 20 | 33.9% | 57.2% | 41.5% | 56% |
| 30 | 60% | 64% | 62.5% | 62% |
| 40 | 53.3% | 59% | 62.5% | 50% |
| 50 | 66.7% | 73.9% | 75% | 65.4% |

Table 4.6: Evaluation of different number input for single neuron design with 6 bit architecture

#### 4.3.3.2 Network Testing

The hardware design was tested using other EEG waveforms. It was found that the n-1-1 network configuration has a very bad recognition rate when compared with the simulation results obtained from MATLAB. A detailed comparison between the validation and testing stage of the design process is illustrated in Figure 4.17. The parameters used in the comparison are the evaluation metrics described above.

With the result analysis, it can be concluded that a a multi input single neuron is not sufficient to detect epilepsy accurately. Therefore, different configurations of network has been tested. The network configuration used is a 40-n-1 network with n number of hidden neurons. The architecture used in these network has a 12 bit precision to obtain better accuracy. Table 4.7 presents the response of the 40-n-1 network using MATLAB results as a form of comparison.

| No. of hidden neurons | Correct Responses Out of 50 Trials (Software / Hardware) |
|---|---|
| 1 | 49 / 5 |
| 10 | 49 / 31 |
| 20 | 48 / 26 |
| 30 | 47 / 29 |
| 40 | 49 / 26 |

Table 4.7: Correct Recognition of different inputs for Bit-Serial Vector Processor Using Mean(40-n-1 network, 12 bit precision)

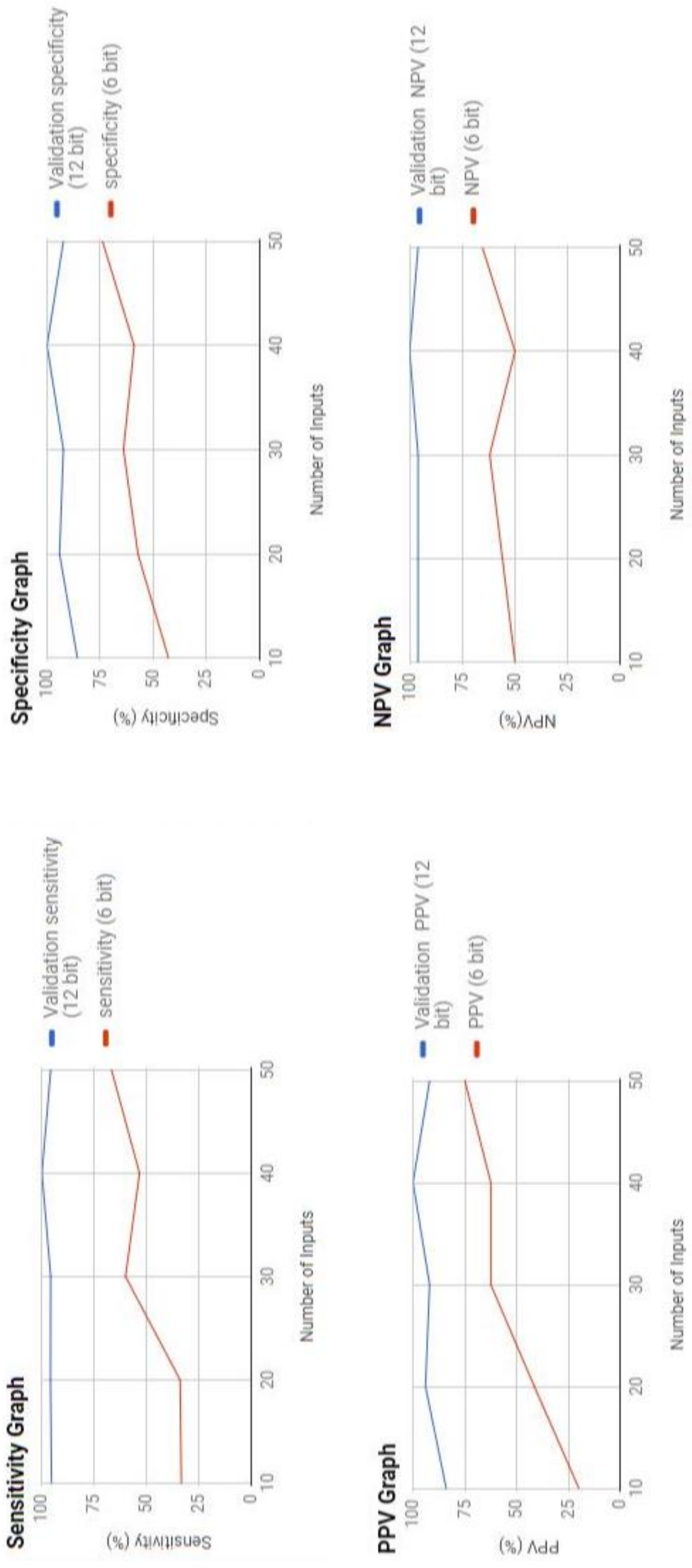


Figure 4.16: Various Evaluation Metrics

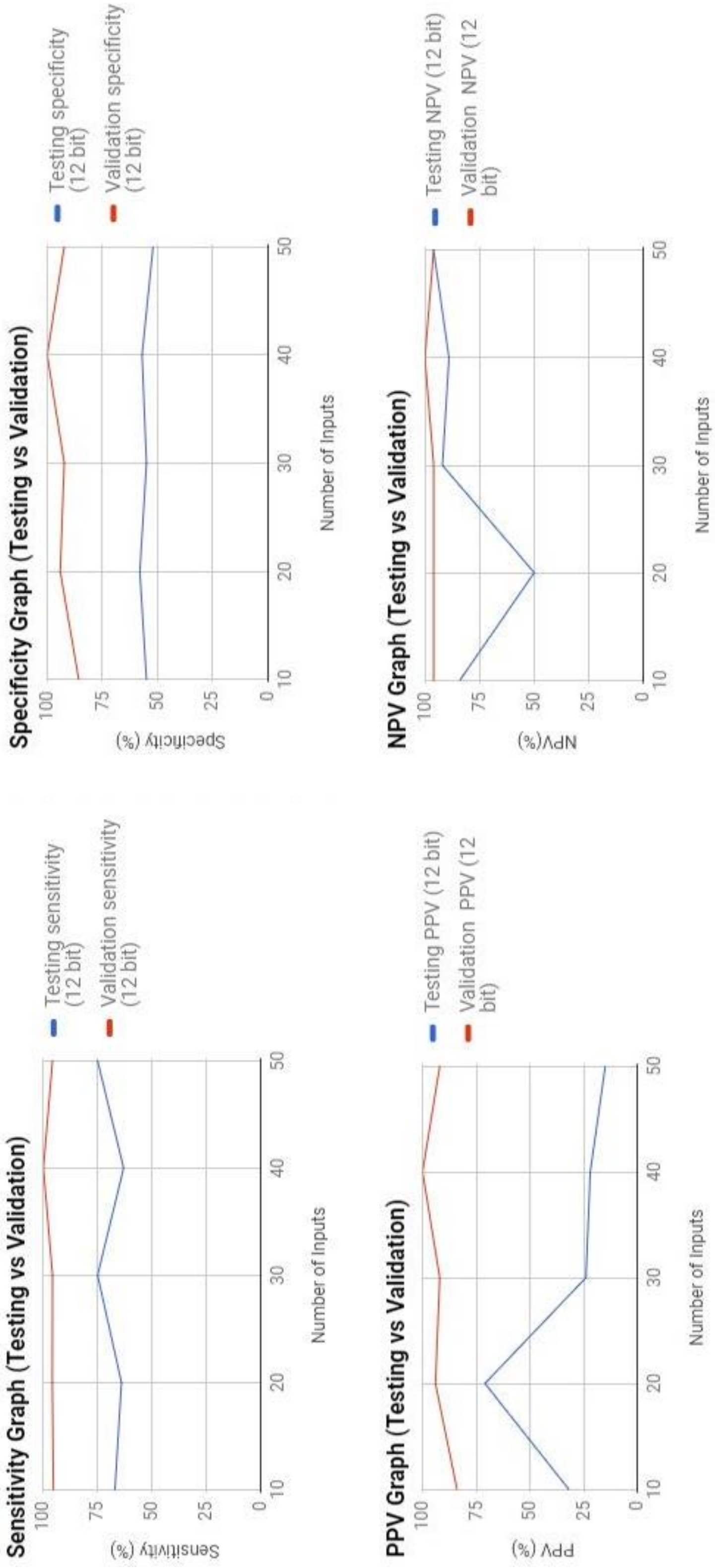


Figure 4.17: Various Evaluation Metrics

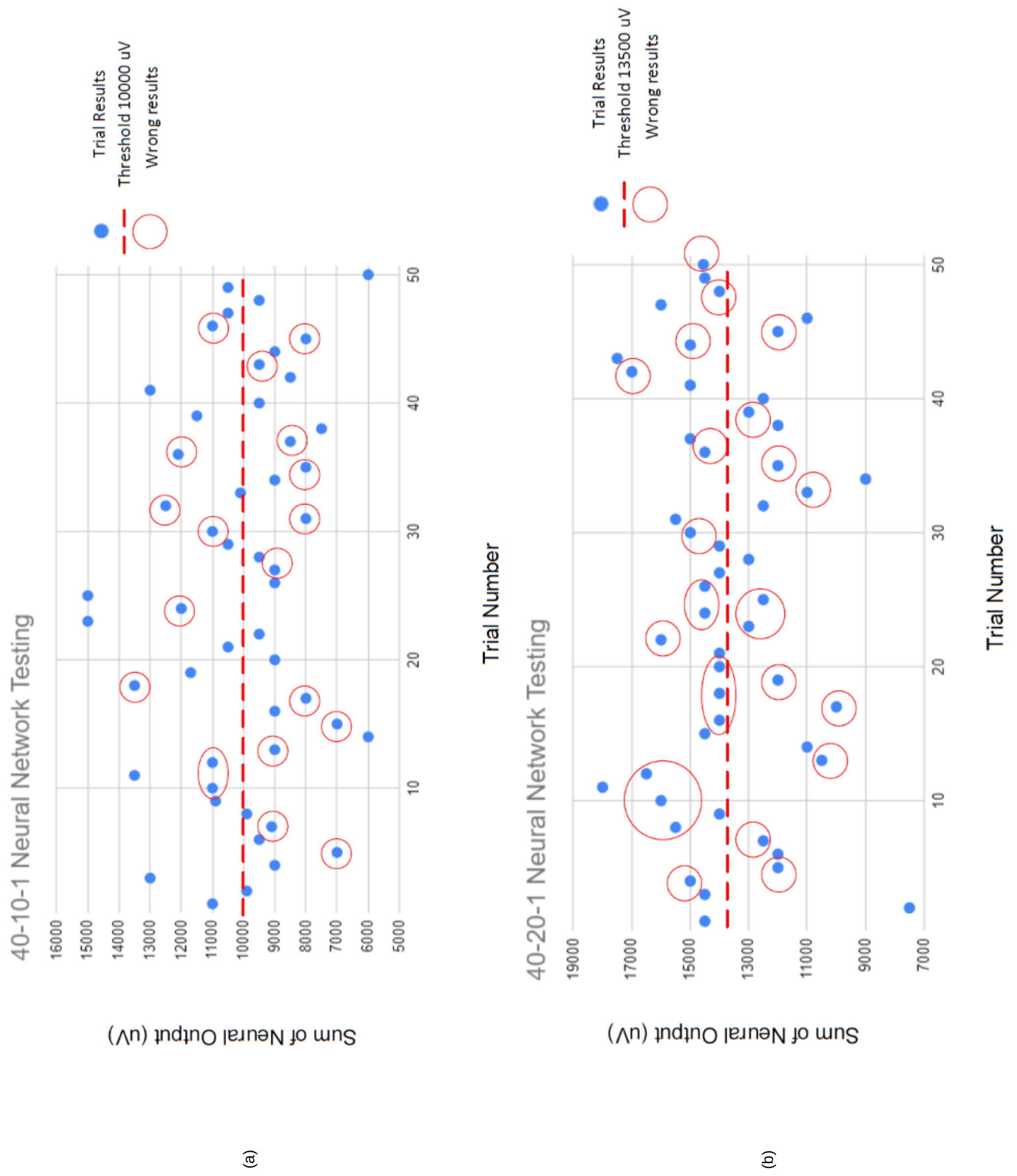


Figure 4.18: (a) 40-10-1 network configuration (b) 40-20-1 network configuration

The output results for these different configurations is included here in Figure 4.18, 4.19. The evaluation metrics for these configuration are analysed in Table 4.8 and illustrated with a column chart (Figure 4.20).

In summary, the network configuration of 40-30-1 provides some promising results at detecting epileptic waveforms. Further tests were conducted using a larger number of inputs and additional data in the next chapter. This allows an optimal network configuration for epilepsy detection to be obtained.

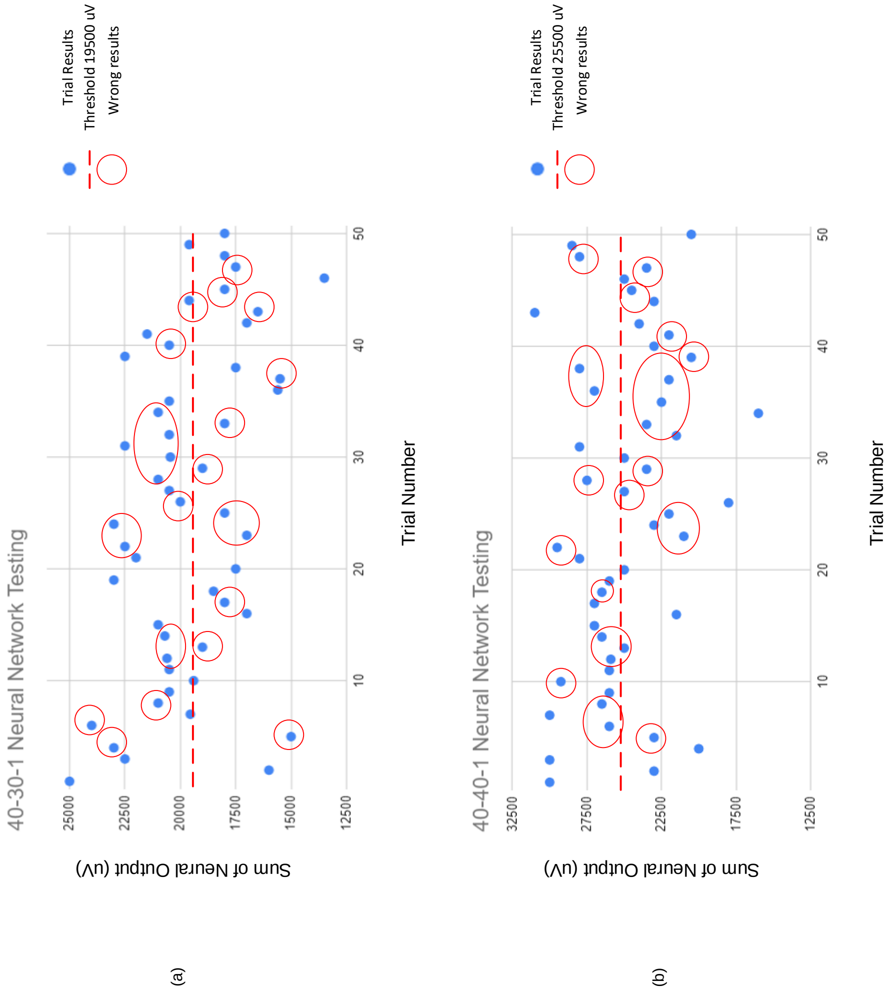


Figure 4.19: (a) 40-30-1 network configuration (b) 40-40-1 network configuration

| No. of hidden neurons | Sensitivity | Specificity | PPV | NPV |
|---|---|---|---|---|
| 10 | 63.6% | 60.7% | 56% | 68% |
| 20 | 51.7% | 52.3% | 60% | 44% |
| 30 | 55.5% | 56.5% | 60% | 52% |
| 40 | 50% | 50% | 48% | 52% |

Table 4.8: Evaluation of different number of hidden neuron network design (40 inputs, 12 bit precision)

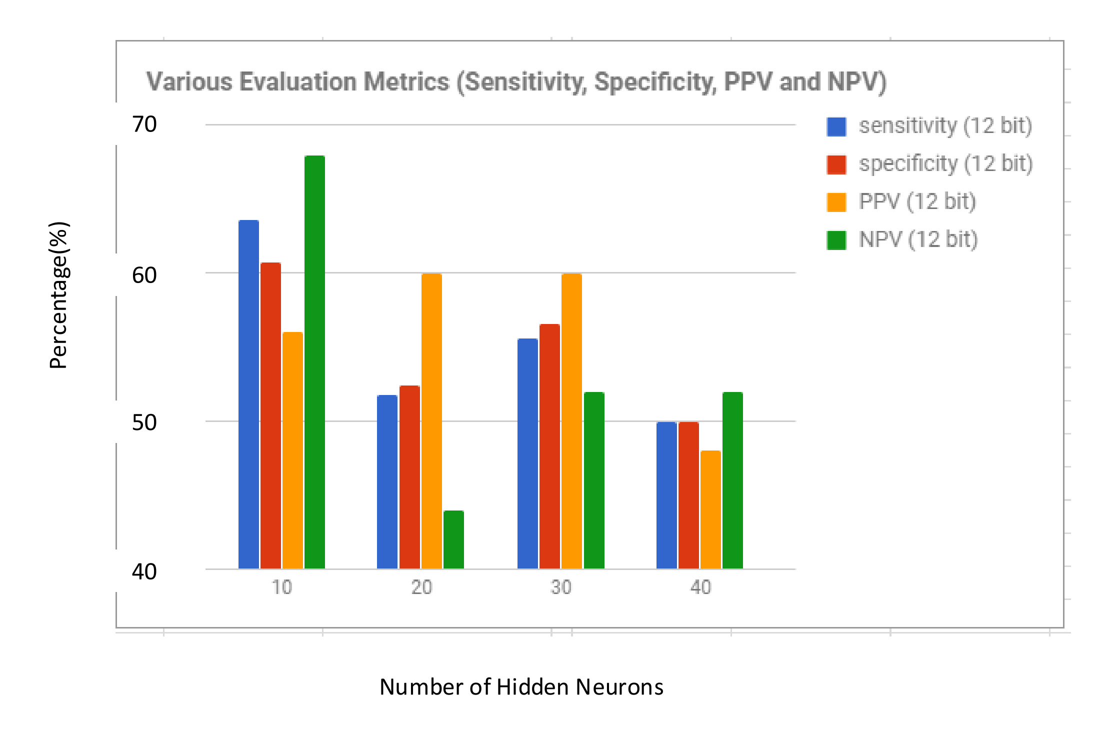


Figure 4.20: Evaluation Metrics

## 4.4 Discussion and Comparison

As mentioned in Chapter 3, there are two main techniques when implementing a neural processor, i.e. analog or digital. As the analog method is less accurate and inflexible, the digital approach has been used to develop the DPU. The digital vector processor approach is used to develop the needed ANN as it is simple and have the required accuracy to tackle epilepsy detection.

### 4.4.1 Evaluation of MATLAB Results

The software simulation of the different neural networks are proven to be more accurate with an error margin of less than 5%. Thus, the MATLAB is suitable to be used for the training process.

The training of the proposed neural network conducted using MATLAB acts as a benchmark for the hardware testing. Three different training algorithms have been used to train the data and it was found that the Levenberg-Marquardt backpropagation function is the best training algorithm to be used for this application with a mse of 0.081. As part of the training process, bias values and additional training data have been used to minimize the chances of overfitting.

With the training data, the accuracy of the network using MATLAB is 100% and can still maintain a minimum correct recognition rate of 90% when tested with additional data. This shows that the proposed neural network is viable for epilepsy detection.

### 4.4.2 Evaluation and discussion of hardware results

The synthesised hardware can be implemented on a Altera Cyclone FPGA V board with the extracted weights from the simulation software. The inputs and weights are first truncated before the experiments. The research also presents some comparison of the hardware cost when different number of inputs are used for a single neuron. These comparisons were made using different bit precision, i.e. 6, 8, 12, 16 bits as shown in Table 4.9, 4.10, 4.11, 4.12. The FPGAs used for comparison are Cyclone IV, Cyclone V and Stratix IV. From the results in the tables, it is concluded that Cyclone V is still the best option when implementing the BSNN . Figure 4.21 illustrates how the cost increases when different number of inputs are used when implemented on an Altera Cyclone FPGA V board. The figure clearly illustrates the results of different bit precision.

| Number of Inputs | | Logic Element Cost | |
|---|---|---|---|
| | Cyclone IV | Cyclone V | Stratix IV |
| 10 | 113 | 96 | 97 |
| 20 | 116 | 103 | 103 |
| 30 | 131 | 103 | 96 |
| 40 | 138 | 107 | 110 |
| 80 | 148 | 107 | 106 |
| 100 | 148 | 109 | 106 |

Table 4.9: Logic elements needed for a single neuron with different number of inputs (6 bit precision)

| Number of Inputs | | Logic Element Cost | |
|---|---|---|---|
| | Cyclone IV | Cyclone V | Stratix IV |
| 10 | 139 | 104 | 106 |
| 20 | 146 | 105 | 101 |
| 30 | 159 | 110 | 106 |
| 40 | 159 | 113 | 113 |
| 80 | 188 | 120 | 119 |
| 100 | 197 | 128 | 128 |

Table 4.10: Logic elements needed for a single neuron with different number of inputs (8 bit precision)

Next, comparison is also made for BSNN with various number of hidden neurons. It should be noted that the comparisons do not include the feature extractor that is proposed in the next chapter. Table 4.13, 4.14, 4.15, 4.16 presents the cost of a 40-n-1 network where n is the number of hidden neurons used. This comparison is also made

| **Number of Inputs** | | **Logic Element Cost** | |
|---|---|---|---|
| | **Cyclone IV** | **Cyclone V** | **Stratix IV** |
| 10 | 188 | 133 | 133 |
| 20 | 201 | 140 | 140 |
| 30 | 206 | 144 | 144 |
| 40 | 219 | 155 | 156 |
| 80 | 253 | 187 | 180 |
| 100 | 270 | 195 | 194 |

Table 4.11: Logic elements needed for a single neuron with different number of inputs (12 bit precision)

| **Number of Inputs** | | **Logic Element Cost** | |
|---|---|---|---|
| | **Cyclone IV** | **Cyclone V** | **Stratix IV** |
| 10 | 231 | 157 | 156 |
| 20 | 243 | 168 | 168 |
| 30 | 250 | 168 | 168 |
| 40 | 260 | 182 | 188 |
| 80 | 295 | 215 | 216 |
| 100 | 315 | 206 | 226 |

Table 4.12: Logic elements needed for a single neuron with different number of inputs (16 bit precision)

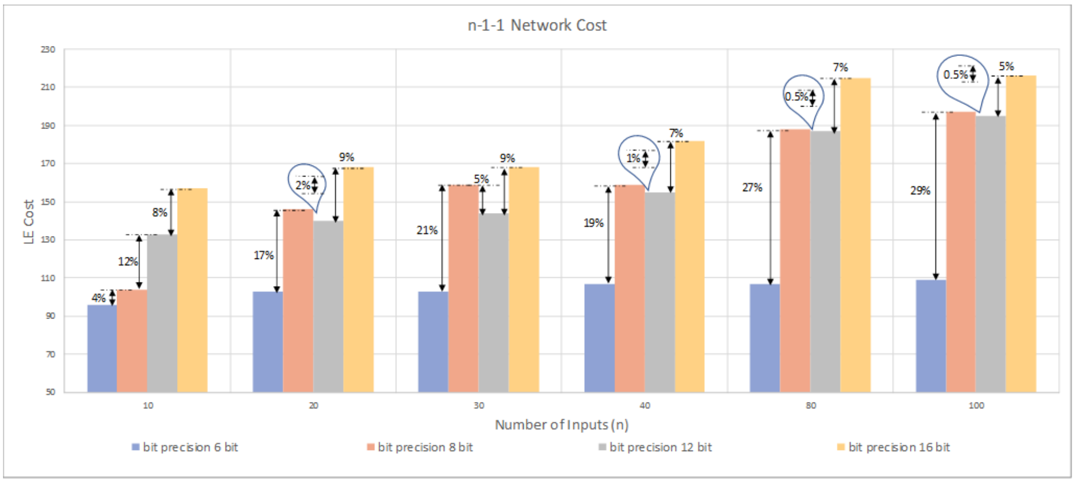


Figure 4.21: n-1-1 Network Cost Comparison

using different bit precision. Figure 4.22 illustrates the different hardware cost needed when implementing the network on a Cyclone V FPGA. The figure clearly shows the increase in LE cost when different bit precision is used. 40 inputs were chosen for this comparison as this network shows some promising results when experiments were conducted. Different features are used as network inputs to find the optimal network configuration in Chapter 5.

| **Bit Precision** | | **Logic Element Cost** | |
|---|---|---|---|
| | **Cyclone IV** | **Cyclone V** | **Stratix IV** |
| 6 | 803 | 381 | 363 |
| 8 | 1172 | 512 | 512 |
| 12 | 1378 | 657 | 674 |
| 16 | 1598 | 811 | 796 |

Table 4.13: Logic elements needed for a 40-10-1 network with different bit architecture

| **Bit Precision** | | **Logic Element Cost** | |
|---|---|---|---|
| | **Cyclone IV** | **Cyclone V** | **Stratix IV** |
| 6 | 1323 | 536 | 536 |
| 8 | 1885 | 748 | 748 |
| 12 | 2577 | 1104 | 1126 |
| 16 | 2866 | 1134 | 1164 |

Table 4.14: Logic elements needed for a 40-20-1 network with different bit architecture

| **Bit Precision** | | **Logic Element Cost** | |
|---|---|---|---|
| | **Cyclone IV** | **Cyclone V** | **Stratix IV** |
| 6 | 1835 | 709 | 712 |
| 8 | 2681 | 911 | 1016 |
| 12 | 3527 | 1383 | 1385 |
| 16 | 4297 | 1678 | 1679 |

Table 4.15: Logic elements needed for a 40-30-1 network with different bit architecture

| **Bit Precision** | | **Logic Element Cost** | |
|---|---|---|---|
| | **Cyclone IV** | **Cyclone V** | **Stratix IV** |
| 6 | 2364 | 847 | 853 |
| 8 | 3417 | 1147 | 1147 |
| 12 | 5657 | 2763 | 2786 |
| 16 | 6541 | 3240 | 3084 |

Table 4.16: Logic elements needed for a 40-40-1 network with different bit architecture

Some evaluation have been added to this subsection. It is clearly interesting to see how bit-precision affects the hardware cost across different network configuration and it is apparent there is an exponential increase in hardware cost when hidden neurons were increased from 30 to 40. This caused an increase of 41% in logic elements (LE). It is also quite clear when analysing Table 4.13 to 4.16, Cyclone IV requires more logic elements to synthesise the same network compared to Cyclone V and Stratix IV indicating that

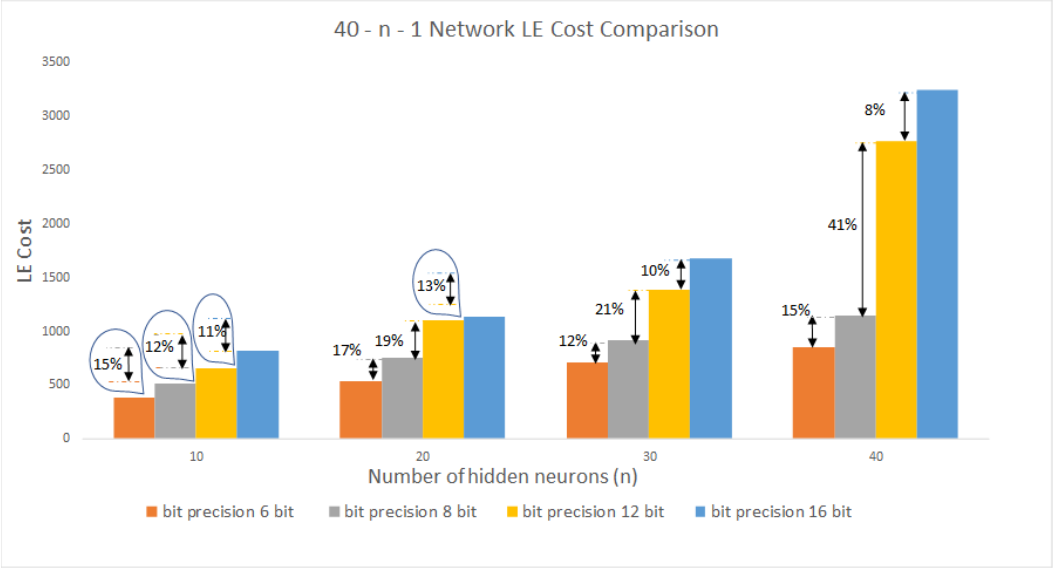


Figure 4.22: 40-n-1 Network Cost Comparison

our application still requires an up to date development chip such as one which is on par with Cyclone V.

## 4.5 Conclusive Remarks

In conclusion, different methods of implementing a neural processor were discussed as part of the introduction to the research solution. After much consideration, the vector approach proved to be the most suitable approach in providing an alternative approach in the ongoing research for epilepsy detection. The full design of a vector processor is discussed in detail with a novel, efficient, yet low cost data processing unit (DPU) as the basis of the neural processor. This vector processor will form a single layer of the neural network. A simple central control FSM provides the necessary states to begin the neural operation. In this research with a 12 bit architecture, the DPU requires 39 logic elements; a state machine and the control path needs 48 logic elements.

The research proceeded with the testing of EEG waveforms using the vector processor network design to test the full functionality of the network in the field of medicine. The peak detection neural network design which is a fairly complex neural network is displayed in Figure 4.1 which can be used as an alternative approach in the ongoing research of developing an epileptic detection classification device.

In summary, this proposed compact vector processor design is a novel methodology based on many basic architecture and ideas to allow the construction of complex hardware ANNs for the use in the application of epilepsy detection. The ideal functionality of the neural network is to be used as an ASIC chip for a portable healthcare device that can be made available easily for epileptic patients to use in daily life, which can identify and predict the occurrences of any impending major seizures.

Chapter 5

# EEG Feature Analysis for Complete Epilepsy Prediction System

This chapter incorporates the features extracted from the EEG waveform, the slope and mean energy value. These features are used as inputs to the BSNN. In order to accomplish this, different dedicated feature extraction hardware have been designed and tested on Cyclone V FPGAs. Figure 5.1 illustrates the proposed system and their connections. The full BSNN design is explained in detail in Chapter 4 and the proposed DPU in Chapter 3. Different BSNN configurations have been used to find the optimal network configuration for this application. Figure 5.1 also show that each hidden layer can consist of $n$ number of DPUs and the end result (u) is taken from the DPU in the output layer.

The chapter is organized as follows: Section 5.1 will first explore the optimal allocation technique for the EEG signal that is needed for the experiments conducted in this chapter; Section 5.1.1 experiments with various network configurations proposed by related work; Section 5.2 illustrates the two proposed feature extractors mentioned above and experiments conducted using these hardware; Section 5.3 presents the complete system with the results obtained from multiple experiments; Section 5.4 compares our complete system and results with other related work; Section 5.5 concludes this chapter.

## 5.1 Optimal Allocation Sampling of EEG Signals

This section presents the methodology used to obtain an optimized hardware-based BSNN for epilepsy diagnosis. It is based on the proposed DPU design in Chapter 3. Various techniques have been reviewed in order to optimize the hardware for epilepsy

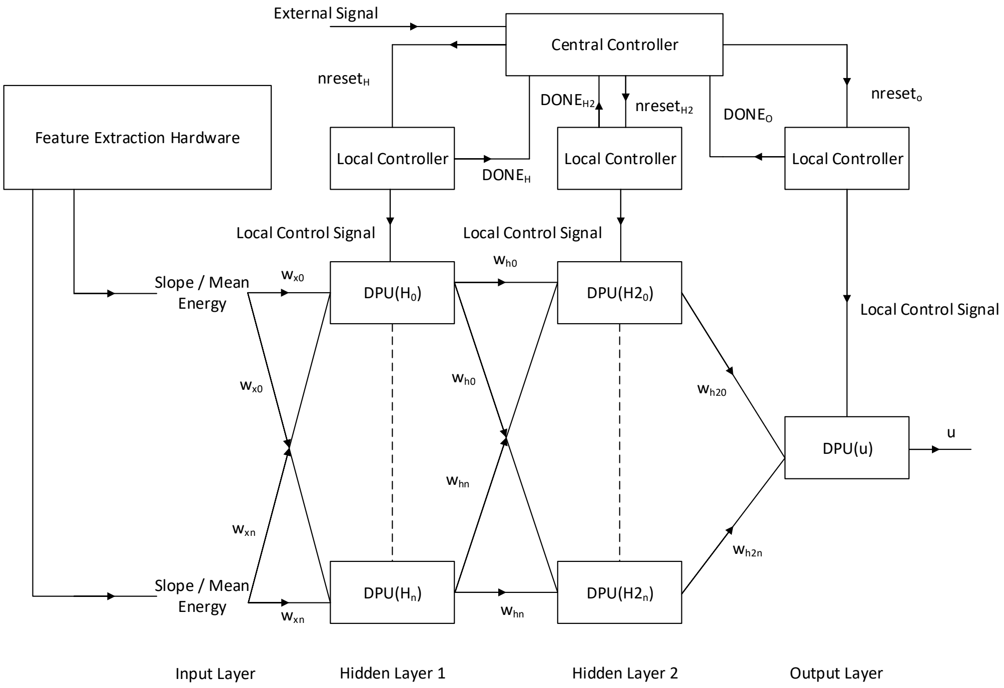


Figure 5.1: Proposed System Design

detection such as optimal allocation sampling techniques (OAT) [134]. Two feature extraction hardware are proposed in this chapter which involves the slope of an EEG waveform and the mean energy value which will be fed as inputs into the BSNN.

The proposed BSNN hardware can be optimized by using the feature extraction. With OAT [134], the extracted features are then used as inputs to the BSNN. These two equations (equation 5.1 5.2) are used during the sample allocation for each segment. These equations can be implemented using a simple MATLAB script.

The EEG dataset used in this research is taken from an on-line open source [36], the Epilepsy Center of the University of Bonn, Germany [60]. The EEG waveforms have 4097 data points for each and every channel (100 channels). Therefore, $N$ in this study is the number of data points on a single channel.

Equation 5.1 is used to calculate $n_o$, the desired sample size using: $z$, the standard normal variate of a desired confidence level (commonly 95% or 99%); $p$, an estimated proportion of an attribute (seizure or free seizure) present in the population, $N$; $d$, the margin of error.

The parameters used:

- $z = 2.58$ (value used to achieve the desired confidence level of 99% [134])

- $p = 0.5$ (Chosen as characteristic remain unknown in order to produce the maximum sample size)
- $d = 0.01$ (error margin for the desired 99% confidence level)

$$n_o = \frac{z^2 * p * (1 - p)}{d^2} \tag{5.1}$$

With the parameters used, $n_o$ obtained is 16641. $n$ which is the sample size needed in each class (free seizure or seizure) can then be calculated using equation 5.2 with $N$, the total population of a single EEG waveform (4097 data points) [134].

$$n = \frac{n_o}{1 + \frac{(n_o - 1)}{N}} \tag{5.2}$$

The initial population for each EEG window, $N_i$ (segment 1,2,3) is 1024 samples and 1025 for segment 4 respectively. The numbers are found by dividing $N$ by 4. Table 5.1 provides the determined number of samples using OAT,$b_i$ for each segment in the EEG dataset. As the EEG dataset used in this thesis is the same as in Kabir et al. work [134], the number for each segment have been taken from their paper to allow consistent comparison with the thesis design.

| **Classes** | $n_1$ | $n_2$ | $n_3$ | $n_4$ | **Total $n$ per class** |
|---|---|---|---|---|---|
| Free Seizure | 839 | 841 | 780 | 828 | 3288 |
| Seizure | 833 | 844 | 815 | 796 | 3288 |

Table 5.1: Sample Number Determined using OAT for Each Segment

The sample sizes in each class are different because of the variability of the samples in each segment. The size of each segment depends on the variability, large variability will lead to a larger EEG segment and vice versa. The sample data are extracted from the dataset used in Chapter 4. In Chapter 4, the testing focused on detecting seizure waveforms on a single channel. This chapter attempts to distinguish the waveforms on all 100 channels. This will produce more reliable results.

Figure 5.2 illustrates the work flow of the design decision made in optimizing the design for this thesis.

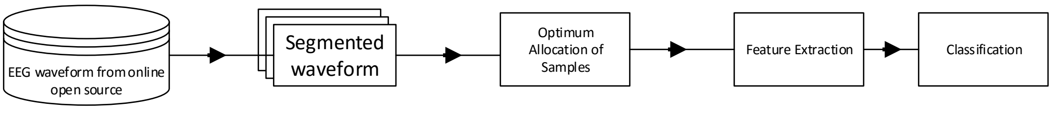

EEG waveform from online open source
Segmented waveform
Optimum Allocation of Samples
Feature Extraction
Classification

Figure 5.2: Work Flow of Epilepsy Detection with Optimum Allocation

### 5.1.1 Optimal BSNN Configuration for Epilepsy Detection

In order to fully develop a small and energy saving hardware device for epilepsy detection, the BSNN design proposed in this research has been optimized further following the experiments conducted in Chapter 4. This section explore various methods in order to optimize the hardware network to provide a decent level of correct recognition when distinguishing both free seizure and seizure waveforms.

There are certain rules of thumb for ANN optimization. Input layers mainly consist of $n$ number of neurons which depends on the input data. The number of inputs are optimized using the optimal sample allocation technique discussed in the previous section. Output layer neurons will depend on the number of classes that needs to be classified. The number of hidden neurons usually lies between the number of input and output neurons. However, a n-1-1 neural network would not perform well in detecting a epileptic waveform as shown in Chapter 4.

Next, we explore various network configurations to achieve the best correct recognition rate possible. These different configurations include high amount of hidden neurons and also different number of hidden layers. The experiment begins with a 11-7-1 network. The whole EEG waveform obtained from the benchmark dataset is segmented into 4 independent segments. Each segment is taken with a period of 5.9 s from a whole waveform of 23.6 s. A comparison of the accuracy and other metrics will be included in a later section.

### 5.1.2 Hardware Network Validation and Testing

As the research's main focus is to distinguish free seizure waveform from seizure waveforms, the healthy patient brain waveforms from the database used are not included in the design testing. This design is tested and compared with different software implementations for epilepsy detection [135].

Using the training dataset, the 11-7-1 hardware neural network with a 12 bit architecture has a sensitivity, specificity and sensitivity of 60%. It could recognise 30 out of 50 waveforms when training the network in MATLAB. In order to perform this controlled experiment, the same inputs will be used that consist of the same feature vectors.

The feature vector values consist of the same metrics as those provided in a related work. These values contain mean ($X_{Mean}$), median ($X_{Median}$), mode ($X_{Mode}$), standard deviation ($X_{StdDev}$), first quartile ($X_{Q1}$), third quartile ($X_{Q3}$), inter-quartile range ($X_{IQR}$), skewness ($X_{skew}$), kurtosis ($X_{kurtosis}$), minimum ($X_{Min}$), and maximum ($X_{Max}$)[135].

Ten other network configurations have been designed and tested. These configurations were chosen using MATLAB in order of decreasing mean square error (mse). Table

5.2 presents the configurations and their corresponding mean square error. When the hardware results were obtained, a compromise between the size and performance will be made and the design used as the ASIC model in the next chapter.

From Table 5.2, it can be seen that a single hidden layer with 100 neurons has a similar performance to that of a double layer network (10 neurons in each layer). Furthermore, it is apparent that the results have a much lower disparity when identifying waveforms from different sets of data.

| **Network Configuration** | **Correct Recognition against training data** | **Correct Recognition against additional tests** |
|---|---|---|
| 11-25-1 | 52% | 60% |
| 11-40-1 | 56% | 50% |
| 11-65-1 | 60% | 30% |
| 11-100-1 | 66% | 55% |
| 11-10-10-1 | 62% | 60% |
| 11-20-20-1 | 56% | 80% |
| 11-30-30-1 | 58% | 60& |
| 11-40-40-1 | 64% | 45% |
| 11-10-10-10-1 | 54% | 50% |
| 11-5-5-5-1 | 56% | 30% |

Table 5.2: Correct Recognition of different hardware ANN configuration

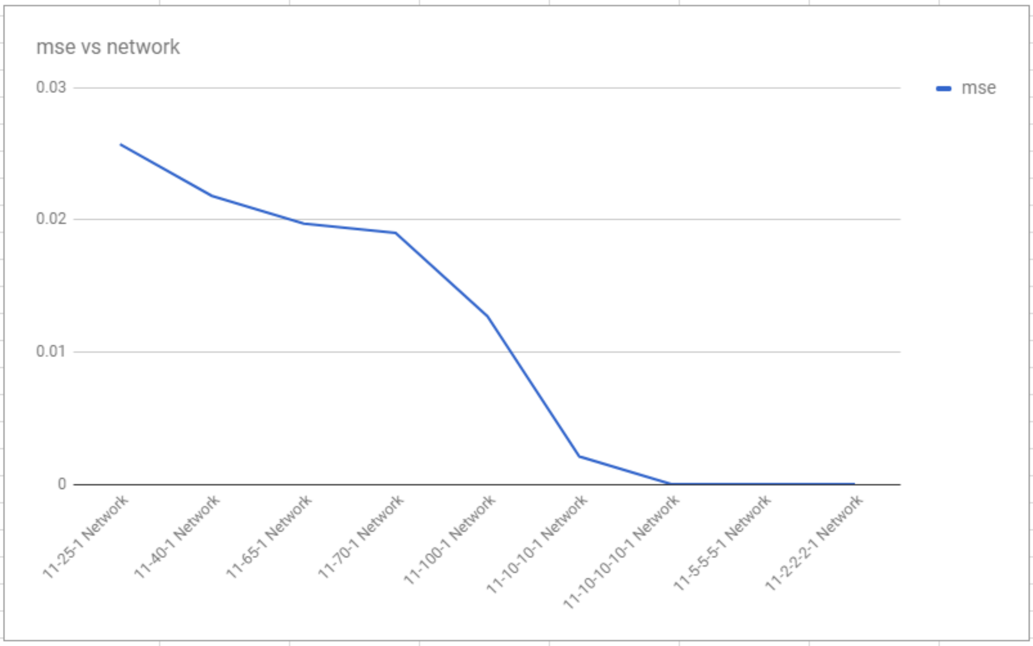


Figure 5.3: Performance (MSE) of different ANN configuration

By analysing these results, it can be seen that this simple feature vector may prove lacking in providing a very accurate classification. Thus, it is decided to proceed with a different input vector consisting of multiple slope values of the EEG waveform. These values can be preprocessed to form a feature vector.

## 5.2 Proposed Feature Extraction Hardware

In order to complete the wearable seizure detection system, it is imperative to develop a simple feature extraction hardware as a component to provide inputs to the BSNN designed in Chapter 4. The two different feature extraction hardware consist of a single ALU with two synchronous RAM and a simple controller.

### 5.2.1 Slope calculator

The data path of the feature extractor illustrated in Figure 5.4 consists of a synchronous RAM, a simple subtractor implemented as an ALU and registers. The data path is controlled by a simple FSM module. The hardware cost for the ALU requires only 13 ALMs when synthesised on a Altera Cyclone V chip. This hardware will serve as a method of extracting the slope, $S$ of the EEG waveform from two adjacent points ($x_1$ and $x_0$) on the EEG sample. It is calculated using this simple equation, $S = x_1 - x_0$. Each $S$ value is stored in the registers and used as inputs for the BSNN. As part of the research, it is possible to build the extraction hardware in two different forms. The first method is as described above with the use of multiple registers. The second method is possible to build the same number of subtractor modules as the number of input neurons for the BSNN. However, the latter method would require many more logic elements and time to complete the operation which is not suitable for the specification of a low cost and efficient wearable epilepsy detection system.

Figure 5.4 provides a simple understanding of the connection between the different modules. The feature extraction hardware can be controlled easily with a simple state machine. With the inclusion of the feature extraction hardware, a new network configuration of 100 inputs (100-40-40) is tested and synthesised. This new network configuration has a 75% correct recognition rate when tested against training and additional data.

### 5.2.2 EEG waveform slope Used as Feature Vector

In the previous subsection, a feature vector consisting of various statistic metrics is used. The maximum accuracy was 80% when tested using additional data. However, the disparity when testing the same network configuration against the training data should be noted. Thus, 11-20-20-1 network shows some promising results.

This subsection presents results of experiments that have been conducted to obtain better accuracy by using the slope of the EEG waveform using multiple slope values as

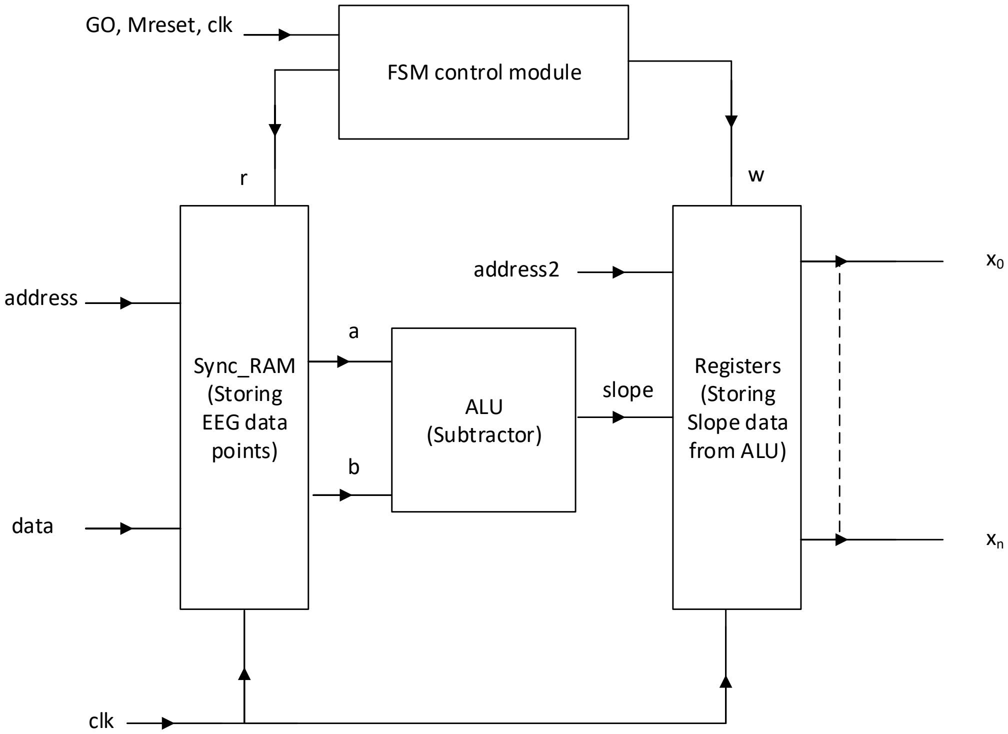


Figure 5.4: Feature Extraction Hardware (Slope Feature)

a feature vector. The tested network configurations are 11-10-10-1, 11-20-20-1, 11-30-30-1 and 11-40-40-1. The results are evaluated using the same statistic metrics used in the above section. The recognition rate for each configuration are included in Table 5.3. The metrics are presented in Table 5.4 and Table 5.5. With 11 inputs, the best correct recognition rate that was obtained was the 11-40-40-1 configuration with 70% and precision rate of 100% when tested using training data. When tested with additional data, the network configuration have an recognition rate of 61% and a precision rate of 80%.

| **Network Configuration** | **Correct Recognition against training data** | **Correct Recognition against additional tests** |
|---|---|---|
| 11-10-10-1 | 60% | 70% |
| 11-20-20-1 | 54% | 60% |
| 11-30-30-1 | 65% | 40% |
| 11-40-40-1 | 70% | 61% |

Table 5.3: Correct Recognition of different hardware ANN configuration using EEG waveform slope

| Network Configuration | TPR | TNR | PPV |
|---|---|---|---|
| 11-10-10-1 | 57% | 100% | 80% |
| 11-20-20-1 | 52% | 44% | 42% |
| 11-30-30-1 | 66% | 64% | 58% |
| 11-40-40-1 | 63% | 100% | 100% |

Table 5.4: Statistic for Network Configuration Evaluation (Against Training Data)

| Network Configuration | TPR | TNR | PPV |
|---|---|---|---|
| 11-10-10-1 | 75% | 33% | 43% |
| 11-20-20-1 | 50% | 50% | 40% |
| 11-30-30-1 | 25% | 44% | 10% |
| 11-40-40-1 | 53% | 33% | 80% |

Table 5.5: Statistic for Network Configuration Evaluation (Against Additional Data)

### 5.2.3 Experiments with Mean Energy

In addition to the slope calculator presented above, another feature was also extracted from the EEG input signals which is the energy of a designated EEG signal window. This feature would be able to perform better when dealing with larger sample datasets, i.e. datasets that involve over 700 data points. The equation used is included here which is adapted from the equation presented in a recent work [136].

$$MeanEnergy = \frac{1}{w} * \sum_{i=1}^{I} a(i)^2 \tag{5.3}$$

The mean energy is calculated once the summation of energy from each window is obtained. $a$ is the amplitude values of the EEG signal spikes. $w$ then represents the number of $a$ values being used. A new system which uses the feature extraction hardware component was used on FPGAs and achieved a 62% accuracy in 50 different EEG samples. 25 of the samples are free seizure and 25 are EEG seizure waveforms. Figure 5.5 below presents the output obtained from the experiment. The waveforms that are not classified correctly are clearly circled in red in the figures. The low accuracy could be attributed to the way of feature extraction. Raw data from the EEG dataset was used to obtain the mean energy feature but the work [136] treated the data before feature extraction.

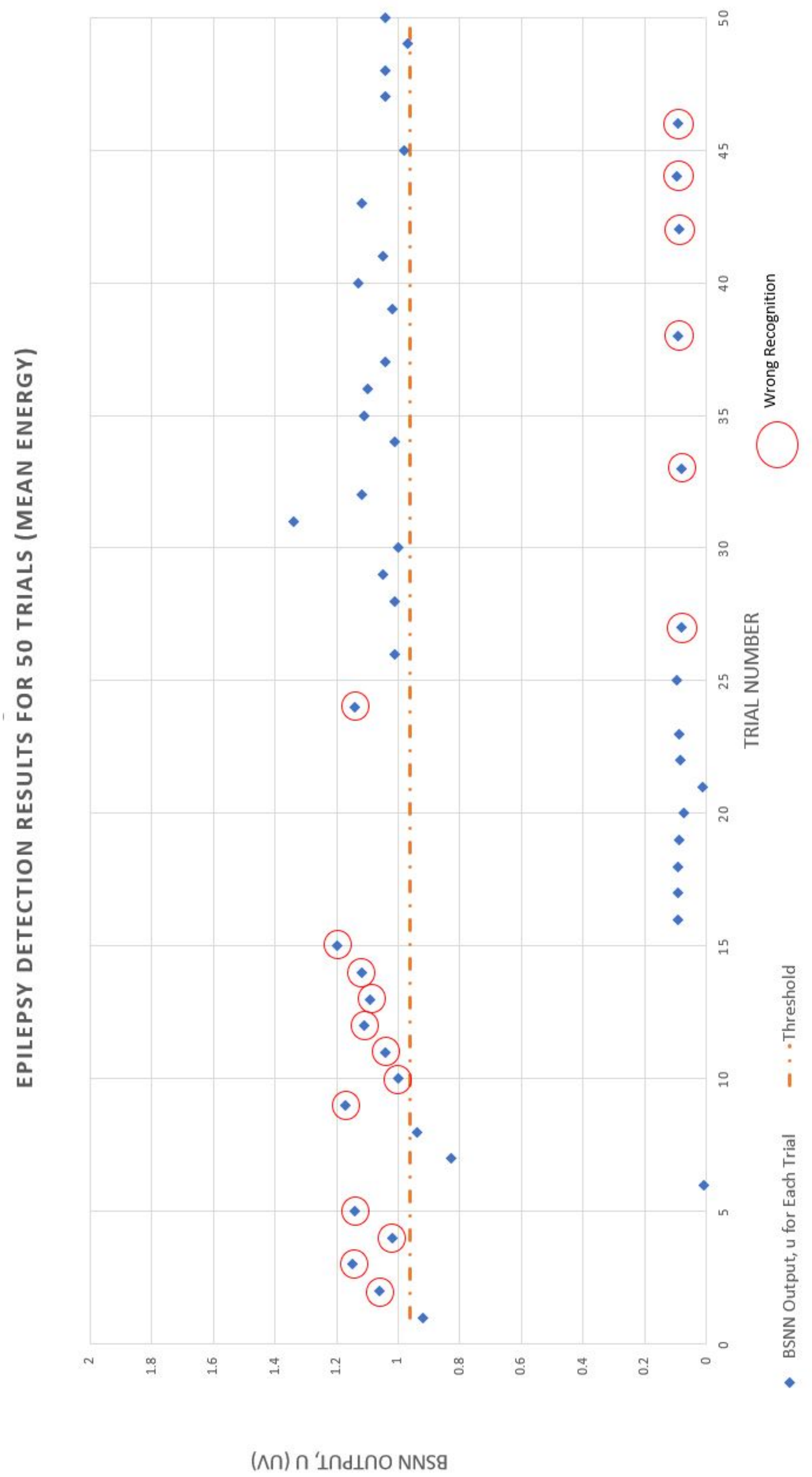


Figure 5.5: Mean Energy System Experiment Output

## 5.3 Proposed System: Feature Extraction + BSNN

As part of the plan to create a simple and wearable epilepsy detection system, this research integrated the feature extraction hardware with the input to the BSNN. The feature extraction hardware will use raw EEG input data points to extract the desired features, in this case the slopes between two adjacent points. The network configuration with a double hidden layer has been tested with a different number of inputs yielding different results.

The first configuration consist of 11 inputs with 40 hidden neurons in each layer. The next configuration consists of 50 inputs with the same number of hidden neurons and the last configuration has 100 inputs. Table 5.6 below presents the correct recognition rate and estimated power consumption of these designs.

| **Network Configuration** | **Correct Recognition (Against Training Data)** | **Correct Recognition (Against Other Data)** | **power consumption (estimated (mW))** |
|---|---|---|---|
| 11-40-40-1 | 70% | 61% | 485.97 |
| 50-40-40-1 | 75% | 75% | 497.8 |
| 100-40-40-1 | 90% | 90% | 513.06 |

Table 5.6: Network Configuration with a different number of inputs

With these results, it can be seen that the 100-40-40-1 network configuration provides the best detection possible and can be a reliable starting point for any future research.

This whole system has been tested and synthesised on an Altera FPGA board to verify its functionality. Different multi-layer hardware neural network examples are included here along with their costs. The network designs and costs will be compared with some other recent state of the art design in this subsection. The designs included here are a 100-20-20-1 and a 100-40-40-1 configuration.

### 5.3.1 Improved System

The newly improved system includes both mean energy and slope values extracted from the EEG signals to be used as features for the proposed network. The main network that is tested and compared to is the 100-40-40-1 network configuration with only slope features. The main network obtained a 88% recognition rate. With this improved system, a 90% recognition rate can be obtained. The experiment uses 20 samples for the additional tests. Figures 5.6, 5.7, 5.8 illustrates the output obtained when conducting these experiments. The trials that are not correctly classified are clearly circled in red in the figures. When 16, 12, 8 bits are used, 194us, 374us and 612us are needed to complete a single trial respectively.

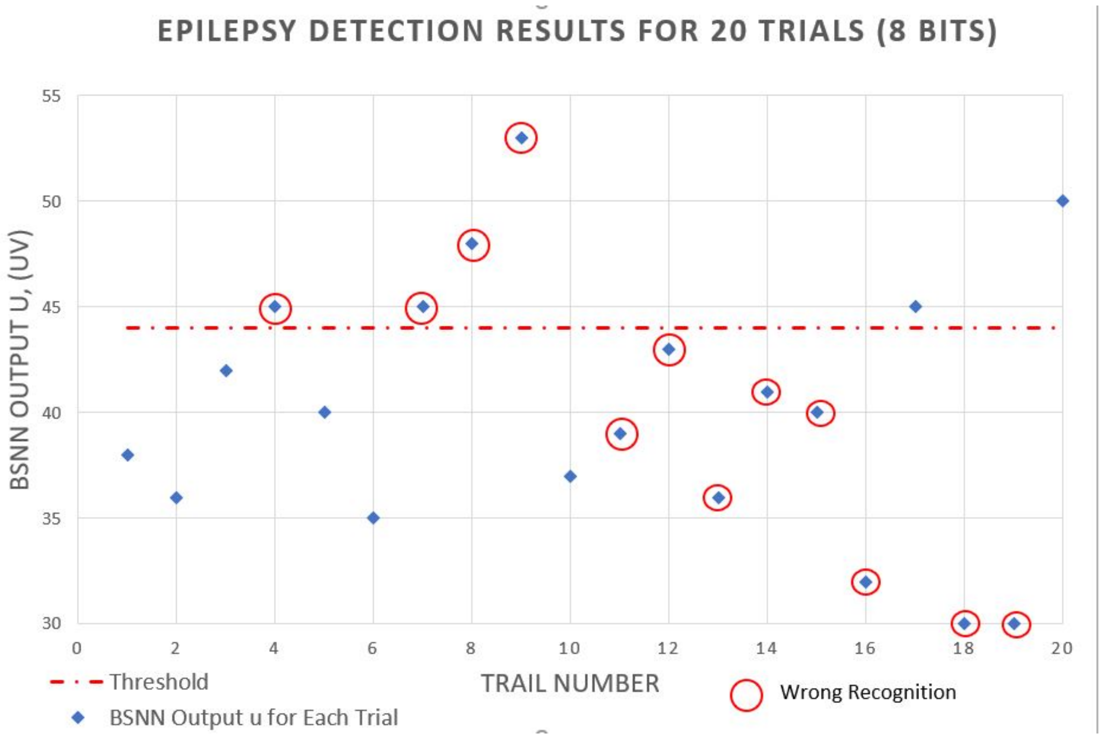


Figure 5.6: Output of Improved System Using 8 bit Architecture

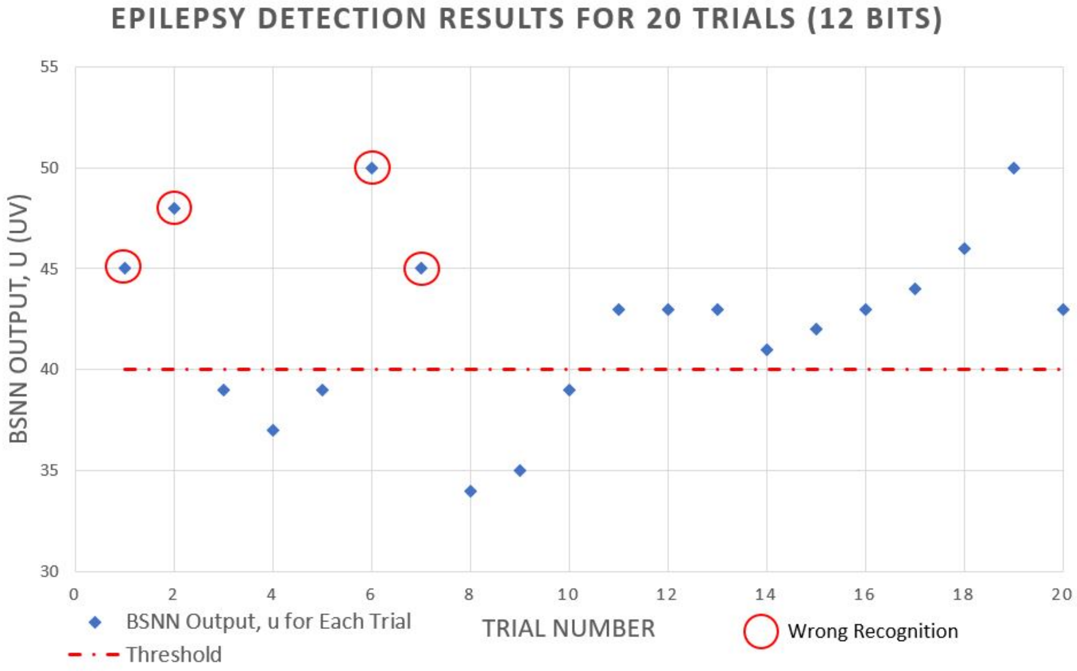


Figure 5.7: Output of Improved System Using 12 bit Architecture

Detailed comparison of the statistics obtained from the experiment output is shown clearly in Table 5.7 below.

From the statistics obtained from the experiments, it can be seen that a 16 bit system has the highest correct recognition rate. However, this system could be made smaller at

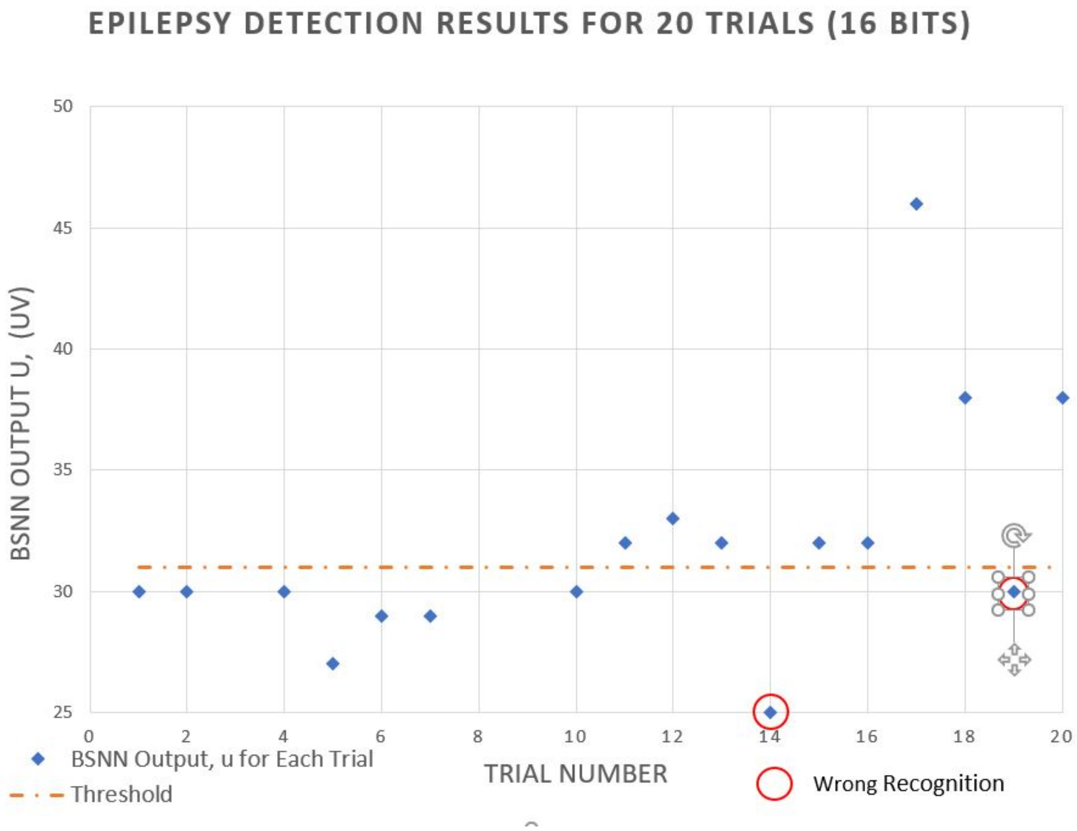


Figure 5.8: Output of Improved System Using 16 bit Architecture

| Bit Architecture | Recognition rate | TPR | TNR | PPV |
|---|---|---|---|---|
| 16 | 90% | 100% | 83% | 80% |
| 12 | 80% | 100% | 60% | 71% |
| 8 | 40% | 33% | 43% | 33% |

Table 5.7: Improved system statistics using 100-40-40-1 network configuration

the expense of some accuracy as a 12 bit system still have a high possibility of correctly identifying a seizure event.

### 5.3.2 Potential for Massively Parallel BSNN System

The proposed dedicated hardware neuron in this research has shown that it is possible to create a complex neural network. This section attempts to explore the possibility to of creating a massive parallel neural network using the proposed dedicated hardware. The method of generating a massive number of neuron hardware SystemVerilog code involves a basic python script. This will reduce the possibility of any human error when writing the code by hand.

Python scripts were produced to generate the required hardware code for the various designs used in the experiments of this thesis, which can remove the possibility of any human errors. Heavy text manipulation was used in the scripts. A loop module was integrated into the script to allow any number of DPUs to be generated. The weights and inputs extracted from the data files were preprocessed and included in the hardware codes by using the scripts.

A few samples of the optimized system had been synthesised and tested on an FPGA Cyclone V board. Furthermore, Table 5.8, 5.9, 5.10 present a comparison between different samples of the complete system and their respective hardware cost. The cost comparison is made using three different bit architectures, 8; 12; 16.

| **Network** | | **Bit Precision** | |
|---|---|---|---|
| **Configuration** | 8 bits | 12 bits | 16 bits |
| 100-25-25-25-1 | 7527 LE | 10737 LE | 12546 LE |
| 100-80-1 | 9704 LE | 12227 LE | 15527 LE |
| 200-100-1 | 11242 LE | 20043 LE | 21208 LE |

Table 5.8: Hardware Cost (LE) Implemented on Cyclone IV FPGA

| **Network** | | **Bit Precision** | |
|---|---|---|---|
| **Configuration** | 8 bits | 12 bits | 16 bits |
| 100-25-25-25-1 | 2718 LE | 4945 LE | 5532 LE |
| 100-80-1 | 4212 LE | 5259 LE | 6073 LE |
| 200-100-1 | 3897 LE | 6569 LE | 7907 LE |

Table 5.9: Hardware Cost Implemented on Cyclone V FPGA

| **Network** | | **Bit Precision** | |
|---|---|---|---|
| **Configuration** | 8 bits | 12 bits | 16 bits |
| 100-80-1 | 4211 LE | 5463 LE | 5284 LE |
| 100-25-25-25-1 | 2693 LE | 4387 LE | 4905 LE |
| 200-100-1 | 4979 LE | 7658 LE | 7607 LE |

Table 5.10: Hardware Cost Implemented on Stratix IV FPGA

Figure 5.9, 5.10, 5.11 present a clear picture of the hardware cost difference between each system using variable bit precision. Figure 5.10 shows that the 100-80-1 network have the lowest hardware cost difference when implemented on three different FPGAs. From the tables and figures above, it can be seen that the Cyclone V FPGA is more conducive for implementing our proposed system.

## 5.4 Comparison with Related Work

The optimized design was first tested against the whole range of EEG waveform obtained from an on-line open source [36] provided by the Epilepsy Center of the University of

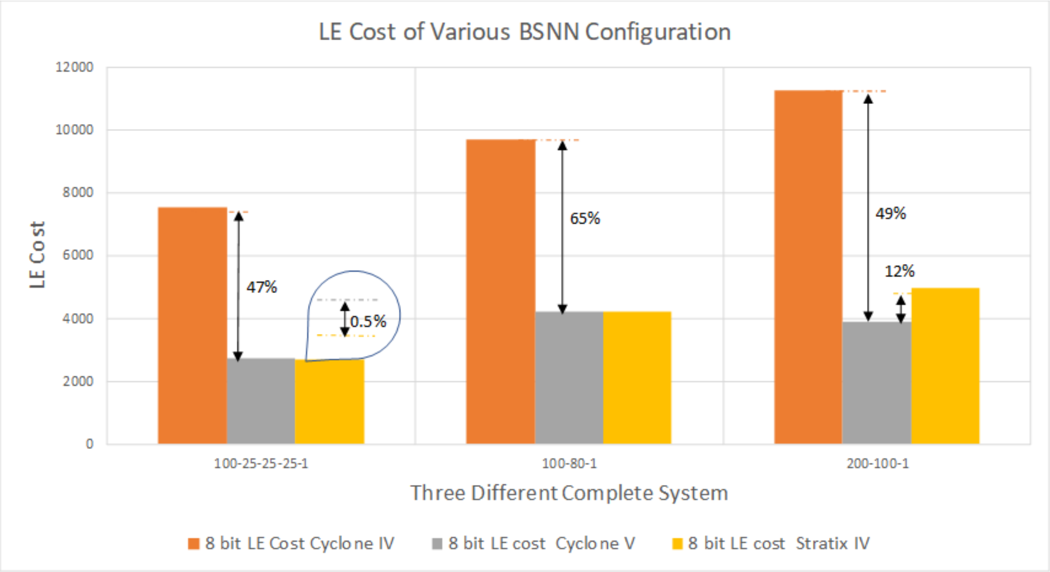


Figure 5.9: LE Cost of Three Different Systems Using 8 Bit Precision

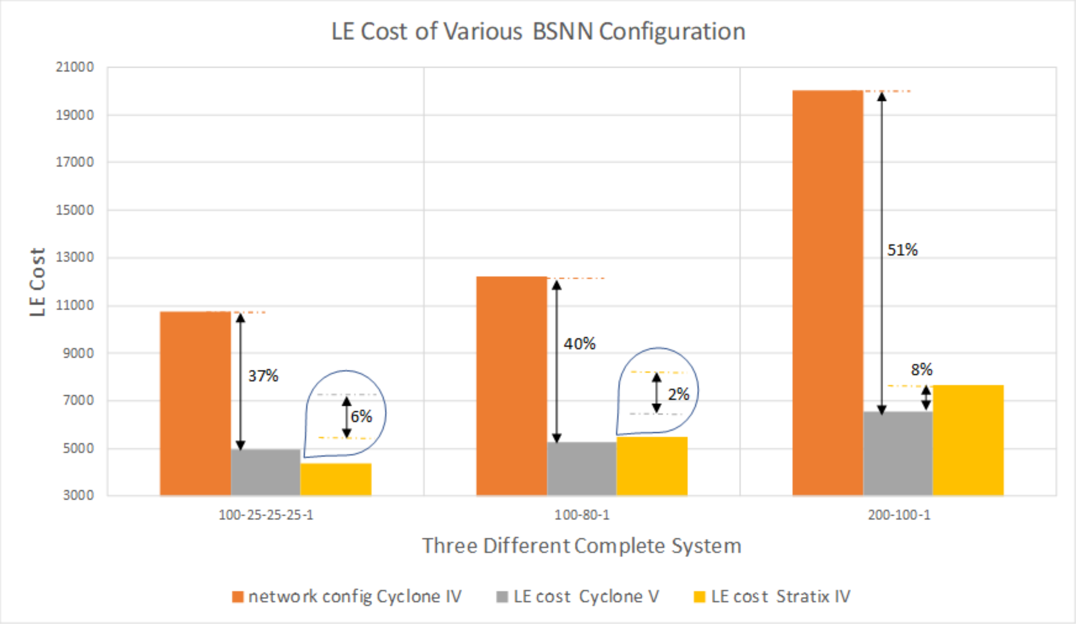


Figure 5.10: LE Cost of Three Different Systems Using 12 Bit Precision

Bonn, Germany [60]. The source provide sets of EEG waveforms for both seizure free instances and EEG waveforms during seizures taken from the brain (epileptogenic zone) of the same patient [29]. The datasets were separated into four different segments using the OAT methodology mentioned in the section 5.1. The results of the tests are included here in Table 5.11 and the output values are illustrated clearly in Figures 5.12, 5.13, 5.14, 5.15. Half of the trials included in our experiments consist of free seizure samples and the other half are seizure samples. Each trial is a single EEG window taken from the

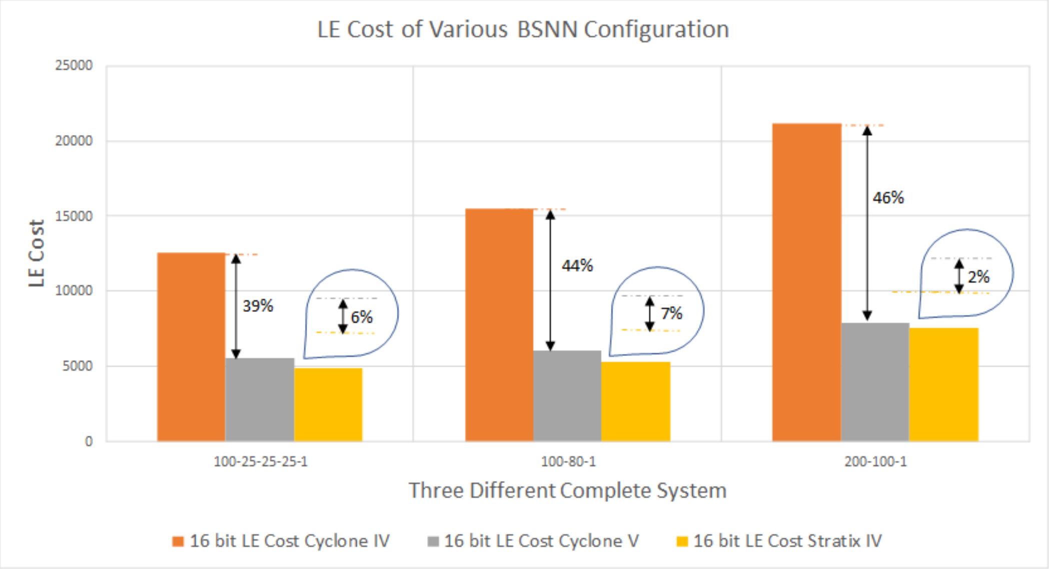


Figure 5.11: LE Cost of Three Different Systems Using 16 Bit Precision

EEG waveform provided by the open source mentioned above. The waveforms that are wrongly recognized is clearly circled in red in the figures.

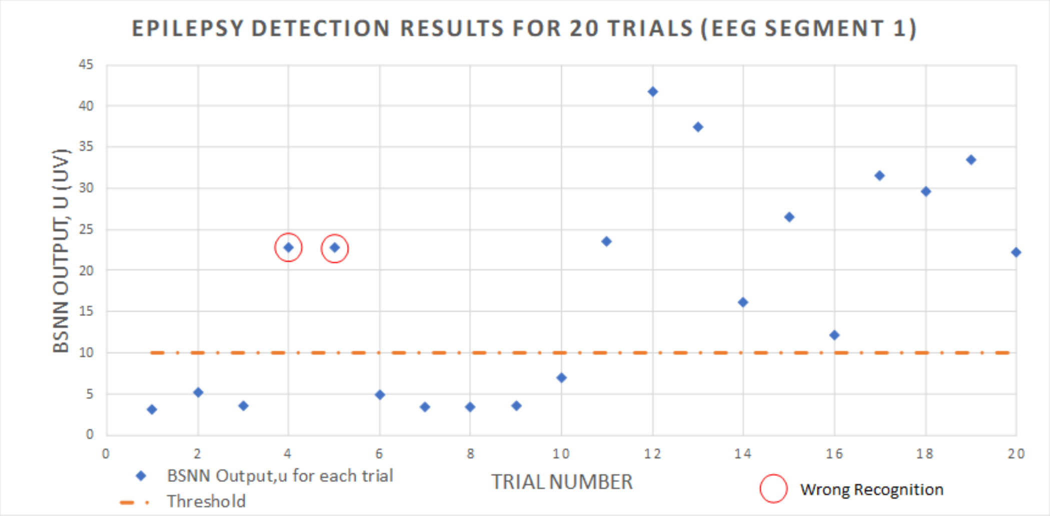


Figure 5.12: Segment 1 Output Using Slope Feature

| EEG Segment | Correct Recognition | TPR | TNR | PPV |
|---|---|---|---|---|
| 1 | 90% | 83% | 100% | 100% |
| 2 | 90% | 83% | 100% | 100% |
| 3 | 85% | 82% | 78% | 82% |
| 4 | 90% | 100% | 83% | 100% |

Table 5.11: Results Obtained when tested with different EEG Segments

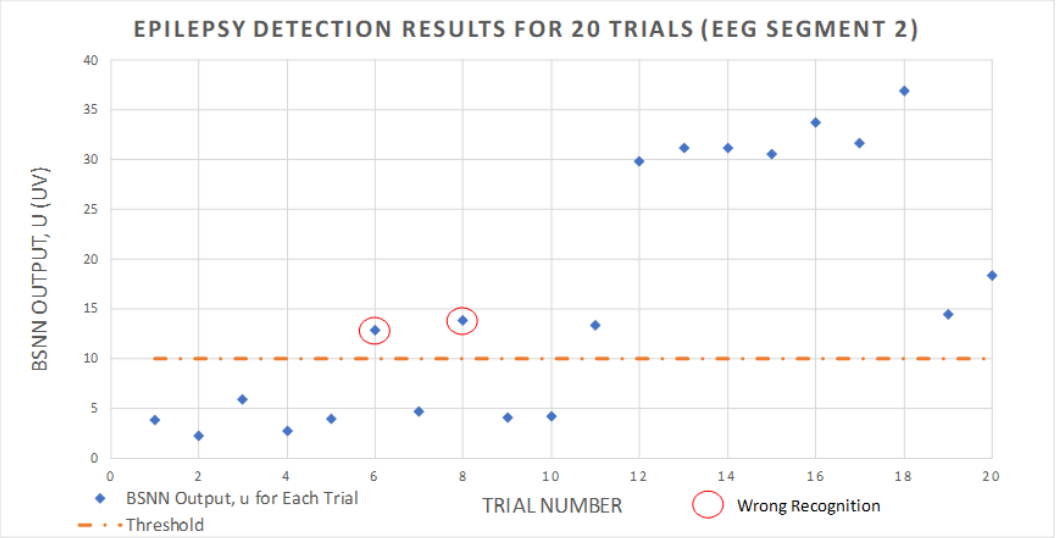


Figure 5.13: Segment 2 Output Using Slope Feature

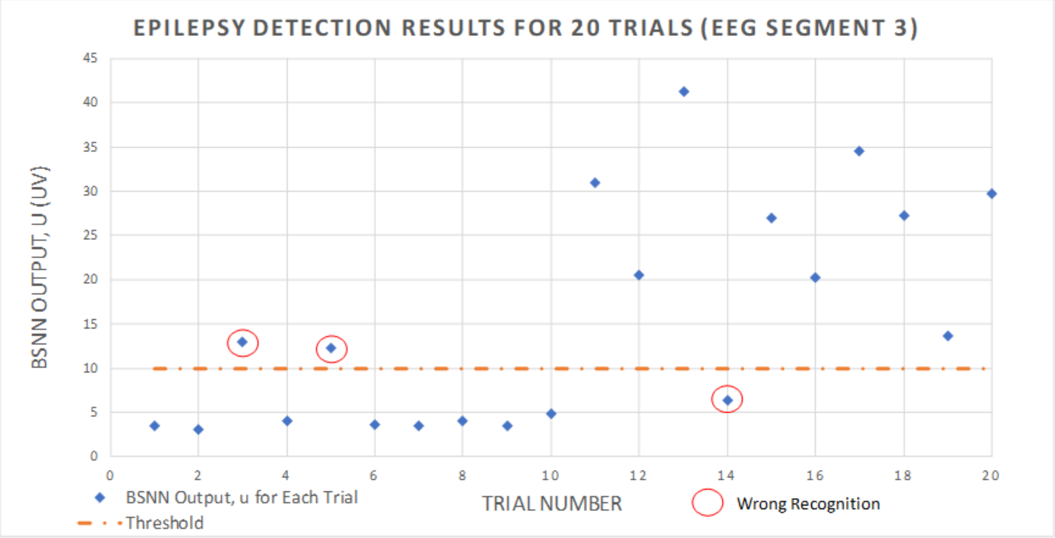


Figure 5.14: Segment 3 Output Using Slope Feature

Both optimized hardware neural network systems are tested and compared against several software implementations commonly used for epilepsy detection [135, 134]. When compared with the results from another paper [134], it is possible to argue that the design proposed and developed in this thesis works better when compared with the SVM approach mentioned in the paper. As the design will need to be a simple wearable hardware design, many more input neurons are used in comparison with the design proposed in the paper [134]. The software implementation of a epilepsy detection system used in the paper [134] were LMT, MLR and SVM classifiers. Table 5.12 below presents a close comparison between our design and the designs implemented in the paper [134]. From the results, both optimized design proposed in this research fair much better than an

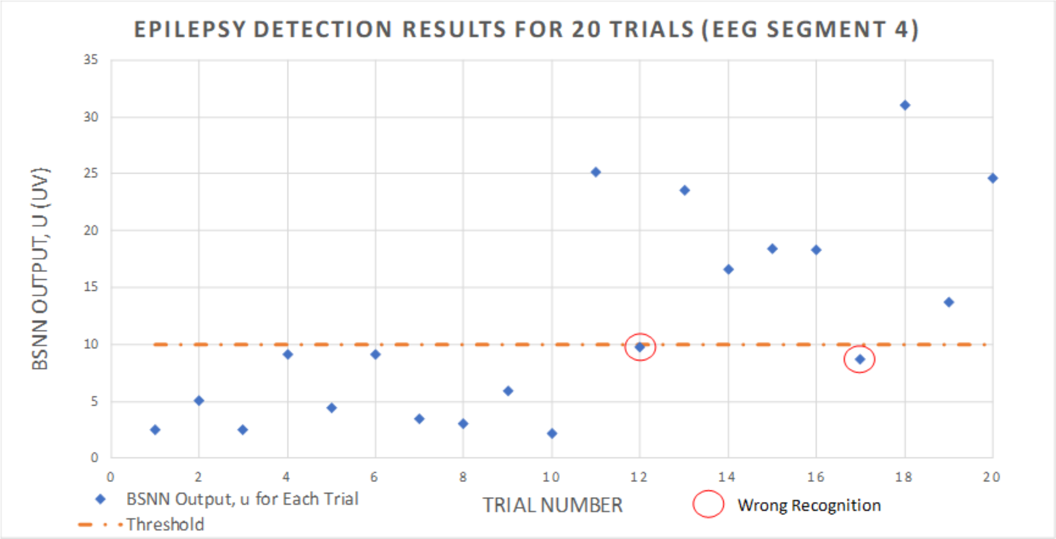


Figure 5.15: Segment 4 Output Using Slope Featuret

SVM approach and has the level of competency between a MLR and a LMT classifier. The system using a slope feature can provide a reliable accuracy with a lower hardware cost when compared with the latter system that uses both mean energy value and slope feature.

| Classifier | Overall Accuracy | TPR | TNR | PPV |
|---|---|---|---|---|
| LMT | 95.33% | 95.3% | 97.7% | 95.3% |
| MLR | 82.67% | 82.7% | 91.3% | 82.9% |
| SVM | 36% | 36% | 68% | 78.1% |
| BSNN (S) | 88.8% | 87% | 90.25% | 95.5% |
| BSNN (S & E) | 90% | 100% | 83% | 80% |

Table 5.12: Results Obtained when tested with different Classifiers (S = Slope Feature, E = Mean Energy Feature)

| Hardware Classifier | Overall Accuracy | Estimated Power (mW) | Development Chip (Cost) | Latency |
|---|---|---|---|---|
| FPGA based co-processor [79] | - | - | Virtex II - 8% | - |
| SOM Neuroprocessor [10] | 80% | - | Virtex II - 17% | - |
| Stripes [106] | 95% | Varies | ASIC - 122.1 mm$^2$ | Varies |
| BSNN (S) | 88.8% | 500 | Cyclone V - 4% | 500$us$ |
| BSNN (S & E) | 90% | 600 | Cyclone V - 4% | 600$us$ |

Table 5.13: Results Obtained when tested with different Hardware Classifiers (S = Slope Feature, E = Mean Energy Feature , Cost = in terms of total hardware resources provided by the development chip)

There exists a possibility of unfair comparison between software and hardware techniques such as in Table 5.12. The thesis attempts to reduce such possibility by providing more comparison between different proposed epilepsy detection hardware system as listed in Table 5.13 below. This table presents different hardware classifier for epilepsy detection as proposed in different literature. The common metrics used in this comparison mainly includes overall accuracy, power, LE cost and latency. From the table, it can be seen that our design (BSNN) is smaller and slightly more accurate then the SOM neuroprocessor. However, the speed and power could not be compared in this case as the values were not reported in the literature. Next, we can compare the proposed system against an FPGA based co-processor which consumes 8% of the total hardware resource provided by a Virtex II FPGA development chip. Our design is smaller in this context yet we could not compare with other parameters.

Furthermore, we have compared our design with a bit-serial neural network design (Stripes) [106]. It should be stressed that Stripes has been implemented on an ASIC chip. The thesis design is still in the stage of prototyping using an FPGA. Our accuracy is slightly lower than that of Stripes. However, the author is confident with further optimization the BSNN design will surpass the Stripes design in the context of power and latency. It can also be justified that Stripes was created as an accelerator and extension to DaDianNao which differs from the thesis BSNN which was designed specifically for epilepsy detection.

It can also be seen that both of BSNN proposed differs only in terms of overall accuracy.

## 5.5 Conclusive Remarks

In conclusion, this chapter reviewed different methods used to optimize the hardware neural network in order to develop a low cost and efficient ASIC prototype in the near future. Optimum allocation of samples would provide a non-bias array of samples. Different type of inputs have been used as a comparison (i.e. feature vector [135], slope feature vector and mean energy value). The statistic of these three systems are included in Table 5.14. Furthermore, there is a 2% increase in performance for the improved system (Combination of EEG slope and Mean Energy Feature); yet the 12-bit network using only EEG slope features can still provide a reliable performance when predicting seizure events.

| System | Recognition rate | TPR | TNR | PPV |
|---|---|---|---|---|
| EEG slope system | 88% | 87% | 90% | 95% |
| Improved system | 90% | 100% | 83% | 80% |
| Mean energy system | 62% | 59% | 67% | 76% |

Table 5.14: Comparison between three different proposed systems

There are some scientific merits to this proposed system that will be discussed here. As a proof of concept, the research team have conceived a viable low energy and low hardware cost epilepsy detection system in this thesis. Being a low energy and low hardware cost system, it is of great interest to health personnels as it can be integrated into complex health monitoring environment such as AAL and smart homes. Next, it is crucial for the reader to understand the significance of the algorithm chosen by the research team which is a simple shift and add algorithm. There is also the use of a simple threshold activation function to determine the existence of a seizure event. Both of this are significant as they are simple to implement and do not put a lot of stress on the hardware system and this in turn reduces the amount of power required for the hardware to function properly. Furthermore, the research team finds that this work will impact modern hardware / software architecture in many ways, i.e. increase in the use of bit-serial technology in areas that are viable and in certain cases implementing the epilepsy detection system through a fully software approach. By analysing the impact of this result, it is possible to find some means to integrate the proposed system into different uses such as smart homes and AAL environment as mentioned above. By using this proposed hardware system as a long term health monitoring device / system, various groups of individuals can benefit from it. These groups of individuals include the elderly and patients with long term disabilities or illness. The research team hopes that by implementing this system as part of the AAL environment, these individuals can live independently and safely.

In the next chapter, we present certain possibilities in the near future for this proposed system. We emphasize the possibility of implementing this system using an ASIC approach. With the ASIC model, the area and power of the proposed system can be directly compared with state of the art dedicated neural ASIC processors.

# Chapter 6

# Conclusion

The start of this thesis provides a comprehensive understanding of the problem that is tackled in the research. The contributions of this research are included in four different publications. In 2016, the first paper published by the author illustrates the feasibility of using bit-serial architecture as an alternative when developing epilepsy detection hardware [128]. Furthermore, the paper introduced a novel neural processing element which uses bit-serial architecture. In 2017, another paper was published which presents a novel data processing unit (DPU) which is the basic building block for the BSNN. Several experimental results were included in the paper [128]. In 2018, a paper was published at a symposium which reviews the work that has been done along with a proposal of a simple feature extraction hardware for the epilepsy detection system. Furthermore, a journal paper was published in 2018 which included the proposed system and comparison of the performance with other software implementations. The general organization of this thesis is also detailed in the introduction. The thesis is followed by a literature review, the proposed approach of a novel bit-serial DPU, the bit-serial neural network (BSNN) design and EEG feature analysis as well as the proposal of a novel feature extraction hardware.

In summary, the bulk of the literature review covers a large scope of studies with relation to epileptic detection, neural processors and bit-serial architecture. Chapter 2 reviews state of the art EEG research that is used to analyse and extract certain necessary features from the EEG signal. Furthermore, different types of neural processors have been researched to devise a novel approach. It is clear that a multitude of funding and focus have gone into the theoretical study and software implementation of epilepsy detection with various types of technology over the past decades. There are different prototype hardware designs that have been used in animal experimentation procedures. It would appear that the novel BSNN should incorporate the best of these prototypes [97], overcome the weaknesses and adapt the designs to accommodate the specifications of the current research. The literature review further focuses on the bit-serial architecture which is proven to be an energy-saving, reliable and cost-effective design. Different bit-serial architectures developed over the last few decades such as the COLUMNUS and bit-serial CORDIC architecture were also reviewed.

Chapter 3 introduced the novel DPU design concept to implement a biological neuron accurately. Next, the DPU design is synthesised on different FPGA boards to verify the number of logic elements needed for a single DPU. The smallest DPU was synthesised on a Cyclone V FPGA board. The cost for a DPU with 8 bit architecture only required 28 ALMs. Simple experiments were conducted in this chapter to fully test the functionality of the DPU. Three general classification problems were used for testing the BSNN: an XOR problem, a simple data classification problem and a crude ECG peak detection problem. The XOR problem result is similar to a XOR truth table. Next, correct response to epileptic EEG signal dataset is completed to test the capability of the novel BSNN design when used for epilepsy detection. Various statistic metrics were used as a method of comparison with recent works such as the sensitivity (TPR), specificity (TNR), positive predictive value (PPV), and negative predictive value (NPV). It was found that the DPU can solve simple problems perfectly and is able to be act as the basis for our bit-serial neural network.

Chapter 4 explores the potential of the bit-serial DPU to be used in a complex neural network. The hardware neural network and the design control path were proposed. In this research, the DPU with a 12 bit architecture requires 39 logic elements and the control path needs 48 logic elements. Various classification experiments were conducted. These tests are needed to fully assess the functionality of the novel neuron hardware. The training was performed off-line using MATLAB. The neural network toolbox provided by the software was used to verify which training function would be suitable for the research application. Example datasets are also provided by the simulation software. By comparing the results obtained from simulation and the synthesised hardware, the feasibility of the novel bit-serial neural network (BSNN) was verified. The above metrics were used in this chapter to compare the networks efficiency.

Chapter 5 presents an optimized epilepsy detection system with simple feature extraction hardware and the BSNN proposed in Chapter 4. The feature extraction hardware use two adjacent points on a single EEG waveform to calculate the slope to be used as BSNN inputs. Different designs were tested to find the optimum network configuration. The 100-40-40-1 network was found to have a correct recognition rate of 90% for a single segment and an overall accuracy of 88.8% with the data from all four EEG segments. This network has a precision rate of 95.5%. This approach has a better overall accuracy than certain software implementations of epilepsy detection when using the same EEG benchmark waveforms. A comparison was made with a 16 bit architecture design which had a far lower recognition and precision. Therefore, it is better to use a 12 bit architecture for this thesis.

There are a few possibilities that can be pursued in the near future. One such possibility involves the synthesis of an ASIC model of this DPU and the fabrication of this DPU. This can be done to test the size and power efficiency of this proposed design. There are some important implications of this particular research. This research presents the feasibility of an alternative approach for an wearable epilepsy detection system. Furthermore, the proposed system will be an important step forwards for a reasonable cost and accessible epilepsy detection hardware for patients around the globe.

There are a few limitations in this research. The main limitation to this proposed system is the compromise between performance and size of the hardware cost. Furthermore, the testing conducted during this research were mainly performed using EEG benchmark waveforms. Furthermore, it is crucial to address the non-linearity component of the brain when trying to improve this design. Further research would be helpful to fully establish the model as an alternative approach for epilepsy detection hardware and fabrication can be made with an ASIC approach.

## 6.1 Further Work

In this thesis, the optimal BSNN network have been designed and tested. In the near future, it is crucial that the physical layout of the network be developed and optimized further. This design can be implemented as an ASIC model. The balance between size and energy efficiency would need to be explored and the model can then be compared with existing state of the art neural networks.

The main requirements for an optimized epilepsy detection ASIC design would require low power and small in area while still maintaining high recognition rate. A potential ASIC chip can be used in this area. The first one was designed for a subdermal implant. This was to be used as an ASIC detector. This design was fabricated in a TSMC 0.18 micro meter CMOS processor and consumes only an average of 2 micro w of power per EEG channel [137]. Another epilepsy detection hardware [138] was also proposed to be implemented using this same technology. In this case, the chip is used in a proposed smart headband.

As an FPGA based model, it is not practical to be used as a wearable epilepsy detection system. Therefore, it is necessary to develop an ASIC model in order to integrate with a wearable epilepsy detection system to be used by the patient in a mobile environment. At the end of this research, the FPGA synthesis provides the first step in designing a ASIC dedicated hardware neural network. With the synthesis program, it synthesises the RTL level of the hardware neural network design. The ASIC model would be a cell based design are more simplistic as they used already made blocks. The ASIC route would use the technology library to perform gate level synthesis and then proceed with physical library for physical synthesis. The Synopsys software package is used here in this research to complete the task. This thesis design was completed on an FPGA with the end result was on the RTL level. In the future, further minimization can be made to the RTL design and ASIC modelling would be necessary for design physical fabrication.

In the process of optimizing the ASIC design, there are certain exploitations that can be taken into consideration. The place and route procedure in the ASIC design would need to be completed efficiently as it allows smaller chip to be developed. As part of developing a wearable and optimized system, power and size needs to be optimized. Power optimization can involve the process of clock gating.

The RTL level design would need to be optimized as the first step in area optimization. Various unnecessary logic need to be removed before proceeding to gate level synthesis and physical synthesis. Area optimization involves constant propagation, elimination of redundant logics and a 2-level SOP optimization. The critical path of the design should also be improved as the design uses a bit-serial architecture which is much slower compared to state of the art system. This stage involves the process of reducing slack on the critical path.

# Chapter 7

# Publications

1. S. M. Kueh and T. J. Kazmierski, “Massively-parallel bit-serial neural networks for fast epilepsy diagnosis: A feasibility study,” vol. 10, no. 1. World Academy of Science, Engineering and Technology, 2016, pp. 233-237. [Online]. Available: $http : //waset.org/Publications?p = 109$

2. S. M. Kueh and T. Kazmierski, “A dedicated bit-serial hardware neuron for massively-parallel neural networks in fast epilepsy diagnosis,” in *2017 IEEE Healthcare Innovations and Point of Care Technologies (HI-POCT)*, Nov 2017, pp. 105-108.

3. S. M. Kueh and T. J. Kazmierski, “Low-power and low-cost dedicated bit-serial hardware neural network for epileptic seizure prediction system,” *IEEE Journal of Translational Engineering in Health and Medicine*, pp. 1-1, 2018.

4. S. M. Kueh and T. J. Kazmierski, “Low-power and low-cost dedicated bit-serial hardware neural networks for epileptic seizure prediction,” *Proceedings of Small Systems Symposium 2018*, pp. 13-17, February 2018. [Online]. Available: $http : //ssss.elfak.ni.ac.rs/2018/proceedings/$

# References

[1] "Who epilepsy fact sheet," http://www.who.int/mediacentre/factsheets/fs999/en/, accessed: 2018-03-01.

[2] "Who epilepsy fact sheet," https://www.epilepsysociety.org.uk/medication-epilepsy?gclid=CjwKCAjw1ufKBRBYEiwAPI_r4TdJ-tsGZzDrSoIa1bdjthLs583laE9e7ovdIm6w7_rdEKvW8GygIBoCeGgQAvD_BwE#.WVpQbIgrKUl, accessed: 2017-07-03.

[3] D. Gupta, "Advances in epileptic seizure onset prediction in the eeg with ica and phase synchronization," April 2009. [Online]. Available: http://eprints.soton.ac.uk/72166/

[4] B. Litt and J. Echauz, "Prediction of epileptic seizures," *The Lancet Neurology*, vol. 1, no. 1, pp. 22 – 30, 2002.

[5] G. Giannakakis, V. Sakkalis, M. Pediaditis, and M. Tsiknakis, *Methods for Seizure Detection and Prediction: An Overview.* Springer, NY, 2015, pp. 131–157.

[6] P. Rashidi and A. Mihailidis, "A survey on ambient-assisted living tools for older adults," *IEEE Journal of Biomedical and Health Informatics*, vol. 17, no. 3, pp. 579–590, 2013.

[7] A. Dohr, R. Modre-Opsrian, M. Drobics, D. Hayn, and G. Schreier, "The internet of things for ambient assisted living," in *2010 Seventh International Conference on Information Technology: New Generations*, 2010, pp. 804–809.

[8] Kiranmayi, G.R. and Udayashankara, V., "Neural network classifier for the detection of epilepsy," in *Circuits, Controls and Communications (CCUBE), 2013 International conference on*, Dec 2013, pp. 1–4.

[9] E. Painkras, L. Plana, J. Garside, S. Temple, S. Davidson, J. Pepper, D. Clark, C. Patterson, and S. Furber, "Spinnaker: A multi-core system-on-chip for massively-parallel neural net simulation," in *Custom Integrated Circuits Conference (CICC), 2012 IEEE*, Sept 2012, pp. 1–4.

[10] J. Raygoza-Panduro, S. Ortega-Cisneros, and E. Boemo, "Fpga implementation of a synchronous and self-timed neuroprocessor," in *Reconfigurable Computing and*

*FPGAs, 2005. ReConFig 2005. International Conference on*, Sept 2005, pp. 8 pp.–.

[11] S. M. Kueh and T. Kazmierski, “A dedicated bit-serial hardware neuron for massively-parallel neural networks in fast epilepsy diagnosis,” in *2017 IEEE Healthcare Innovations and Point of Care Technologies (HI-POCT)*, Nov 2017, pp. 105–108.

[12] S. M. Kueh and T. J. Kazmierski, “Low-power and low-cost dedicated bit-serial hardware neural network for epileptic seizure prediction system,” *IEEE Journal of Translational Engineering in Health and Medicine*, pp. 1–1, 2018.

[13] ——, “Low-power and low-cost dedicated bit-serial hardware neural networks for epileptic seizure prediction,” *Proceedings of Small Systems Symposium 2018*, pp. 13–17, February 2018. [Online]. Available: http://ssss.elfak.ni.ac.rs/2018/proceedings/

[14] Feng Xie and Zhuangzhi Yan and Shupeng Liu, “Automatic detection of epileptiform discharges in EEG using a back-propagation network,” in *Engineering in Medicine and Biology Society, 2001. Proceedings of the 23rd Annual International Conference of the IEEE*, vol. 2, 2001, pp. 1781–1783 vol.2.

[15] Kiranmayi, G.R. and Udayashankara, V., “Neural network classifier for the detection of epilepsy,” in *Circuits, Controls and Communications (CCUBE), 2013 International conference on*, Dec 2013, pp. 1–4.

[16] Feng Xie and Zhuangzhi Yan and Shupeng Liu, “Automatic detection of epileptiform discharges in EEG using a back-propagation network,” in *Engineering in Medicine and Biology Society, 2001. Proceedings of the 23rd Annual International Conference of the IEEE*, vol. 2, 2001, pp. 1781–1783 vol.2.

[17] Akin, M., e.a., “A new approach for diagnosing epilepsy by using wavelet transform and neural networks,” in *Engineering in Medicine and Biology Society, 2001. Proc. 23rd Annual Int. Conf. of IEEE*, vol. 2, 2001, pp. 1596–1599 vol.2.

[18] Bates, R.R. and Mingui Sun and Scheuer, M.L. and Sclabassi, R.J., “Detection of seizure foci by recurrent neural networks,” in *Engineering in Medicine and Biology Society, 2000. Proceedings of the 22nd Annual International Conference of the IEEE*, vol. 2, 2000, pp. 1377–1379 vol.2.

[19] Tetzlaff, R. and Niederhofer, C. and Fischer, P., “Feature extraction in epilepsy using a cellular neural network based device - first results,” in *Circuits and Systems, 2003. ISCAS ’03. Proceedings of the 2003 International Symposium on*, vol. 3, May 2003, pp. III–850–III–853 vol.3.

[20] Arthur Kleinman, e.a., “The social course of epilepsy: Chronic illness as social experience in interior China,” *Social Science Medicine* , vol. 40, no. 10, pp. 1319 – 1330, 1995.

[21] Ghosh-Dastidar, S. and Adeli, Hojjat and Dadmehr, N., “Mixed-Band Wavelet-Chaos-Neural Network Methodology for Epilepsy and Epileptic Seizure Detection,” *Biomedical Engineering, IEEE Transactions on*, vol. 54, no. 9, pp. 1545–1551, Sept 2007.

[22] Kiranmayi, G.R. and Udayashankara, V., “Neural network classifier for the detection of epilepsy,” in *Circuits, Controls and Communications (CCUBE), 2013 International conference on*, Dec 2013, pp. 1–4.

[23] M. Schneider, P. Mustaro, and C. Lima, “Automatic recognition of epileptic seizure in eeg via support vector machine and dimension fractal,” in *Neural Networks, 2009. IJCNN 2009*, June 2009, pp. 2841–2845.

[24] N. Arunkumar, K. Ramkumar, S. Hema, A. Nithya, P. Prakash, and V. Kirthika, “Fuzzy lyapunov exponent based onset detection of the epileptic seizures,” in *Information Communication Technologies (ICT), 2013 IEEE Conference on*, April 2013, pp. 701–706.

[25] M. T. Rosenstein, J. J. Collins, and C. J. D. Luca, “A practical method for calculating largest lyapunov exponents from small data sets,” *Physica D: Nonlinear Phenomena*, vol. 65, no. 1, pp. 117 – 134, 1993. [Online]. Available: http://www.sciencedirect.com/science/article/pii/016727899390009P

[26] Z. Haydari, Y. Zhang, and H. Soltanian-Zadeh, “Semi-automatic epilepsy spike detection from eeg signal using genetic algorithm and wavelet transform,” in *Bioinformatics and Biomedicine Workshops (BIBMW), 2011 IEEE International Conference on*, Nov 2011, pp. 635–638.

[27] D. Rivero, J. Dorado, J. Rabunal, and A. Pazos, “Evolving simple feed-forward and recurrent anns for signal classification: A comparison,” in *Neural Networks, 2009. IJCNN 2009. International Joint Conference on*, June 2009, pp. 2685–2692.

[28] R. Sarang, “A strong adaptive and comprehensive evaluation of wavelet based epileptic eeg spike detection methods,” in *Biomedical and Pharmaceutical Engineering, 2006. ICBPE 2006. International Conference on*, Dec 2006, pp. 432–437.

[29] S. Mousavi, M. Niknazar, and B. Vahdat, “Epileptic seizure detection using ar model on eeg signals,” in *Biomedical Engineering Conference, 2008. CIBEC 2008. Cairo International*, Dec 2008, pp. 1–4.

[30] B. Yu, T. Mak, X. Li, F. Xia, A. Yakovlev, Y. Sun, and C. S. Poon, “A stream-based hebbian eigenfilter for real-time neurophysiological signal processing,” in

*2010 Biomedical Circuits and Systems Conference (BioCAS)*, Nov 2010, pp. 90–93.

[31] S. T. Chakradhar, M. L. Bushnell, and V. D. Agrawal, “Automatic test generation using neural networks,” in *[1988] IEEE International Conference on Computer-Aided Design (ICCAD-89) Digest of Technical Papers*, Nov 1988, pp. 416–419.

[32] M. Arai, T. Nakagawa, and H. Kitagawa, “An approach to automatic test pattern generation using strictly digital neural networks,” in *Neural Networks, 1992. IJCNN., International Joint Conference on*, vol. 4, Jun 1992, pp. 474–479 vol.4.

[33] Y. Yang, C. S. Boling, A. M. Kamboh, and A. J. Mason, “Adaptive threshold neural spike detector using stationary wavelet transform in cmos,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 23, no. 6, pp. 946–955, Nov 2015.

[34] A. T. Tzallas, M. G. Tsipouras, and D. I. Fotiadis, “Epileptic seizure detection in eegs using time x2013;frequency analysis,” *IEEE Transactions on Information Technology in Biomedicine*, vol. 13, no. 5, pp. 703–710, Sept 2009.

[35] A. Sharmila and P. Geethanjali, “Dwt based detection of epileptic seizure from eeg signals using naive bayes and k-nn classifiers,” *IEEE Access*, vol. 4, pp. 7716–7727, 2016.

[36] “Eeg time series download page,” http://epileptologie-bonn.de/cms/front_content.php?idcat=193&lang=3&changelang=3, accessed: 2018-03-15.

[37] S. R. Safavian and D. Landgrebe, “A survey of decision tree classifier methodology,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 21, no. 3, pp. 660–674, May 1991.

[38] R. Coggins, M. Jabri, B. Flower, and S. Pickard, “A hybrid analog and digital vlsi neural network for intracardiac morphology classification,” *IEEE Journal of Solid-State Circuits*, vol. 30, no. 5, pp. 542–550, May 1995.

[39] L. Arbach, J. M. Reinhardt, D. L. Bennett, and G. Fallouh, “Mammographic masses classification: comparison between backpropagation neural network (bnn), k nearest neighbors (knn), and human readers,” in *CCECE 2003 - Canadian Conference on Electrical and Computer Engineering. Toward a Caring and Humane Technology (Cat. No.03CH37436)*, vol. 3, May 2003, pp. 1441–1444 vol.3.

[40] “Chapter 4: K nearest neighbors classifier,” https://medium.com/machine-learning-101/k-nearest-neighbors-classifier-1c1ff404d265, accessed: 2019-02-18.

[41] T. P. Hong and S. S. Tseng, “Models of parallel learning systems,” in *Distributed Computing Systems, 1991., 11th International Conference on*, May 1991, pp. 125–132.

[42] Y. Jewajinda and P. Chongstitvatana, "Fpga-based online-learning using parallel genetic algorithm and neural network for ecg signal classification," in *Electrical Engineering/Electronics Computer Telecommunications and Information Technology (ECTI-CON), 2010 International Conference on*, May 2010, pp. 1050–1054.

[43] L. Vokorokos, N. Adam, and J. Trelova, "Algorithmic mapping of mlp network on neural df kpi architecture," in *Computational Cybernetics, 2006. ICCC 2006. IEEE International Conference on*, Aug 2006, pp. 1–6.

[44] M. Zhenhui, "Research on serial grey neural network mode," in *Measurement, Information and Control (MIC), 2012 International Conference on*, vol. 2, May 2012, pp. 980–983.

[45] C. Liu, T. Shu, S. Chen, S. Wang, K. K. Lai, and L. Gan, "An improved grey neural network model for predicting transportation disruptions," *Expert Systems with Applications*, vol. 45, pp. 331 – 340, 2016.

[46] S. R. Shahamiri and S. S. B. Salim, "A multi-views multi-learners approach towards dysarthric speech recognition using multi-nets artificial neural networks," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 22, no. 5, pp. 1053–1063, Sept 2014.

[47] Akin, M. and Arserim, M. A. and Kiymik, M.K. and Turkoglu, I., "A new approach for diagnosing epilepsy by using wavelet transform and neural networks," in *Engineering in Medicine and Biology Society, 2001. Proceedings of the 23rd Annual International Conference of the IEEE*, vol. 2, 2001, pp. 1596–1599 vol.2.

[48] Arthur Kleinman and Wen-Zhi Wang and Shi-Chuo Li and Xue-Ming Cheng and Xiu-Ying Dai and Kun-Tun Li and Joan Kleinman, "The social course of epilepsy: Chronic illness as social experience in interior China," *Social Science Medicine* , vol. 40, no. 10, pp. 1319 – 1330, 1995.

[49] Kharat, P. A. and Dudul, S.V., "Clinical decision support system based on Jordan/Elman neural networks," in *Recent Advances in Intelligent Computational Systems (RAICS), 2011 IEEE*, Sept 2011, pp. 255–259.

[50] Shukla, A. and Tiwari, R. and Kaur, P., "Intelligent System for the Diagnosis of Epilepsy," in *Computer Science and Information Engineering, 2009 WRI World Congress on*, vol. 5, March 2009, pp. 755–758.

[51] Bao, F.S. and Lie, D.Y.-C. and Yuanlin Zhang, "A New Approach to Automated Epileptic Diagnosis Using EEG and Probabilistic Neural Network," in *Tools with Artificial Intelligence, 2008. ICTAI '08. 20th IEEE International Conference on*, vol. 2, Nov 2008, pp. 482–486.

[52] S. Bezobrazova and V. Golovko, “Comparative analysis of forecasting neural networks in the application for epilepsy detection,” in *Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, 2007. IDAACS 2007. 4th IEEE Workshop on*, Sept 2007, pp. 202–206.

[53] Bates, R.R. and Mingui Sun and Scheuer, M.L. and Sclabassi, R.J., “Detection of seizure foci by recurrent neural networks,” in *Engineering in Medicine and Biology Society, 2000. Proceedings of the 22nd Annual International Conference of the IEEE*, vol. 2, 2000, pp. 1377–1379 vol.2.

[54] Srinivasan, V. and Eswaran, C. and Sriraam, N., “Approximate Entropy-Based Epileptic EEG Detection Using Artificial Neural Networks,” *Information Technology in Biomedicine, IEEE Transactions on*, vol. 11, no. 3, pp. 288–295, May 2007.

[55] Pincus, StevenM. and Gladstone, IgorM. and Ehrenkranz, RichardA., “A regularity statistic for medical data analysis,” *Journal of Clinical Monitoring*, vol. 7, no. 4, 1991. [Online]. Available: http://dx.doi.org/10.1007/BF01619355

[56] Ye Yuan and Yue Li and Dongyan Yu and Mandic, D.P., “Delay Time-Based Epileptic EEG Detection Using Artificial Neural Network,” in *Bioinformatics and Biomedical Engineering, 2008. ICBBE 2008. The 2nd International Conference on*, May 2008, pp. 502–505.

[57] Bao, F.S. and Jue-Ming Gao and Jing Hu and Lie, D.Y.-C. and Yuanlin Zhang and Oommen, K. J., “Automated epilepsy diagnosis using interictal scalp EEG,” in *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, Sept 2009, pp. 6603–6607.

[58] Donald F. Specht, “Probabilistic neural networks ,” *Neural Networks* , vol. 3, no. 1, pp. 109 – 118, 1990.

[59] V. Srinivasan, C. Eswaran, and N. Sriraam, “Epileptic detection using artificial neural networks,” in *Signal Processing and Communications, 2004. SPCOM ’04. 2004 International Conference on*, Dec 2004, pp. 340–343.

[60] Andrzejak, Ralph G., e.a., “Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state,” *Phys. Rev. E*, vol. 64, p. 061907, Nov 2001.

[61] A. Arista-Jalife and R. Vazquez, “Implementation of configurable and multipurpose spiking neural networks on gpus,” in *Neural Networks (IJCNN), The 2012 International Joint Conference on*, June 2012, pp. 1–8.

[62] T. Matsumoto, Y. Shin, H. Takase, H. Kawanaka, and S. Tsuruoka, “A learning method for extended spikeprop without redundant spikes x2014; automatic adjustment of hidden units,” in *Soft Computing and Intelligent Systems (SCIS), 2014*

*Joint 7th International Conference on and Advanced Intelligent Systems (ISIS), 15th International Symposium on*, Dec 2014, pp. 1465–1469.

[63] H. Fang, Y. Wang, and J. He, "Spiking neural networks for cortical neuronal decoding," *Neural Computation*, vol. 22, no. 4, pp. 1060–1085, April 2010.

[64] Izhikevich, E.M., "Simple model of spiking neurons," *Neural Networks, IEEE Transactions on*, vol. 14, no. 6, pp. 1569–1572, Nov 2003.

[65] Cheng-Wen Ko, Hsiao-Wen Chung , "Automatic spike detection via an artificial neural network using raw EEG data: effects of data preparation and implications in the limitations of online recognition," *Clinical Neurophysiology - 1*, vol. 111, no. 3, pp. 477–481, March 2000.

[66] S. Li, Z.-Q. Liu, and A. Chan, "Heterogeneous multi-task learning for human pose estimation with deep convolutional neural network," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*, June 2014, pp. 488–495.

[67] S. Thomas, S. Ganapathy, G. Saon, and H. Soltau, "Analyzing convolutional neural networks for speech activity detection in mismatched acoustic conditions," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, May 2014, pp. 2519–2523.

[68] S. Zhang, Y. Bao, P. Zhou, H. Jiang, and L. Dai, "Improving deep neural networks for lvcsr using dropout and shrinking structure," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 6849–6853.

[69] S. Brassai, L. Bako, G. Pana, and S. Dan, "Neural control based on rbf network implemented on fpga," in *Optimization of Electrical and Electronic Equipment, 2008. OPTIM 2008. 11th International Conference on*, May 2008, pp. 41–46.

[70] M. Krid, D. S. Masmoudi, and M. Chtourou, "Hardware implementation of bfnn and rbfnn in fpga technology: Quantization issues," in *Electronics, Circuits and Systems, 2005. ICECS 2005. 12th IEEE International Conference on*, Dec 2005, pp. 1–4.

[71] T. Wang, H. Wang, and X. Hao-fei, "Networked synchronization control method by the combination of rbf neural network and genetic algorithm," in *Computer and Automation Engineering (ICCAE), 2010 The 2nd International Conference on*, vol. 3, Feb 2010, pp. 9–12.

[72] D. Zhang, H. Li, and S. Foo, "A simplified fpga implementation of neural network algorithms integrated with stochastic theory for power electronics applications," in *Industrial Electronics Society, 2005. IECON 2005. 31st Annual Conference of IEEE*, Nov 2005, pp. 6 pp.–.

[73] Y. Ago, A. Inoue, K. Nakano, and Y. Ito, “The parallel fdfm processor core approach for neural networks,” in *Networking and Computing (ICNC), 2011 Second International Conference on*, Nov 2011, pp. 113–119.

[74] M. Norouzi, M. Ranjbar, and G. Mori, “Stacks of convolutional restricted boltzmann machines for shift-invariant feature learning,” in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, June 2009, pp. 2735–2742.

[75] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov 1998.

[76] M. Freeman and J. Austin, “Designing a binary neural network co-processor,” in *Digital System Design, 2005. Proceedings. 8th Euromicro Conference on*, Aug 2005, pp. 223–226.

[77] D. Roy Chowdhury, I. Gupta, and P. Pal Chaudhuri, “A low-cost high-capacity associative memory design using cellular automata,” *Computers, IEEE Transactions on*, vol. 44, no. 10, pp. 1260–1264, Oct 1995.

[78] C. Ang, C. Jin, P. Leong, and A. van Schaik, “Spiking neural network-based auto-associative memory using fpga interconnect delays,” in *Field-Programmable Technology (FPT), 2011 International Conference on*, Dec 2011, pp. 1–4.

[79] A. van Schaik, “Building blocks for electronic spiking neural networks,” *Neural Networks*, vol. 14, no. 67, pp. 617 – 628, 2001. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0893608001000673

[80] S. Merchant and G. Peterson, “An evolvable artificial neural network platform for dynamic environments,” in *Circuits and Systems, 2008. MWSCAS 2008. 51st Midwest Symposium on*, Aug 2008, pp. 77–80.

[81] G. Baines, “Neural networks for boiler emission prediction,” in *Instrumentation and Measurement Technology Conference, 1999. IMTC/99. Proceedings of the 16th IEEE*, vol. 1, 1999, pp. 435–439 vol.1.

[82] R. K. Weinstein, M. S. Reid, and R. H. Lee, “Methodology and design flow for assisted neural-model implementations in fpgas,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 15, no. 1, pp. 83–93, March 2007.

[83] C. Qi, Y. Chen, and T. Huang, “Realization of neural network controller for electromotor frequency conversion speed-regulating based on lonworks,” in *Cellular Neural Networks and Their Applications, 2005 9th International Workshop on*, May 2005, pp. 31–35.

[84] M. Khan, D. Lester, L. Plana, A. Rast, X. Jin, E. Painkras, and S. Furber, “Spinnaker: Mapping neural networks onto a massively-parallel chip multiprocessor,” in *Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*, June 2008, pp. 2849–2856.

[85] X. Jin, F. Galluppi, C. Patterson, A. Rast, S. Davies, S. Temple, and S. Furber, “Algorithm and software for simulation of spiking neural networks on the multi-chip spinnaker system,” in *Neural Networks (IJCNN), The 2010 International Joint Conference on*, July 2010, pp. 1–8.

[86] P. Diehl and M. Cook, “Efficient implementation of stdp rules on spinnaker neuromorphic hardware,” in *Neural Networks (IJCNN), 2014 International Joint Conference on*, July 2014, pp. 4288–4295.

[87] X. Jin, M. Lujan, L. Plana, S. Davies, S. Temple, and S. Furber, “Modeling spiking neural networks on spinnaker,” *Computing in Science Engineering*, vol. 12, no. 5, pp. 91–97, Sept 2010.

[88] Izhikevich, E.M., “Simple model of spiking neurons,” *Neural Networks, IEEE Transactions on*, vol. 14, no. 6, pp. 1569–1572, Nov 2003.

[89] E. Painkras, L. Plana, J. Garside, S. Temple, F. Galluppi, C. Patterson, D. Lester, A. Brown, and S. Furber, “Spinnaker: A 1-w 18-core system-on-chip for massively-parallel neural network simulation,” *Solid-State Circuits, IEEE Journal of*, vol. 48, no. 8, pp. 1943–1953, Aug 2013.

[90] E. Stromatias, F. Galluppi, C. Patterson, and S. Furber, “Power analysis of large-scale, real-time neural networks on spinnaker,” in *Neural Networks (IJCNN), The 2013 International Joint Conference on*, Aug 2013, pp. 1–8.

[91] A. Rast, F. Galluppi, S. Davies, L. A. Plana, T. Sharp, and S. Furber, “An event-driven model for the spinnaker virtual synaptic channel,” in *Neural Networks (IJCNN), The 2011 International Joint Conference on*, July 2011, pp. 1967–1974.

[92] H. Shawkey, H. Elsimary, H. Haddara, and H. F. Ragaie, “Design of a vlsi neural network arrhythmia classifier,” in *Radio Science Conference, 1999. NRSC ’99. Proceedings of the Sixteenth National*, Feb 1999, pp. C27/1–C2710.

[93] J. Dragas, D. Jckel, A. Hierlemann, and F. Franke, “Complexity optimization and high-throughput low-latency hardware implementation of a multi-electrode spike-sorting algorithm,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 23, no. 2, pp. 149–158, March 2015.

[94] H. Soliman, H. Wang, B. Gadalla, and F. Blaabjerg, “Condition monitoring for dc-link capacitors based on artificial neural network algorithm,” in *2015 IEEE 5th International Conference on Power Engineering, Energy and Electrical Drives (POWERENG)*, May 2015, pp. 587–591.

[95] Q. Huang, S. Chang, J. Peng, X. Mao, Y. Zhou, and H. Wang, "An implementation of sopc-based neural monitoring system," *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 9, pp. 2469–2475, Sept 2012.

[96] B. Yu, T. Mak, L. Smith, Y. Sun, A. Yakovlev, and C. S. Poon, "Memory efficient on-line streaming for multichannel spike train analysis," in *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Aug 2011, pp. 2315–2318.

[97] M. Saleheen, H. Alemzadeh, A. Cheriyan, Z. Kalbarczyk, and R. Iyer, "An efficient embedded hardware for high accuracy detection of epileptic seizures," in *Biomedical Engineering and Informatics (BMEI), 2010 3rd International Conference on*, vol. 5, Oct 2010, pp. 1889–1896.

[98] T. Ahola, P. Korpinen, J. Rakkola, T. Ramo, J. Salminen, and J. Savolainen, "Wearable fpga based wireless sensor platform," in *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, Aug 2007, pp. 2288–2291.

[99] X. Liu, M. Zhang, B. Subei, A. G. Richardson, T. H. Lucas, and J. V. der Spiegel, "The pennbmbi: Design of a general purpose wireless brain-machine-brain interface system," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 9, no. 2, pp. 248–258, April 2015.

[100] X. Liu, B. Subei, M. Zhang, A. G. Richardson, T. H. Lucas, and J. V. der Spiegel, "The pennbmbi: A general purpose wireless brain-machine-brain interface system for unrestrained animals," in *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*, June 2014, pp. 650–653.

[101] M. Neschen, "COLUMNUS - an SIMD architecture for pattern recognition and simulations of statistical physics," in *Proc. Int. Conf. on Application-Specific Array Processors, 1993.*, Oct 1993, pp. 168–171.

[102] J. Lofgren and P. Nilsson, "Bit-serial cordic: Architecture and implementation improvements," in *Circuits and Systems (MWSCAS), 2010 53rd IEEE International Midwest Symposium on*, Aug 2010, pp. 65–68.

[103] Y. Chen and W. du Plessis, "Neural network implementation on a fpga," in *Africon Conference in Africa, 2002. IEEE AFRICON. 6th*, vol. 1, Oct 2002, pp. 337–342 vol.1.

[104] T. Yamamoto and V. Moshnyaga, "A new bit-serial architecture of rank-order filter," in *Circuits and Systems, 2009. MWSCAS '09. 52nd IEEE International Midwest Symposium on*, Aug 2009, pp. 511–514.

[105] and, "Partitioning and mapping algorithms into fixed size systolic arrays," *IEEE Transactions on Computers*, vol. C-35, no. 1, pp. 1–12, Jan 1986.

[106] P. Judd, J. Albericio, and A. Moshovos, “Stripes: Bit-serial deep neural network computing,” *IEEE Computer Architecture Letters*, vol. 16, no. 1, pp. 80–83, Jan 2017.

[107] T. Luo, S. Liu, L. Li, Y. Wang, S. Zhang, T. Chen, Z. Xu, O. Temam, and Y. Chen, “Dadiannao: A neural network supercomputer,” *IEEE Transactions on Computers*, vol. 66, no. 1, pp. 73–88, 2017.

[108] C. Eckert, X. Wang, J. Wang, A. Subramaniyan, R. Iyer, D. Sylvester, D. Blaaauw, and R. Das, “Neural cache: Bit-serial in-cache acceleration of deep neural networks,” in *2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA)*, June 2018, pp. 383–396.

[109] N. R. Strader and V. T. Rhyne, “A canonical bit-sequential multiplier,” *IEEE Transactions on Computers*, vol. C-31, no. 8, pp. 791–795, Aug 1982.

[110] T. Szabo, L. Antoni, G. Horvath, and B. Feher, “A full-parallel digital implementation for pre-trained nns,” in *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, vol. 2, July 2000, pp. 49–54 vol.2.

[111] S. Akhter and S. Chaturvedi, “Hdl based implementation of n x00d7;n bit-serial multiplier,” in *Signal Processing and Integrated Networks (SPIN), 2014 International Conference on*, Feb 2014, pp. 470–474.

[112] R. Gnanasekaran, “A fast serial-parallel binary multiplier,” *Computers, IEEE Transactions on*, vol. C-34, no. 8, pp. 741–744, Aug 1985.

[113] A. Shafer, L. Parker, and E. Swartzlander, “The fully-serial pipelined multiplier,” in *Signals, Systems and Computers (ASILOMAR), 2011 Conference Record of the Forty Fifth Asilomar Conference on*, Nov 2011, pp. 1817–1822.

[114] J. Li, Y. Du, and J. Wang, “Design a pocket multi-bit multiplier in fpga,” in *ASIC, 1996., 2nd International Conference on*, Oct 1996, pp. 275–279.

[115] B. K. Mohanty and P. Meher, “Bit-serial systolic architecture for 2-d non-separable discrete wavelet transform,” in *Intelligent and Advanced Systems, 2007. ICIAS 2007. International Conference on*, Nov 2007, pp. 1355–1358.

[116] D. Sukumaran, Y. Enyi, S. Shuo, A. Basu, D. Zhao, and J. Dauwels, “A low-power, reconfigurable smart sensor system for eeg acquisition and classification,” in *Circuits and Systems (APCCAS), 2012 IEEE Asia Pacific Conference on*, Dec 2012, pp. 9–12.

[117] Y. Wang, S. Wang, and K. K. Lai, “A new fuzzy support vector machine to evaluate credit risk,” *IEEE Transactions on Fuzzy Systems*, vol. 13, no. 6, pp. 820–831, Dec 2005.

[118] K. Xia, G. Xu, and N. Xu, “Lung cancer diagnosis system based on support vector machines and image processing technique,” in *Intelligent Information Hiding and Multimedia Signal Processing, 2006. IIH-MSP ’06. International Conference on*, Dec 2006, pp. 143–146.

[119] A. Ranjan, A. Raha, S. Venkataramani, K. Roy, and A. Raghunathan, “Aslan: Synthesis of approximate sequential circuits,” in *2014 Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2014, pp. 1–6.

[120] J. Lehrner, R. Kalchmayr, W. Serles, A. Olbrich, E. Pataraia, S. Aull, J. Bacher, F. Leutmezer, G. Grppel, L. Deecke, and C. Baumgartner, “Health-related quality of life (hrqol), activity of daily living (adl) and depressive mood disorder in temporal lobe epilepsy patients,” *Seizure*, vol. 8, no. 2, pp. 88 – 92, 1999. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1059131199902728

[121] J. A. Cramer, D. Blum, M. Reed, and K. Fanning, “The influence of comorbid depression on quality of life for people with epilepsy,” *Epilepsy Behavior*, vol. 4, no. 5, pp. 515 – 521, 2003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1525505003001914

[122] J. Ko, C. Lu, M. Srivastava, J. Stankovic, A. Terzis, and M. Welsh, “Wireless sensor networks for healthcare,” *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1947–1960, Nov 2010.

[123] A. Al-armaghany, B. Yu, T. Mak, K. F. Tong, and Y. Sun, “Feasibility study for future implantable neural-silicon interface devices,” in *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Aug 2011, pp. 3009–3015.

[124] A. S. Al-Fahoum and A. A. Al-Fraihat, “Methods of eeg signal features extraction using linear analysis in frequency and time-frequency domains,” *ISRN neuroscience*, vol. 2014, 2014.

[125] H. Peng, B. Hu, F. Zheng, D. Fan, W. Zhao, X. Chen, Y. Yang, and Q. Cai, “A method of identifying chronic stress by eeg,” *Personal Ubiquitous Comput.*, vol. 17, no. 7, pp. 1341–1347, Oct. 2013. [Online]. Available: http://dx.doi.org/10.1007/s00779-012-0593-3

[126] M. T. Salam, J. L. Perez-Velazquez, and R. Genov, “Seizure suppression efficacy of closed-loop versus open-loop deep brain stimulation in a rodent model of epilepsy,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. PP, no. 99, pp. 1–1, 2015.

[127] M. Ueda, Y. Nishitani, Y. Kaneko, and A. Omote, “Back-propagation operation for analog neural network hardware with synapse components having hysteresis

characteristics," *PLOS ONE*, vol. 9, no. 11, pp. 1–10, 11 2014. [Online]. Available: https://doi.org/10.1371/journal.pone.0112659

[128] S. M. Kueh and T. J. Kazmierski, "Massively-parallel bit-serial neural networks for fast epilepsy diagnosis: A feasibility study," vol. 10, no. 1. World Academy of Science, Engineering and Technology, 2016, pp. 233 – 237. [Online]. Available: http://waset.org/Publications?p=109

[129] B. Svensson and T. Nordstrom, "Execution of neural network algorithms on an array of bit-serial processors," in *Proc.10th Int. Conf. on Pattern Recognition*, vol. ii, June 1990, pp. 501–505 vol.2.

[130] D. Walsh and P. Dudek, "A compact fpga implementation of a bit-serial simd cellular processor array," in *2012 13th International Workshop on Cellular Nanoscale Networks and their Applications*, Aug 2012, pp. 1–6.

[131] "Fpga logic cells comparison," http://ee.sharif.edu/~asic/Docs/fpga-logic-cells_V4_V5.pdf, accessed: 2019-09-16.

[132] H. J. Carey, M. Manic, and P. Arsenovic, "Epileptic spike detection with eeg using artificial neural networks," in *2016 9th International Conference on Human System Interactions (HSI)*, July 2016, pp. 89–95.

[133] R. G. Andrzejak, K. Lehnertz, F. Mormann, C. Rieke, P. David, and C. E. Elger, "Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state," *Phys. Rev. E*, vol. 64, p. 061907, Nov 2001. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevE.64.061907

[134] E. Kabir, Y. Zhang *et al.*, "Epileptic seizure detection from eeg signals using logistic model trees," *Brain informatics*, vol. 3, no. 2, pp. 93–100, 2016.

[135] Wang L., e.a., "Automatic epileptic seizure detection in EEG signals using multi-domain feature extraction and nonlinear analysis," *Entropy*, vol. 19, no. 6, 2017.

[136] N. Moghim and D. W. Corne, "Predicting epileptic seizures in advance," *PLOS ONE*, vol. 9, no. 6, pp. 1–17, 06 2014. [Online]. Available: https://doi.org/10.1371/journal.pone.0099334

[137] B. G. Do Valle, S. S. Cash, and C. G. Sodini, "Low-power, 8-channel eeg recorder and seizure detector asic for a subdermal implantable system," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 10, no. 6, pp. 1058–1067, Dec 2016.

[138] S. Lin, Y. Lin, C. Lin, and H. Chiueh, "A smart headband for epileptic seizure detection," in *2017 IEEE Healthcare Innovations and Point of Care Technologies (HI-POCT)*, Nov 2017, pp. 221–224.