Department of Electrical Engineering

University of Sydney

Arrhythmia Classification using Low Power VLSI

Philip H. W. Leong B.Sc., B.E.

A thesis submitted in fulfilment

of the requirements for the degree of

Doctor of Philosophy

December 1992

To my parents, in appreciation of their sacrifices.

Abstract

The implantable cardioverter–defibrillator (ICD) is a device which monitors the heart and delivers defibrillation and/or pacing therapy in the event of a life threatening arrhythmia. Such devices have had great success in preventing what otherwise would be sudden cardiac death. The arrhythmia classifier used in current devices makes its decision primarily based on rate criteria. Unfortunately, this method is very crude and can cause incorrect therapy to be delivered for certain arrhythmias, sometimes with fatal consequences.

This thesis describes a two chamber arrhythmia classification algorithm, called MATIC, which classifies heart signals based on timing and morphological criteria. MATIC attempts to mimic a cardiologist when making a classification and uses a decision tree to examine timing relations within and between channels. For most arrhythmias, this information is sufficient to make a reliable classification. For some arrhythmias, correct classification cannot be achieved based on rate alone, and in these cases, a neural network is used to analyze the morphology of the heart signal. The MATIC algorithm achieved 99.6% correct classification on a database of intracardiac electrogram signals containing 12483 QRS complexes recorded from 67 patients during electrophysiological studies.

The neural network used for morphology classification in the MATIC algorithm is computationally expensive and would have excessive power consumption for an implantable device. An analogue low power neural network VLSI chip called Kakadu was designed to address this problem. Kakadu's design is based on synapses implemented as multiplying digital to analogue converters which can

operate at very low bias currents. Kakadu implements a three layer feedforward network with 10 inputs, 6 hidden units and 4 outputs and achieves typical power consumption figures of tens of microwatts. When Kakadu is incorporated in an arrhythmia classification system, power consumption of less than 25 nW can be achieved.

When Kakadu is used to provide the morphology classification of the MATIC classifier, the resulting system classifies arrhythmias both reliably and with low power consumption – the two necessary features of a classifier suitable for an ICD.

Acknowledgements

I would firstly like to acknowledge the support of my supervisor, Dr. Marwan Jabri for his role in this thesis: the idea of developing a low power arrhythmia classifier, organisation of funds for this project, many ideas and inspiration, untiring enthusiasm and long patient hours of listening to me complain about this and that. For all these things, I warmly thank him.

I thank Ross Halgren, my former boss at AWA Research Laboratory, who gave me the opportunity to start on my thesis.

All the work in this thesis was done whilst I was employed on the ICEG project at the Systems Engineering and Design Automation Laboratory (SEDAL), University of Sydney. The ICEG project was supported by the Australian Department of Industry, Technology and Commerce and Telectronics Pacing Systems. I am especially greatful to Dr. Peter Nickolls of Telectronics Pacing Systems who supplied all the ICEG data used in this thesis and gave me the opportunity to use the facilities at Telectronics. Without him, this thesis would not have been possible.

I would also like to thank all my coworkers at the University of Sydney, in particular, Dr. Peter Henderson, Barry Flower, Edward Tinker, Albert Hirawan, Jeevan Vasimalla, Zheru Chi and Agatha Shotam for their daily help, ideas, support and encouragement; and especially Dr. Stephen Pickard, Mark Hedley and Richard Coggins for reviewing my manuscript.

Finally, I thank my parents. My mother Norma Leong for her lifelong chore of bringing me up, her warmth, generosity, encouragement and sacrifices; and my

father Harry Leong, a kind and brilliant engineer who died in 1979 from sudden cardiac death. This thesis is dedicated to both of them.

Statement of Originality

The work described in this thesis was carried out at the Department of Electrical Engineering, University of Sydney, between 1989 and 1992, under the supervision of Dr. Marwan Jabri.

The work in this thesis is entirely original except where duly referenced. In particular

1. The idea of using neural networks for arrhythmia classification in ICDs and implementing them in low power was that of Dr. Marwan Jabri.

2. The MATIC algorithm is the work of the author.

3. The Bourke chip architecture, design, simulation, layout and testing is the work of the author.

4. The combined search algorithm (CSA) is the work of Yun Xie. CSA, in turn, uses weight perturbation which is the work of Dr. Marwan Jabri.

5. The hierarchical training strategy used to train Bourke on XOR, and Kakadu on the parity 4 problem, is the work of the author.

6. The Kakadu chip architecture, design, layout and testing is the work of the author.

7. The Jiggle chip tester was developed jointly by the author and Jeevan Vasimalla.

8. The QRS detector and ICEG data used to test the MATIC algorithm were supplied by Telectronics Pacing Systems.

9. The system architecture for implementing MATIC is the work of the author.

10. The simple threshold detector algorithm was derived from the Telectronics Guardian ATP 4210 Clinician's Information manual [Ltd89].

11. All software used for simulation, modeling, testing and training in this thesis was written by the author unless otherwise stated.

The material in this thesis has not been submitted for any other degree at this or any other institution.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation and Aims

Sudden cardiac death is a major public health problem worldwide, claiming an estimated 450,000 lives in the USA every year, between 80% and 90% of these being due to ventricular tachyarrhythmias [Gil89]. In February 1980, the first human implant of the implantable cardioverter–defibrillator (ICD) was performed and early results were encouraging. Since 1980 more than 10,000 patients have received ICD devices, and no other therapy has been able to prevent sudden cardiac death so effectively [Tro91].

The ICD, invented by Michael Mirowski, is an electronic device which is permanently implanted in a patient. This device monitors the heart and applies to it a high energy defibrillating shock when a life threatening tachyarrhythmia is detected.

Despite the spectacular acceptance and performance of these devices throughout the 1980s and early 1990s, they are not perfect. Arrhythmia classification in current ICDs is performed by heart rate alone or heart rate combined with simple morphological criteria. Although these techniques are successful

in differentiating between certain tachyarrhythmias, ICDs often produce incorrect classifications. This can lead to the delivery of inappropriate therapy [FLS84, Win91, MMK+91, BGRL+92] – sometimes with fatal consequences.

As ICD devices must be battery powered and have a lifetime of up to 10 years, low power consumption is a very important constraint in their design. Although many excellent algorithms for tachyarrhythmia classification already exist, they usually need a large amount of computing power and hence are not suitable for use in implantable devices.

The main aim of this thesis is to develop a system for classifying tachyarrhythmias which is both accurate and draws little power. Even without a power limit, classification is a difficult problem due to a large variation and overlap in the distinguishing features of tachyarrhythmias between patients. This problem is made even more difficult when constrained by a very low power budget.

1.2 Contributions

This thesis presents a new algorithm for tachyarrhythmia classification which was developed to accurately classify a wide range of tachyarrhythmias. The algorithm classifies intracardiac signals in much the same way as a human expert: a decision tree is used to examine timing relations in the signal and an artificial neural network (ANN) is used to recognise certain morphological features. Neural networks have an advantage over conventional techniques in that they can produce nonlinear mappings as well as store multiple morphologies. The algorithm thus combines the simplicity and efficiency of a decision tree with the improved pattern matching ability of a neural network. The decision tree is computationally inexpensive and can be implemented in the microprocessor of an ICD. However, the ANN is computationally expensive, and so a new analogue very large scale integration (VLSI) architecture and implementation was developed to achieve the very low power consumption required.

The algorithm developed in this thesis is called MATIC (morphology and timing intracardiac electrogram classifier) and can classify a wide range of tachyarrhythmias which commonly occur in candidates for ICD devices. MATIC differs from earlier work [PF88, TDNC91, ZT88, DBW90, LDJ88, TJD90a, CJ90, Lee89, ABJ+84, MF86, SJSL88] in that it was designed with low power implementation in mind. It also produces a classification based on both timing and morphology, thus improving the ability to recognise a wider range of tachyarrhythmias. In particular, it directly addresses the problem of differentiating between sinus tachycardia, supraventricular tachycardias, ventricular tachycardias and ventricular fibrillation. The number of patients tested on MATIC and the classification performance achieved by MATIC surpasses those in all the reports cited above.

The implementation of the MATIC algorithm in an ICD relies on being able to implement a morphology based classifier that has very low power consumption. To achieve this, an architecture for a general purpose low power VLSI ANN chip (called Kakadu) was developed. Although many subthreshold VLSI designs inspired by biological neural networks have been made [Mea89, AB89, And90], no comparable implementations with as low a power consumption as Kakadu have been reported.

An implementation of the MATIC classifier using the Kakadu chip, which allows classifier performance and power consumption to be measured, was built and successfully tested. Although many proposals for using VLSI neural networks to solve real world problems have been made, few have produced a working implementation of an entire system. None has been applied to the tachyarrhythmia classification problem.

Many morphology classifiers have been proposed for ICDs [ABJ+84, LJDM88, LDJ88, TJD90a, TDJW90, TJD90b, DTJ91]. In the past, ICDs have not incorporated morphology classifiers because the area and power consumption required for their implementation is too great for the small population of patients which

it might benefit. However, the neural network VLSI chip developed in this thesis would not increase the power consumption of an ICD by an appreciable amount and can fit into a small area, thus making it tractable to incorporate it in an ICD at little cost. The addition of morphology classification to ICD devices would enable a wider range of patients to benefit from ICD technology.

1.3 Structure of the Thesis

An introduction to the fundamentals of cardiology, arrhythmia classification and ICD devices is presented in Chapter 2. The chapter begins with an introduction to the physiology of the heart, shows mechanisms which cause tachyarrhythmias, describes the diagnosis of tachyarrhythmias, reviews previous work on automatic arrhythmia classification and ends with a short description of a modern ICD.

In Chapter 3, the design of an algorithm for classifying tachyarrhythmias is presented in detail. The chapter describes the design considerations for the classifier and then presents the MATIC algorithm in detail. The MATIC algorithm uses a neural network for template matching and a decision tree to perform timing based classification. It is shown that the implementation of the neural network using a microprocessor inside an ICD would have too high a power consumption. For this reason, a low power VLSI implementation of the neural network is proposed and an architecture appropriate for implementing the MATIC algorithm in an ICD is described.

Chapter 4 covers the design, implementation and testing of the first of two artificial neural network chips. It begins with a review of previous work on implementing ANNs in analogue VLSI. Design considerations for a neural network classifier chip suitable for use in MATIC are then discussed and the design of the Bourke prototype neural network chip is described. Bourke implements a small

neural network and a bucket brigade device. The chapter ends with a presentation of the simulation, training and test results from the Bourke chip as tested on the XOR problem.

Results from Bourke assisted in the design of a second chip, called Kakadu. This chip design is described in Chapter 5. Kakadu contains a much larger neural network than Bourke and is used later as a morphology classifier for the MATIC algorithm. The experimental results obtained from the Kakadu chip are also described in this chapter. These include measurements of the chip's transfer function, power dissipation measurements and models, training and the results of classification tests.

The results of applying the MATIC algorithm to a large database of intracardiac electrograms are presented in Chapter 6. A comparitive study between the Kakadu chip and a normal software three layer perceptron is described and the MATIC classifier is also compared with a standard tachyarrhythmia classifier typical of that used in an ICD device.

In the final chapter, a summary of the work in this thesis and directions for future work are presented.

Chapter 2

Tachyarrhythmia Classification

2.1 Introduction

This chapter presents an introduction to the concepts and terminology from cardiology relevant to this thesis. It describes the principles of operation of the heart, the mechanisms which cause arrhythmias, classification of arrhythmias, previous techniques for automatic classification of arrrhythmias and features of a modern implantable cardioverter–defibrillator (ICD).

The chapter begins with a brief introduction to the physiology of the heart. This is followed by an outline of the cause, diagnosis and treatment of tachyarrhythmias that are common in ICD recipients. Earlier work conducted on the computer classification of arrhythmias from intracardiac electrograms is then described, including techniques based on the probability density function, heart rate, decision trees, frequency domain, sequential testing of the threshold crossing interval, template matching, neural networks and techniques which combine timing and morphology. The chapter concludes with a description of the Telectronics 4210 as an example of a modern ICD.

2.2 Basic Cardiology

This section is a brief introduction to aspects of cardiology applicable to work on intracardiac electrogram (ICEG) classification. The anatomy and physiology of the heart's conduction system is explained as well as the mechanisms, diagnosis and treatment of tachyarrhythmias common in ICD devices.

Further information on cardiology can be found in Julian's book [Jul88] which provides a good general primer on all aspects of this field. For more detailed works on arrhythmias, Mariott and Conover [MC83] and Josephson and Wellens [JW84] are good references. Durbin [Dub89] is an excellent text on interpreting the 12 lead electrocardiogram. A stepwise approach to diagnosing tachyarrhythmias using the electrocardiogram and electrophysiology is presented by Wyndham [Wyn91]. A glossary is provided at the end of the thesis to assist the reader with the terminology and abbreviations.

2.2.1 Conduction System

The myocardium is cardiac muscle in the middle layer of the heart. The resting cell is polarised to a potential of approximately -90 mV, this value being called the transmembrane resting potential. Polarisation is maintained by a sodium pump within the cell which serves to exclude sodium ions from the cell.

When the cell membrane is electrically stimulated, the membrane potential increases to the threshold potential and the cell permeability is increased causing a rapid influx of sodium ions and a slower influx of calcium ions (see Figure 2.1). This causes depolarisation to occur and the transmembrane potential to become positive. Following depolarisation, repolarisation occurs until the resting potential is again reached and the ionic balance restored. In the first stages of repolarisation, a refractory period exists during which no further depolarisation can occur.

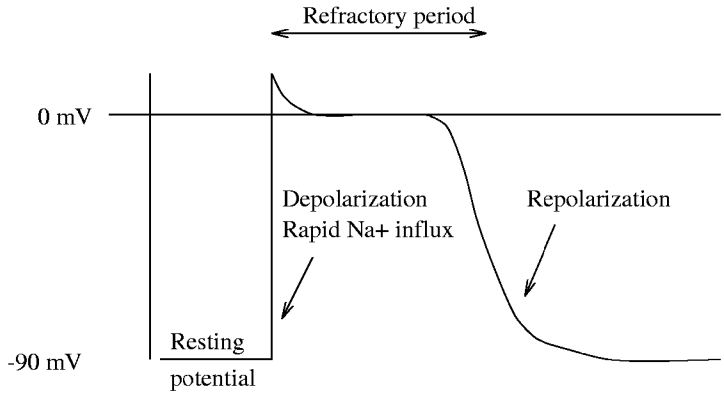

Figure 2.1: Cell potential during depolarisation and repolarisation.

The cells just described are called non–automatic because they require electrical stimulation to depolarise. As well as these, automatic cells which can self excite are also present. These cells exist throughout the conduction system and have different rates of spontaneous depolarisation.

2.2.1.1 Conduction in Normal Sinus Rhythm

The heart is controlled by a specialised network of muscles along which electrical activity (action potentials) is transmitted (see Figure 2.2). A normal heartbeat is initiated by an impulse formed by the automatic self–excitation of the sinoatrial (SA) node. The SA node has the fastest rate of spontaneous depolarisation and so acts as the heart's natural pacemaker. If for any reason the SA node does not depolarise, other automatic cells will spontaneously depolarise to ensure that the heart continues to beat.

The action potential is conducted via muscle fibres to the atrioventricular

Figure 2.2: Heart conduction system.

(AV) node and as it spreads along this route, atrial contraction occurs causing blood to be forced into the ventricles.

The AV node serves to delay the propagation of the action potential and allows time for the atria to complete its contraction. After this delay, the action potential spreads along the Purkinjie fibres through the bundle of HIS and is divided into left and right Purkinjie bundles which extend into the left and right ventricles. The Purkinjie fibres are fast conducting so that the action potential rapidly spreads through the ventricles causing depolarisation of the ventricular muscle. This, in turn, pumps blood throughout the body.

2.2.2 ICEG Data and Electrophysiological Studies

The data used in all experiments described in this thesis were obtained by the process of cardiac catheterisation during electrophysiological studies (EPS). During EPS, temporary catheters are inserted into veins and advanced under fluoroscopic guidance into the internal walls of the heart (called the endocardium).

The heart potential can be monitored and stimulated via these catheters, and tachyarrhythmias can be induced and terminated by appropriate stimulation. Recordings of the heart potential can be made from these catheters, one of which is usually placed in the right ventricular apex (RVA) and one in the high right atrium (HRA). These two probes enable the heart's conduction sequence to be determined. A catheter is also often placed on the HIS bundle so that the direction of propagation can be better determined. Although a HIS bundle recording is very useful for diagnosing arrhythmias, permanent catheters cannot be placed in this position at present. Therefore, classification algorithms for implantable devices should not depend upon information from the HIS bundle.

The recorded trace of the catheter's potential is called an intracardiac electrogram (ICEG). On the RVA lead, the QRS complex is the point when depolarisation of the ventricular muscle occurs. This is usually a large amplitude, very sharp deflection in the electrogram (see Figure 2.3) and is formed from Q, R and S waves. The R wave is any positive deflection of the QRS, the negative deflection preceding the R wave is called the Q wave and the negative deflection following the R wave is called the S wave. In the HRA channel, the P wave corresponds to the spread of electrical activity though the atria and is identified by a sharp deflection in this channel.

The RR interval is the instantaneous ventricular heart rate and is the time between successive R waves. Similarly, the PP interval is the instantaneous atrial rate and this is the time between successive P waves. The PR interval is the time from a P wave to the next R wave and this measures the time between atrial and ventricular depolarisations.

The recordings made via the internal catheters are different from those obtained from the 12 lead electrocardiogram (ECG) which are potentials recorded on the skin surface. The ICEGs record potentials local to the vicinity of the probes whereas the ECG records a global view of cardiac activity. Thus, in an ECG, P waves and R waves can be seen on the same trace, whereas in the RVA

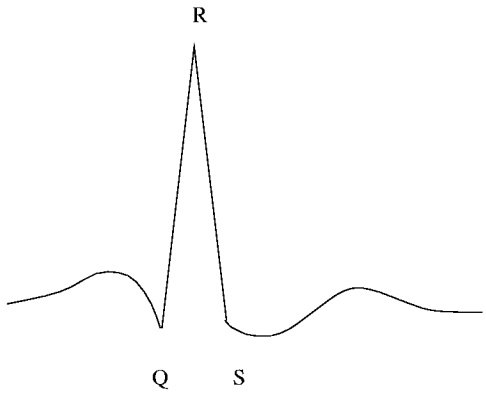

Figure 2.3: QRS complex.

channel, only ventricular depolarisation can be observed and similarly, in the HRA channel, only atrial activity is observed.

2.3 Intracardiac Electrogram Classification

A basic guide to the classification of ICEGs, grouped into the two categories of arrhythmias and conduction system abnormalities is presented in this section. Note that this is a very informal description, providing only enough material to achieve an understanding of ideas in this thesis, many simplifications having been made.

2.3.1 Arrhythmias

Abnormalities in the electrical activity of the heart are called arrhythmias. Arrhythmias occur when there is abnormality in the rate, regularity, conduction or origin of the cardiac impulse. Impulses not originating in the SA node are called ectopic beats and most arrhythmias which are studied are due to this effect. Mechanisms which can cause ectopic beats include

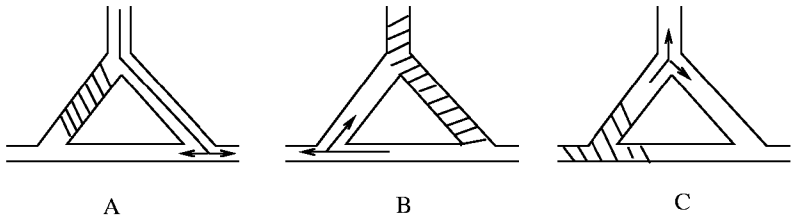

Figure 2.4: Mechanism for re–entry. In A, conduction cannot proceed into the shaded zone since it is refractory. By B, the zone is no longer refractory and conduction proceeds through the previously refractory zone. In C, the wave of depolarisation re enters the area which was previously depolarised by A.

1. Reduction of the spontaneous depolarisation rate of the sinus node to less than that of another group of automatic cells which then take over the pacemaker function of the heart.

2. Increased rate of automatic depolarisation of other groups of automatic cells.

3. The "re–entry" mechanism whereby two or more pathways of conduction with different refractory periods enable ectopic rhythms to be formed (see Figure 2.4).

The following sections offer an introduction to the diagnosis of certain tachycardias including sinus tachycardia (ST), normal sinus rhythm (NSR), atrial tachycardia (AT), atrial flutter (AFlut), atrial fibrillation (AF), supraventricular tachycardia (SVT), ventricular tachycardia (VT), fast ventricular tachycardia (VTF) and ventricular fibrillation (VF). These tachycardias are referred to as "subclasses". In this thesis, all of the arrhythmias are grouped into four major "superclasses", NSR, SVT, VT and VF. This is because these four superclasses

Subclass	Superclass
NSR	NSR
ST	NSR
SVT	SVT
AT	SVT
AFlut	SVT
AF	SVT
VT	VT
VT 1:1	VT
VTF	VF
VF	VF

Table 2.1: Subclass to superclass mapping table.

Tachycardia	Atrial Rate	P wave Morphology	Ventricular Rate	QRS Morphology	PR interval
ST	100 – 140	Normal	100 – 140	Normal	Normal
AT	140 – 220	can be Abnormal	70 – 220	Normal	Lengthened
AFlut	280 – 300	Abnormal	140 – 220	Normal	Dissociated
AF	300+	Chaotic	100 – 150	Normal	Dissociated
VT	60 – 100	Normal	140 – 170	May widen	Dissociated
VTF	60 – 100	Normal	170 – 270	Wide	Dissociated
VF	60 – 100	Normal	270+	Chaotic	Dissociated

Table 2.2: Tachycardia diagnosis guidelines. A rough guide to identifying tachycardias.

correspond to the four different treatments that an ICD can deliver. The mapping from subclasses to superclasses is shown in Table 2.1.

While reading the following sections, it may be useful to refer to Table 2.2 which is a summary of the tachyarrhythmia characteristics.

2.3.1.1 Normal Sinus Rhythm

When the heart is operating normally (as described in Section 2.2.1.1), the rhythm is said to be a normal sinus rhythm and a typical ICEG of NSR is shown in Figure 2.5. During NSR, the normal sequence of events in the ICEG will be that the P wave will occur shortly after depolarisation of the SA node, a delay of 100 to 200 milliseconds (PR interval) will occur as the action potential propagates through the AV node, and then the QRS complex will appear when

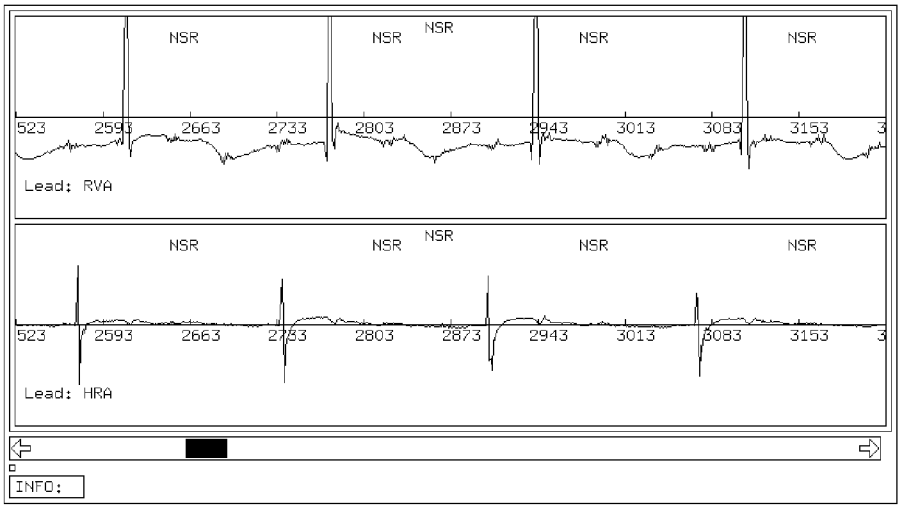

Figure 2.5: Normal sinus rhythm. Numbers along the ICEG represent the sample number of the signal (sampling period is 4 ms).

depolarisation of the ventricles occur. Every P wave has a corresponding R wave for NSR and the heart rate is usually between 60 and 100 beats per minute (bpm).

A tachyarrhythmia is any heart rhythm with ventricular rate greater than 100 bpm. Sinus tachycardia is simply a normal sinus rhythm with a fast heart rate ($>$ 100 bpm) and is commonly induced from fright, excitement, sudden effort or postural change.

2.3.1.2 Supraventricular Tachycardias

Supraventricular tachycardias (Figure 2.6) are caused by ectopic beats originating in or above the AV node. They are typically in the range of 140–220 bpm and usually have a narrow QRS complex. The P wave often becomes abnormal in shape. The atrial rate may be faster than the ventricular rate and the PR interval is usually changed from that seen in NSR. Atrial tachycardia has an atrial rate

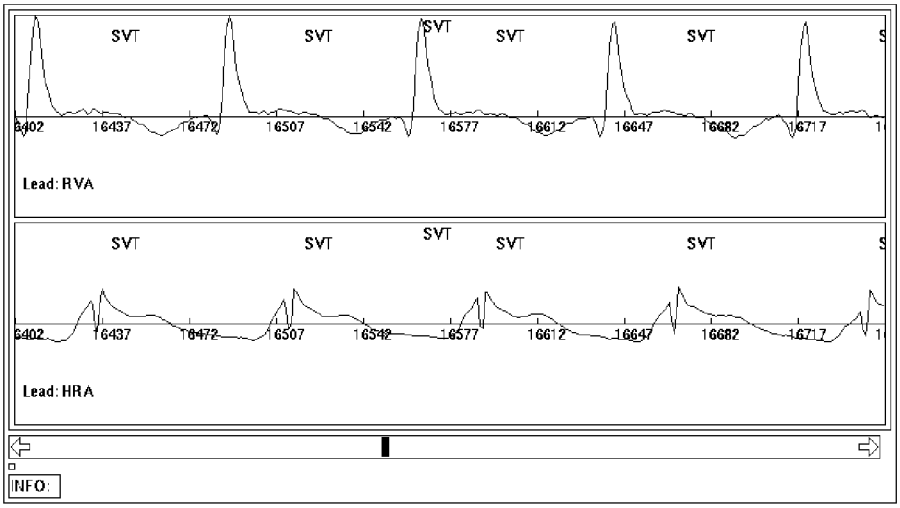

Figure 2.6: Supraventricular tachycardia.

of approximately 140–220 bpm, atrial flutter has an atrial rate of 280–300 bpm and atrial fibrillation is usually faster than 300 bpm. For most SVTs, except AF, ventricular rate is usually 140 220 bpm. During AF, the ventricular rhythm is irregular and the rate is approximately 100–150 bpm.

Current ICD devices have support neither for the identification nor the treatment of SVT. Current therapy is to treat SVT with drugs, surgery or ablation, and these techniques are usually successful [dBC89]. However, experiments have shown that SVT can be terminated by either atrial pacing or dual chamber pacing [SS76, MMK+81], and this may become a therapy in future ICDs.

2.3.1.3 Ventricular Tachycardia

Ventricular tachycardias are caused by ectopic beats originating in the ventricles. The ventricular rate of VT (Figure 2.7) is usually in the range of 120–220 bpm. The P waves are usually normal in rate and shape since the atria remains under

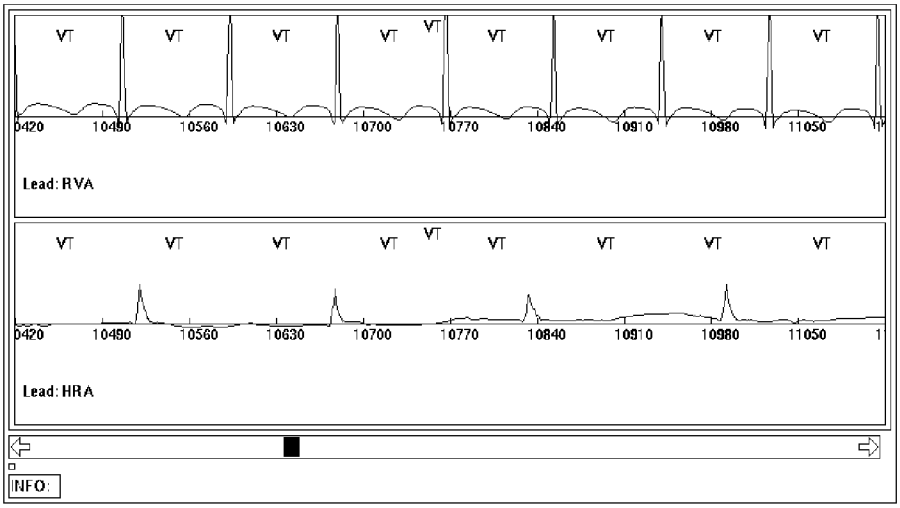

Figure 2.7: Ventricular tachycardia.

control of the SA node. VT is usually dissociated, the atrial and ventricular beats being independent. In addition, VT often causes a widening of the QRS complex.

In VT with 1:1 retrograde conduction (Figure 2.8), no dissociation occurs and the atrial rate is the same as the ventricular rate due to a reverse conduction, the wave of depolarisation passing from the ventricles to the atria. VT 1:1 often causes a widening of the QRS complex or some other change in the QRS morphology. In addition, the PR interval can change from that during NSR. The diagnosis of VT 1:1 often requires HIS bundle recordings to determine the direction of the conduction.

The ICD treatment for VT is usually either antitachycardia pacing or a low energy DC cardioversion shock. In the case of VTF, which is simply VT with a 170–270 bpm ventricular rate, treatment is the same as that for VF.

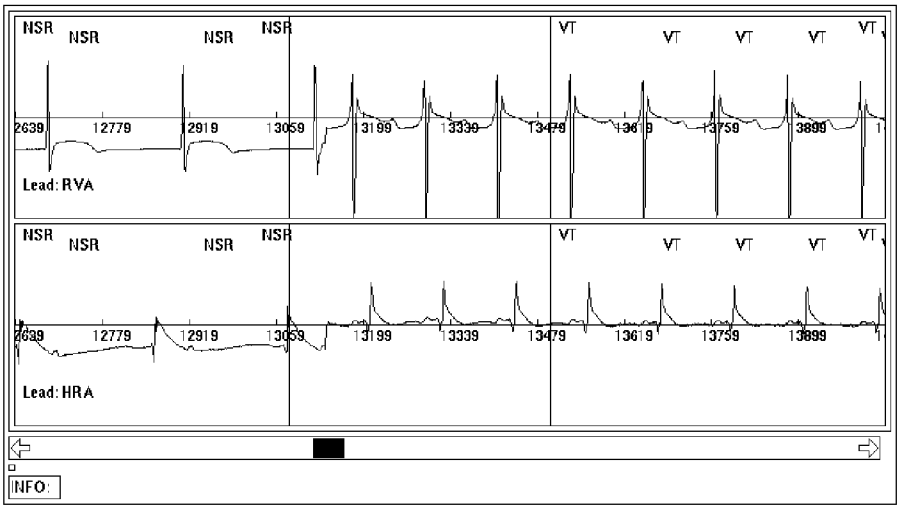

Figure 2.8: Ventricular tachycardia with 1:1 retrograde conduction.

2.3.1.4 Ventricular Fibrillation

Ventricular fibrillation (Figure 2.9) is identified by chaotic ventricular activity on the RVA lead. Ventricular rate is usually greater than 300 bpm. VF will cause permanent brain damage if not treated within four minutes since there is no cardiac output during the episode. The treatment of VF is to apply a high energy electric shock which serves to defibrillate the heart.

2.3.2 Conduction Abnormalities

This section describes cardiac conditions which are caused by delays or blockages in the conduction system of the heart. These are not caused by ectopic beats but rather by conduction delays and blockages. They are important because an arrhythmia classifier can often be fooled by such conduction abnormalities into producing an incorrect classification.

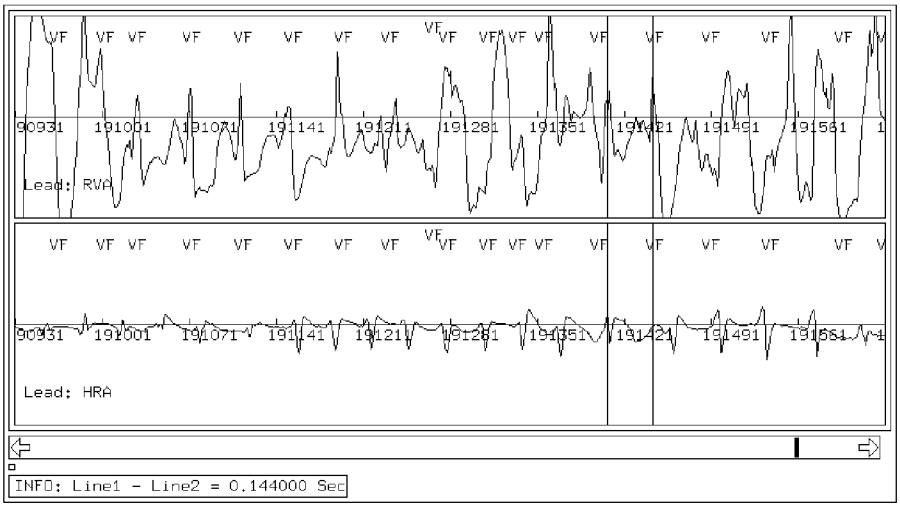

Figure 2.9: Ventricular fibrillation.

2.3.2.1 Bundle Branch Block.

Bundle branch block (BBB) is caused by a delay in the conduction of either the right or left parts of the ventricular conduction system. Because the signal is delayed in one half of the ventricle, the shape of the QRS is widened and sometimes notched.

A BBB which occurs with a sudden onset (paroxysmal BBB) can be indistinguishable from VT in the RVA channel although VT QRS complexes are usually wider than for BBB. However, most BBBs are not paroxysmal, being permanent in nature. Bundle branch block is normally left untreated even though a small decrease in cardiac output occurs.

2.3.2.2 Atrioventricular Block

Atrioventricular (AV) block is any delay or block in the AV node. AV block is grouped into three degrees.

First degree AV block occurs when the AV interval increases beyond 200 ms on an external ECG. In second degree AV block, intermittent conduction through the AV node occurs, and this can result in patterns with two or more atrial depolarisations for each ventricular depolarisation. Third degree AV block is a complete block and no atrial depolarisations pass through the AV node to cause ventricular depolarisation. AV block often occurs after defibrillation therapy and can be treated by pacing.

2.4 Earlier Work on Computer Classification of Arrhythmias

Many ICEG classification algorithms have been proposed and this section provides an overview of techniques described in the literature.

2.4.1 Probability Density Function

The probability density function (PDF) method developed by Langer et al. [LHMM76] was the very first classification algorithm used in an implantable defibrillator and is still available as an optional detection criterion in some ICD devices today. This technique can be implemented by passing the RVA channel through a window comparator and then low pass filtering the output.

The PDF, as defined by Langer, is the function which describes the fraction of time on average that the signal spends between two limits. In the case of NSR the PDF has a large peak about zero since the signal spends a significant amount of time around zero. In the case of VT and VF, the sinusoidal nature of the signal means that very little time is spent near zero potential and the PDF does not have a peak about zero.

Toivonen *et al.* [TVJ92] have applied PDF to differentiate between SVT and VT in patients with implanted ICDs. They concluded that PDF was not

a reliable method for making this distinction since 7 out of 12 patients met the PDF criteria even though the patients were in sinus tachycardia. From this, Toivonen concluded that PDF would not be able to reliably prevent inappropriate discharges of an ICD during fast supraventricular tachycardias.

2.4.2 Single Channel Heart Rate

All current ICDs rely mainly on measurements of the heart rate taken from the RVA channel as their primary means of arrhythmia detection [KW90]. A combination of the following criteria is normally used, specificity being improved by using multiple criteria

- rate cutoff
- sudden onset
- rate stability
- sustained high rate

The rate cutoff criterion is met if the heart rate goes above a programmable limit. This method fails to differentiate between arrhythmias with overlapping heart rates such as ST, SVT and VT. The suddenness of onset can often be used to help differentiate between ST and VT, and this has been shown to yield the highest degree of specificity when used together with rate cutoff. Rate stability often indicates whether the arrhythmia is ventricular in origin, VT usually having high rate stability whereas SVTs often have a large ventricular rate variation. A sustained high rate is used to recognise arrhythmias that do not meet the rate cutoff and sudden onset criterion. In a test involving 50 patients, Warren *et al.* [WM86] showed that the incidence of false tachycardia detection during ST can be reduced from 93% (rate cutoff alone) to 3% by adding sudden onset detection to the rate cutoff criteria.

2.4.3 Dual Channel Decision Tree

Most of the incorrect classifications made by single channel devices are caused by the overlap of heart rate ranges of sinus tachycardia (ST), supraventricular tachycardia (SVT) and ventricular tachycardia (VT). The addition of an atrial lead provides a means whereby the sequence of atrial and ventricular depolarisations can be determined and very reliable classification of these arrhythmias can be made [ABJ+84, MF86, SJSL88].

The dual channel decision tree was first implemented by Arzbaecher *et al.* [ABJ+84]. This simple decision tree monitors the intervals among and between the two channels (see Figure 2.10). Arzbaecher estimated that the duty cycle of a microprocessor implementing his algorithm would be less than 1%. He also suggested that the addition of morphological information may be used to differentiate between ST and retrograde conduction. Arzbaecher tested this algorithm on 22 patients and it was successful in all cases but one. The misclassified arrhythmia was a VT which was "almost" 1:1, 7 out of 8 beats conducting in a retrograde fashion.

2.4.4 Frequency Domain

Fourier transforms were first applied to the ECG by Nygards and Hulting [NH77] in 1977. Pannizzo and Furman [PF88] applied Fourier transforms to the ICEG in order to detect the difference in frequency spectra between NSR and VT. Their studies indicate that there is an 8 Hz difference in the peak amplitude between NSR and VT. However, Fourier transforms are computationally expensive and time domain methods can also be used to make this distinction.

2.4.5 Threshold Crossing Interval

Zhu and Thakor [ZT88] developed a sequential testing method which converts a one second buffer of intracardiac signals into a binary sequence. This is done

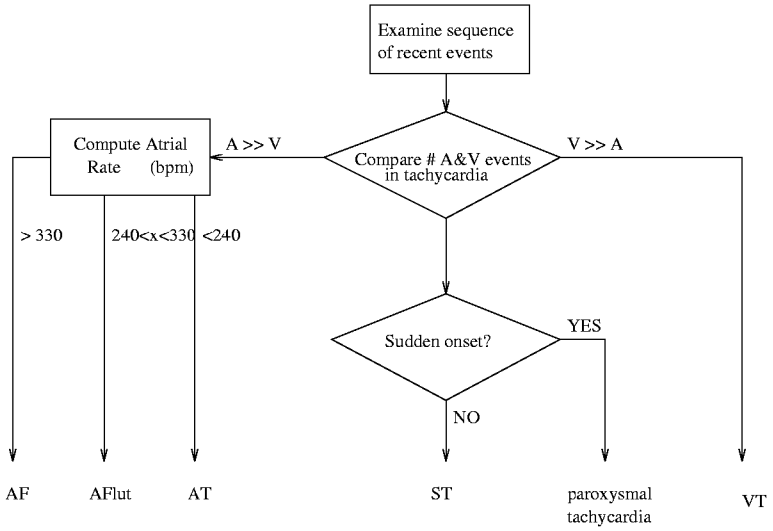

Figure 2.10: Arzbaecher *et al.* method of arrhythmia classification using two leads and a decision tree [ABJ+84].

using the equivalent of a comparator with a threshold set to 20% of the peak value of the stored signal. The number of threshold crossing intervals (TCI's) in the buffer is then computed. The sequential testing algorithm can distinguish between two arrhythmias based on the computed TCI and a precomputed threshold, the output being either VT, VF or NOTSURE. If the output is NOTSURE, the next TCI is used to add confidence to the decision. This process continues until either a decision is made or the number of one second trials exceeds a fixed value, whereupon the algorithm starts again and no decision is made.

To perform the sequential hypothesis test, Gaussian distributions are fitted to the data and the mean and standard deviation are computed. Error probabilities of rejecting the hypothesis of a VF when it is true (α) and of accepting the hypothesis of a VT when it is false (β) are selected and these can be chosen to be arbitrarily small. From α and β, the following thresholds can then be computed

$$E_{VT} = \ln \frac{1-\beta}{\alpha} + m \ln(\frac{\sigma_{VT}}{\sigma_{VF}}) \tag{2.1}$$

$$E_{VF} = \ln \frac{\beta}{1-\alpha} + m \ln(\frac{\sigma_{VT}}{\sigma_{VF}}) \tag{2.2}$$

$$\tag{2.3}$$

where m is the number of observations, σ is the standard deviation and μ is the mean of the respective distribution.

If the log–likelihood function

$$f(m) = \frac{1}{2\sigma_{VF}^2} \sum_{i=1}^{m} (T_i - \mu_{VF})^2 - \frac{1}{2\sigma_{VT}^2} \sum_{i=1}^{m} (T_i - \mu_{VT})^2 \tag{2.4}$$

where T_i is the ith TCI of the test is computed, it is possible to test whether the signal is VT or VF by saying it is VF if $f(m) < E_{VF}$, and correspondingly, it is VT if $f(m) > E_{VT}$. If $E_{VF} \leq f(m) \leq E_{VT}$, the decision can be delayed until a higher confidence level is reached by increasing m. If m gets larger than a fixed number, the observations are discarded and the algorithm starts again with $m = 1$. The sequential testing method thus allows the probability of an error to be traded off against the latency of the algorithm.

2.4.6 Template Matching Techniques

Many template matching techniques have been proposed for differentiating between ST and VT. These methods involve producing a template of the NSR by averaging several complexes and then using correlation techniques to measure the likeness of a rhythm with the template. Such techniques include area of difference [TJD90b], correlation waveform analysis (CWA) [LDJ88], and the bin area method [TJD90a].

Jenkins *et al.* have performed many comparative studies in this area [LJDM88, LDJ88, TJD90a, TDJW90, TJD90b, DTJ91] and have achieved excellent results in differentiating between NSR, VT and retrograde conduction. CWA uses the correlation function defined by [DTJ91]

$$\rho = \sum_{i=1}^{N} \frac{(t_i - \overline{t})(s_i - \overline{s})}{\sqrt{\sum_{k=1}^{N} (t_k - \overline{t})^2} \sqrt{\sum_{k=1}^{N} (s_k - \overline{s})^2}} \tag{2.5}$$

where N = number of points in the template, t_i are the template points, s_i are the signal points to be processed, $\overline{t}$ is the template average and $\overline{s}$ is the signal average. The correlation coefficient ρ is independent of amplitude and baseline fluctuations and produces an output $-1 \leq \rho \leq 1$.

The bin area method is similar except that it was developed to be more computationally efficient. The "bins" are sequential sample points which are grouped together to make the computation more efficient. For M equally sized bins, the equivalent to the correlation coefficient η is given by

$$\eta = 1 - \sum_{i=1}^{M} \left| \frac{(T_i - \overline{T})}{\sum_{k=1}^{M} |T_k - \overline{T}|} - \frac{(S_i - \overline{S})}{\sum_{k=1}^{M} |S_k - \overline{S}|} \right| \tag{2.6}$$

where all the variables are the same as those used for CWA and $S_i = \sum_{k=1}^{M} s_{M\times(i-1)+k}$, $T_i = \sum_{k=1}^{M} t_{M\times(i-1)+k}$, $\overline{S} = \frac{1}{M}\sum_{k=1}^{M} s_k$ and $\overline{T} = \frac{1}{M}\sum_{k=1}^{M} t_k$. For 3 point bins, one–sixth the number of multiplications are required by the bin method as compared to CWA.

2.4.7 Neural Networks

Artificial neural networks are computing architectures inspired by biological neural systems. They are parallel architectures formed from the interconnection of many neurons, and have been successfully applied to many pattern matching problems. In contrast to correlators, they can store more than one pattern as well as solve non–linear problems [Lip87]. Please refer to Appendix A for an introduction to the architecture and training of artificial neural networks.

Individual QRS complexes have been used as inputs to artificial neural networks by Chi and Jabri [CJ90]. This system used 40 samples centered about the QRS complex of the RVA signal to distinguish between NSR, SVT and VT. A correct classification rate of 98.95% was achieved by this single channel technique when applied to a database of 12 patients.

Chi and Jabri [CJ92] have also used a multi–module neural network that uses the features: a history of RR interval, PP interval, PR interval, PR interval

divided by the RR interval, RR interval minus PP interval, the average RR interval, average PR interval, average PP interval and the standard deviation of the RR intervals. The multi–module network approaches the problem of ICEG classification by dividing the classification problem into the simpler subproblems of differentiating between

- NSR and SVT
- VT and VF
- (NSR or SVT) and (VT or VF).

Using this system on a database of 51 patients (16 for training and 35 for testing), 96.2% correct classification was achieved.

A problem that occurs with ICEGs is incorrect classification can result when the QRS complex is not centered in the same fashion as the exemplar patterns. This problem was addressed by Lee using a shift invariant network [Lee90] for single channel signals. The second order network used by Lee was formed by replacing the original inputs x_i with new inputs y_i which can be described by the equations

$$y_0 = \sum_{i=1}^{N} |x_i| \tag{2.7}$$

$$y_k = \sum_{i=1}^{N-k} |x_i - x_{i+k}| \tag{2.8}$$

where N is the number of inputs and $k = 1 \cdots N-1$. The input size was 135 samples of an ICEG digitised at 100 Hz. Of the 52 patients tested, 48 could be classified with better than 90% accuracy when the algorithm was used to distinguish between NSR, VT and VF. One disadvantage of the second order network is that it is much more computationally expensive than a first order network.

Farrugia *et al.* [FYN91] used a combination of features to classify ICEGs:

- a running average of the RR interval

- a filtered value of the rectified signal
- the integrated outputs of 3 bandpass filters
- 10 samples taken from near the detected QRS complex.

This system was trained on 10 patients with NSR, VT and VF and tested on 20 patients, achieving a 95.4% classification rate.

2.4.8 Timing and Morphology

In a sense, timing and morphology criteria have been used as early as 1982 in the form of ventricular rate plus PDF [Tro90]. However, timing was only based on a single channel and the PDF detector could not recognise specific morphologies which template matching techniques can now do.

Jenkins *et al.* [JWA79] have produced an arrhythmia classifier which makes use of both timing and morphology from two leads, an esophageal electrode for monitoring atrial depolarisation and a surface electrode to monitor QRS complexes. This system was able to diagnose a wide variety of arrhythmias including bigeminy, trigeminy, ventricular tachycardia, supraventricular tachycardia, atrial flutter, atrial fibrillation, ventricular tachycardia with retrograde conduction, heart block and bradycardia.

Single beat classification was made according to the PR, RR and PP intervals and the output from the single beat classifier is further analysed by a contextual analysis module which classifies sequences. This classifier was tested on 29 12–second passages and achieved 95.5% classification for single beats and 94% for sequences.

Lin, Jenkins *et al.* [LJDM88] have also produced a classifier which uses the esophageal and surface leads and the same single beat/contextual classifier structure. This system was able to diagnose a wide variety of arrhythmias including ST with aberrant conduction, ST with AV delay, premature atrial depolarisations, ventricular ectopic beats, bigeminy, trigeminy, 1^o AV block, 2^o AV block,

SVT, AF, VT, VFlut, VT with 1:1 retrograde conduction, AF with 1:1 or n:1 conduction and VF.

Timing was examined using interval analysis and morphology examined using CWA. Features are extracted from these two detectors and a single beat classification is made. A contextual diagnosis examines the 8 most current beats to obtain sequence information which is used in the final classification. With this contextual classifier, n:1 conduction patterns can be deduced.

Lin's algorithm was tested on 29 patients and achieved 99.2% correct classification for the single beat diagnosis and 99.5% accuracy for the contextual diagnosis.

2.5 An Example of an Implantable Cardioverter Defibrillator

In this section a description of the features available in the Telectronics Guardian ATP 4210 are described [Ltd89]. This device, currently undergoing clinical trials, is representative of the state of the art of ICDs in 1992 and other companies such as Cardiac Pacemakers Incorporated (CPI), Siemens, Medtronic, Intermedics and Ventritex have similar devices [Tro90].

The 4210 is a truncated oval in shape, has dimensions of $115 \times 81 \times 20$ mm, weighs 269 gm and the entire unit is housed in a titanium case. The battery is a WG 8512 Lithium Silver Vanadium Pentoxide cell with a 4.1 ampere hour capacity. The approximate lifetime of the battery is 6 years if therapy is never applied. Battery life, will of course be much shorter if many discharges are made. There are two lead systems associated with the 4210, a sensing/pacing lead system which is used to sense ventricular activity and to deliver pacing pulses, and a defibrillation lead system for delivering low and high energy cardioversion and defibrillation shocks.

The tachyarrhythmia detection algorithm in the 4210 has four prioritised

algorithms which are fibrillation detection, tachycardia detection, onset detection and noise detection. Each algorithm works by comparing the current rate with a preprogrammed rate and passing the result through an X out of Y detector which ensures that the events are sustained before therapy is applied. In order to avoid confusing noise caused by external sources with VF, tachyarrhythmia detection is turned off if 7 out of 10 intervals are shorter than 100 ms.

Once a tachyarrhythmia is detected, the arrhythmia is classified as either slow VT, fast VT or VF using a rate plus X out of Y algorithm. In the case of slow VT, antitachycardia pacing (ATP) is delivered. This is a (programmable) train of pulses which are synchronised to the QRS events and changed for successive trains if the tachycardia is not broken. In the case of fast VT, if ATP therapy is not successful, then shock therapy is delivered. Both low energy cardioversion and high energy defibrillation shock therapy is available and these are also synchronised to the QRS events. With VF, shock therapy is delivered immediately. If a first shock does not cause the tachyarrhythmia to revert to NSR, shock therapy continues with increasing energy.

Apart from the basic ICD functions described above, the Guardian also includes bradycardia pacing, data logging of ICEG during tachyarrhythmias, tachyarrhythmia induction through stimulation via the sensing/pacing lead, programmability via telemetry, real time ICEG via telemetry and many other features.

2.6 Summary

In this chapter, an introduction to the relevant aspects of cardiology, tachyarrhythmia classification and ICDs was presented. A review of earlier work

on ICEG classification was also presented and these included timing based techniques which rely only on the intervals between atrial and ventricular depolarisations [1], frequency domain techniques, morphological techniques [2], neural networks, and timing plus morphology.

[1] Such as heart rate and decision trees.

[2] Such as probability density function, threshold crossing interval, correlation waveform analysis and the bin area method.

Chapter 3

A System for the Classification of Arrhythmias

3.1 Introduction

Earlier work in automatic classification of ICEG signals concentrated on producing accurate classifiers and power consumption was of little concern. In this chapter, the design of an algorithm which is both accurate and amenable to low power implementation is described. The algorithm, called MATIC (morphology and timing intracardiac electrogram classifier), processes timing information using a decision tree, and morphology using a neural network. An analysis of the power consumption of the neural network implemented on a microcontroller is given and this is shown to consume too much power to be practical in an implantable device. An alternative system architecture which utilises a low power VLSI neural network chip is presented.

3.2 Design Considerations

It is evident that the reliable classification of ICEG signals is difficult because of a large patient variability in heart rate and morphology. When the classifier

must be used in an ICD, the problem is made even more difficult because the goal of low power consumption precludes algorithms which have high computational demands.

Earlier work on ICEG classification has shown that dual channel methods have a clear advantage over single channel techniques since they use the extra information available to glean knowledge about the atrioventricular conduction sequence. Of the morphology classifiers, PDF and TCI are much more crude than CWA and neural networks. This is because only the latter techniques store a detailed template of the waveform to be matched. Systems which use both timing and morphology show a lot of promise since morphology can add specificity to a timing algorithm. In this section, design decisions which lead to the choice of this morphology plus timing technique are detailed.

3.2.1 Design Goals

The main design goals of the MATIC algorithm were as follows:

- implementable with low power consumption
- perform reliable classification of arrhythmias based on two leads
- be adaptable to identify other arrhythmias not previously addressed
- produce the recommended ICD therapy as output

These goals are discussed in relation to their effect on the design of the MATIC algorithm in the following sections.

3.2.1.1 Low Power Consumption

In traditional systems design, the largest performance gains are usually made at the algorithmic level. Power consumption limitations impose a major limitation on the amount of computing that can be performed, thus affecting the choice of algorithm.

The design of the algorithm must be such that it maps well to VLSI technology, *i.e.* a significant power saving can be made over a software implementation and that such an implementation is feasible with available resources. The flexibility of VLSI allows for a custom architecture, optimised for low power consumption.

3.2.1.2 Two Lead Classification

Current ICD devices base their classification on a single ventricular lead, and this is the main reason for their failure to classify many arrhythmias, especially those which are supraventricular in nature. The addition of an atrial lead allows timing sequences to be evaluated between the ventricles and atria, making more accurate classification possible. Note that, in general, it is not possible to reliably distinguish between ST, SVT and VT based on a single lead.

For these reasons the next generation of ICD devices will base their classification on two leads, and so a dual input was used for the MATIC classification algorithm.

3.2.1.3 Patient Dependent Versus Patient Independent Classification

A patient independent classifier can be used in any patient without adjustment. This is clearly advantageous since the time consuming, expensive and potentially dangerous process of determining the parameters can be avoided.

Patient dependent classification, whereby classifier parameters are tuned specifically for each patient, can always achieve at least equal and usually better performance than an equivalent patient independent method. This is because a knowledge of a patient's history and likely tachycardias can be specifically entered into the classification parameters.

Current ICDs are patient dependent, and are able to perform reliable classification using crude rate based algorithms by adjusting thresholds above which VT and VF are presumed. These thresholds are normally determined from the

rate of arrhythmias observed during electrophysiological studies and also from information logged by the ICD.

A compromise between these conflicting goals is the solution taken in MATIC. Patient independent classification is performed for the majority of patients, and special cases can be addressed through patient dependent means.

3.2.1.4 Recommended ICD Therapy

Since the classifier outputs a recommended ICD therapy, it need not be able to distinguish between various arrhythmias which have the same treatment (such as fast ventricular tachycardia and ventricular fibrillation). For this reason, the classifier makes four different classifications, NSR, SVT, VT and VF. Tachycardias such as fast VT for which the treatment is defibrillation, are classified as VF even though it is clinically a VT.

This classification into four crude classes can be used to an advantage. Rather than classify the arrhythmia into specific classes, many different tachycardias are merged thus saving on the number of cases and requiring less exacting classifications of similar subgroups.

3.2.2 Mimic a Cardiologist

MATIC was inspired by observing the reasoning process of cardiologists to see how they make diagnostic decisions. Firstly, it was apparent that a classification of arrhythmias usually uses more information than that available to an ICD. Patient history, 12 lead ECG, information on drug therapy being used and EPS are all taken into account in a diagnosis by a cardiologist.

However, if the information is restricted to the RVA and HRA leads of an ICEG, experts will first look at the timing within and between atrial and ventricular channels and can usually form a decision based on this information alone. In some cases, the morphology of the arrhythmia is compared to the patient's normal sinus rhythm to see if any change has occurred. The actual morphology

of the ICEG is also important, in particular, the shape of the QRS complex has diagnostic value, especially for ventricular tachyarrhythmias. In the following sections, more detailed discussion on the timing and morphology strategies is presented.

3.2.2.1 Timing

The timing decisions can be performed by a decision tree and, in fact, texts such as that by Wyndham [Wyn91], use this approach to describe how to achieve a diagnosis. It is natural for humans to express their thinking process in the form of a decision tree and this approach is used to implement the timing logic since it leads to an easily understood and efficient implementation.

3.2.2.2 Morphology

It is much more difficult to perform morphological classification than timing classification since subtle changes in morphology which are apparent to a cardiologist are hard to detect using conventional computing methods. This task is made even more difficult if low power consumption is required at the same time.

The same success rate that a cardiologist enjoys cannot be expected of a computer based system, nevertheless, template matching can be used to identify morphologies. An artificial neural network approach was chosen to perform this template matching because it has the advantages of being able to store multiple patterns, form arbitrary decision regions and maps well to VLSI.

The ability to store multiple morphologies is desirable since patients may have polymorphic arrhythmias and so it is especially useful to be able to train the neural network to recognise all of these morphologies as well as the morphology of the normal sinus rhythm. The only way this can be achieved for correlation waveform analysis is if several correlators are used. The ability to form arbitrary decision regions makes the neural network a universal approximator and thus in theory it can approximate any other technique to arbitrary accuracy.

Throughout this thesis, morphology is used only to detect ventricular tachycardia with 1:1 retrograde conduction. This is a first step in achieving classification ability for an ICD which approaches that of a human expert. Morphology can be used in many patient dependent ways, and would be able to distinguish between SVT, VT, right bundle branch block, left bundle branch block and many other rhythms which cannot be distinguished based on timing features alone.

ICDs have not used sophisticated morphology classifiers in the past because their cost in terms of power and development have outweighed their advantages. With the development of the low power neural network classifier chips such as the one designed in this thesis, an algorithm such as MATIC can use morphology whilst still maintaining low power consumption.

3.2.3 Morphology Plus Timing

The idea of using decision trees and neural networks to mimic a cardiologist meets all of the design goals discussed in Section 3.2.1. Low power consumption is achieved by using a decision tree plus custom VLSI to implement morphology. A two lead system is used. Timing parameters are extracted from the leads and are used as inputs to the decision tree. Samples of the input signal are used as morphological input to the neural network chip. The timing logic is patient independent, and the majority of patients will not require use of the morphology criteria. Morphology and an updated timing algorithm can be used in a patient dependent fashion for patients which could not be classified for whatever reason by the patient independent technique. A decision can then be made by merging all of the possible classifications into a therapy based classification using simple logic which combines the output of the morphology and timing units.

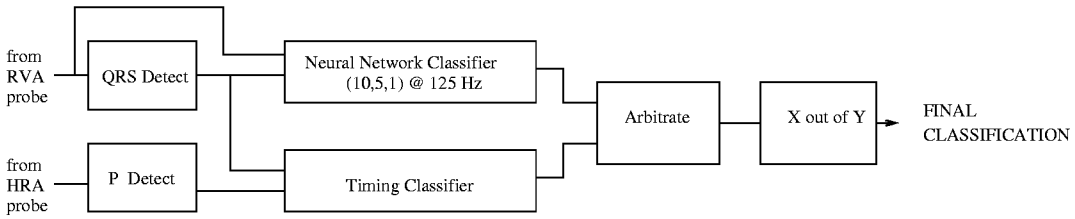

Figure 3.1: MATIC classification system. The timing logic and neural network operate in parallel.

3.3 The Algorithm

The development of the MATIC algorithm started with the design of the timing algorithm. It was found to produce very good classification results, except for ventricular tachycardias with 1:1 retrograde conduction. This was because the 1:1 conduction meant that dissociation did not occur, so one could not differentiate between SVT and VT based on timing. The morphology classifier was able to differentiate between NSR and VT 1:1 based on the QRS complex changing shape during the arrhythmia and so by combining information from both these classifiers, MATIC was able to produce accurate classification on the entire ICEG database.

A block diagram of the MATIC system is shown in Figure 3.1. The system consists of timing and morphology classifiers running independently and in parallel, each being invoked after a QRS complex occurs. The two classifiers are combined through arbitration logic and passed through an "X out of Y" detector to produce the final MATIC classification.

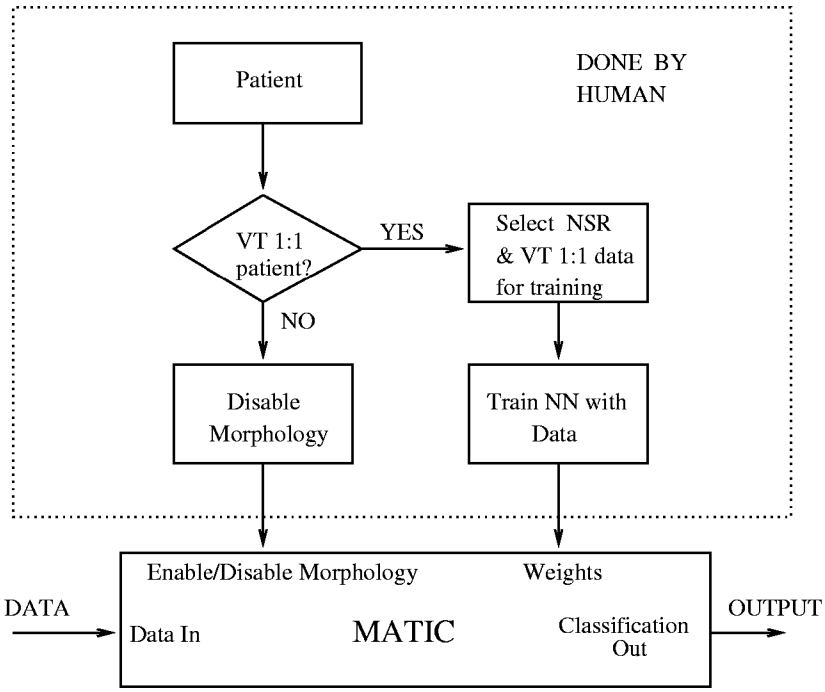

Figure 3.2: Flow chart/signal diagram of the MATIC configuration procedure. In the case of VT 1:1 patients training data must be selected and the neural network trained to produce a set of weights. Note that morphology is only enabled for VT 1:1 patients.

3.3.1 MATIC Configuration

Most patients do not require morphology classification, timing being sufficient to identify the arrhythmia. However, for some patients morphological classification is required. A human must configure the MATIC system before it is used in order to identify whether morphology is required. MATIC is used with morphology only for VT 1:1 patients. In the case of non VT 1:1 patients, the arbitration logic discards all results from the neural network. A flow chart/block diagram of this configuration process is shown in Figure 3.2.

Configuration involves deciding if the patient is a VT 1:1 patient or not and if so, 4 samples each of the patient's NSR and VT 1:1 morphology must be provided to serve as templates for the morphology classifier. As most current ICDs have telemetry features from which one can obtain the raw ICEG data, these samples can be selected via telemetry, VT being invoked through electrophysiological studies (EPS) or from the tachycardia induction feature of an ICD. The chip can

be trained using the microprocessor under the control of the telemetry link. After training, the weights need not be changed, and reliable detection of abnormal morphologies can be verified through further EPS.

After configuration, the MATIC system is fully automatic, taking ICEG data as input and producing NSR, SVT, VT and VF classifications as output.

3.3.2 QRS and P Wave Detection

Each channel of the ICEG signal is passed through a simple detector which compares the slope of the differentiated ICEG signal with an adaptive threshold to identify the QRS (or P) wave. After detection of a QRS or P wave, there is an 80 ms period during which no further detection is allowed. This period is called the refractory period and corresponds to the refractory period after depolarisation of the heart (see Section 2.2.1).

The QRS detector, which is implemented on all modern ICDs, provides an additional benefit in that it performs data reduction. The input to the QRS detector is a continuous ICEG and this is reduced to a signal with the same frequency as the heart rate. Furthermore, the output becomes a binary output, the rising edge corresponding to the occurrence of a QRS. The P wave detector reduces the HRA channel data in the same way.

3.3.2.1 Optimisation of QRS Detector

MATIC requires reliable QRS detection to produce good results since both the temporal centering of the QRS complex in the neural network as well as the timing measurements used by the decision tree assume accurate detection. Since it was known that the QRS detector could be improved by changing the decay time constant, maximum threshold value and minimum threshold, the detector was automatically tuned using iterative improvement [WLL88]. Iterative improvement starts by classifying the entire data set and noting the number of incorrect classifications. Each parameter in the QRS detector is then altered randomly and

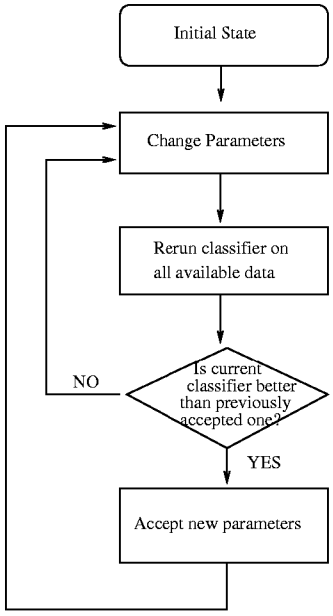

Figure 3.3: Flow chart of iterative improvement algorithm used on QRS detector. The algorithm adjusted the QRS detector by randomly changing them and accepting all good solutions. Algorithm is in an infinite loop and is stopped by user when no further improvement is observed.

the data reclassified using the new parameters. If the new classifier performed better than the old one, the new parameters were accepted, otherwise, another random set of parameters were generated and the process repeated itself until no further improvement was seen (see Figure 3.3).

The success of this optimisation is illustrated in Section 6.4.1.1 where it shown that for a database of 12483 complexes, only 11 classification errors were caused by incorrect QRS detection.

3.3.3 Timing Logic

Single beat classification based on timing decisions is performed in the timing classifier and a classification is made after every R wave. The timing logic classifies the signal based on three parameters, the RR interval of the RVA channel, the PP interval of the HRA channel, and the time between the last P wave and the last R wave (PR interval). A flow chart of the MATIC timing logic is shown in Figure 3.4.

Note that all of the operations in the flow chart involve only very simple, fixed point computations. The computations are either simple comparisons, or multiplications by 2 or 3, all of which can be implemented with at most a single shift and add. The simplicity of the decisions required make the timing algorithm easy to implement and low power.

The timing flow chart is explained in greater detail in the following sections.

3.3.3.1 Ventricular Fibrillation

Ventricular fibrillation is characterised by a fast, chaotic, almost sinusoidal rhythm, no QRS complex being apparent. Although no QRS complex is present, a QRS detector will detect the peaks of the waveform and indicate a very fast heart rate so the criterion for VF is that the RR interval must be less than 220 ms.

Figure 3.5 shows an example of a VF waveform displayed through an ICEG waveform editor. The human classification (made without any knowledge of the MATIC classification) is on the topmost of the screen, and the row of classifications just underneath are those made by MATIC. On the bottom "INFO" line, the number shows the time between the two vertical lines which appear on the screen. These vertical lines (if used) were added to the display after the human and MATIC classifications had been obtained in order to measure timings of interest. The numbers which appear on the middle axes are the sample numbers. The sampling period is 4 ms.

A fast VT is also classified as VF since defibrillation is the correct treatment for this condition. VT is characterised by a fast ventricular rate and also dissociation between the atrial and ventricular channels, the ventricles beating faster than the atrium.

Dissociation occurs when the ventricles and atria depolarise independently. Dissociation is easily recognised when an entire strip of an ICEG is available and ventricular depolarisations can be matched up with the corresponding atrial

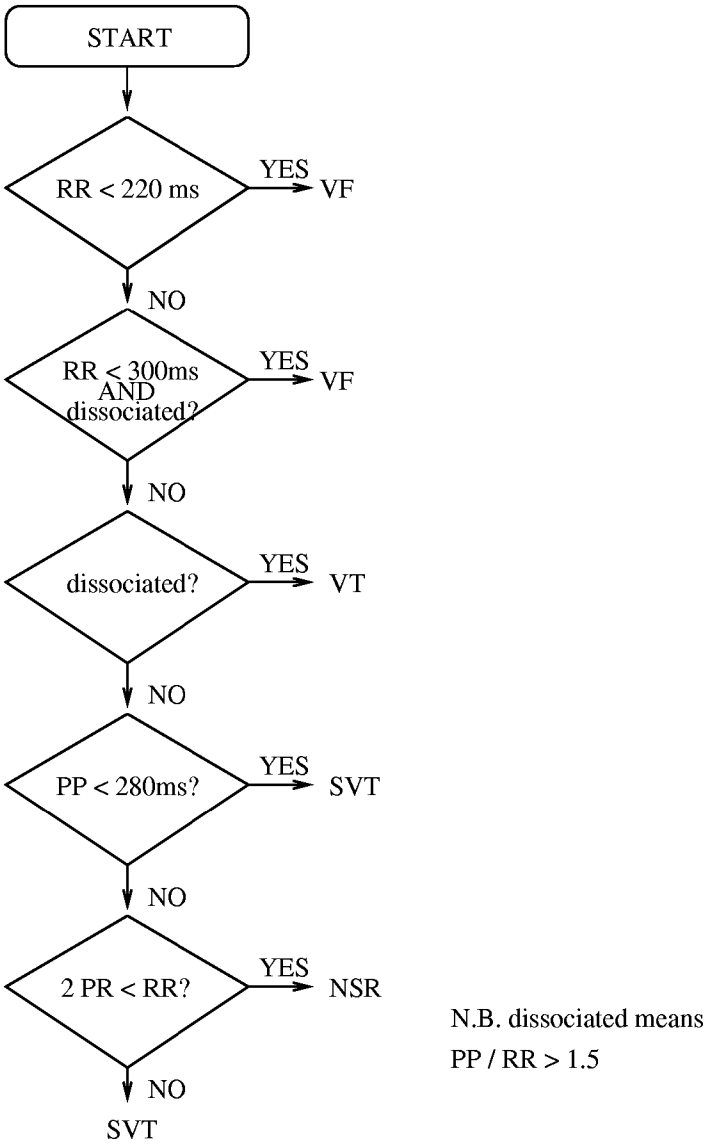

Figure 3.4: Flow chart of timing based decision logic (PP = last atrial PP interval, RR = last ventricular RR interval, PR = time from last P wave to R wave).

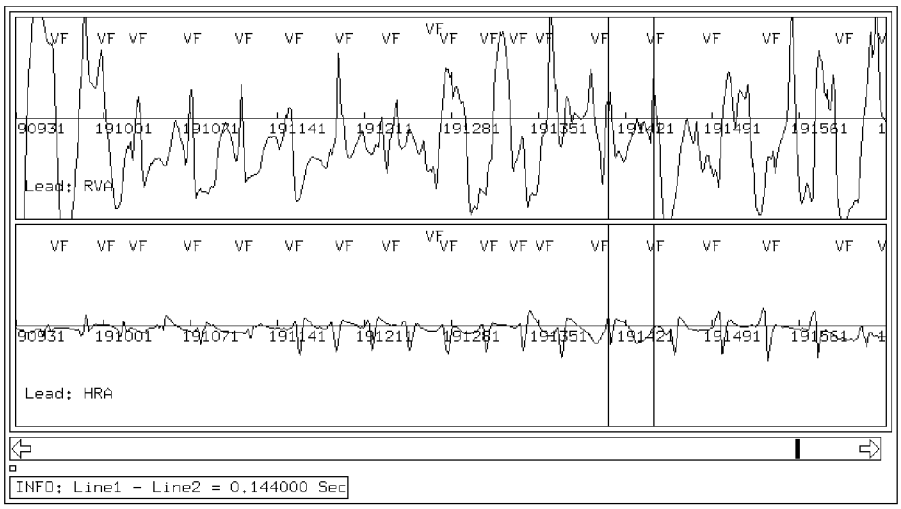

Figure 3.5: Ventricular fibrillation (RR $<$ 220 ms). The human classification appears topmost on the screen, and the row of classifications just underneath are those made by MATIC. The "INFO" line shows the time between the two vertical lines which appear on the screen and the "LEAD" line shows the intracardiac lead from which the signal is recorded. The top trace is RVA and the bottom is HRA. Numbers which appear in the middle waveform are the sample numbers (4 ms sample period).

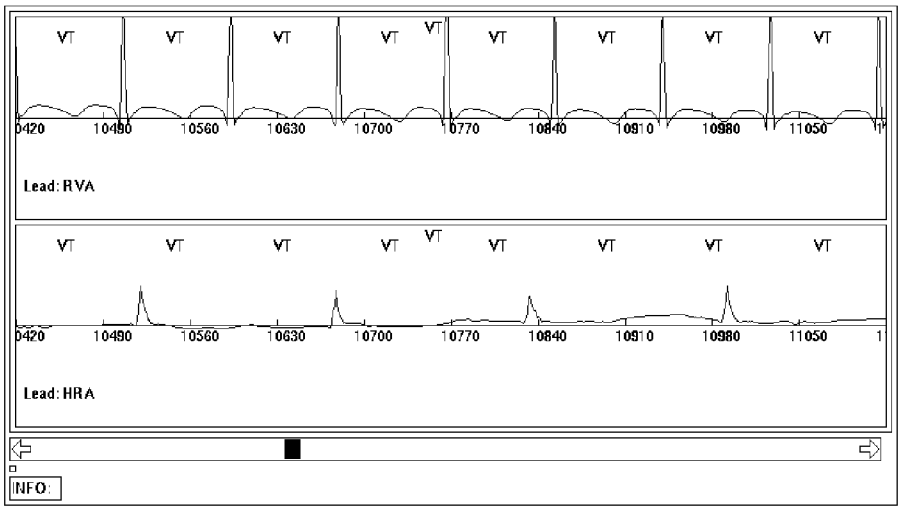

Figure 3.6: Ventricular tachycardia (PP/RR > 1.5).

depolarisation. For single beat classification, a heuristic method is required to determine if dissociation present. In MATIC, the rhythm is considered dissociated if PP/RR > 1.5. That the ventricular rate must be greater than the atrial rate is a nonstandard definition of dissociation.

The "fast VT" rhythm is thus identified if the RR interval is between 200 ms and 300 ms and also dissociated.

3.3.3.2 Ventricular Tachycardia

If dissociation is present and the ventricular rate is faster than the atrial rate the signal is classified as VT (irrespective of heart rate). In Figure 3.6, AV dissociation is present and the tachycardia is classified as VT. Note that although AV dissociation is easy to observe on two leads, it is often impossible to distinguish using only the ventricular lead.

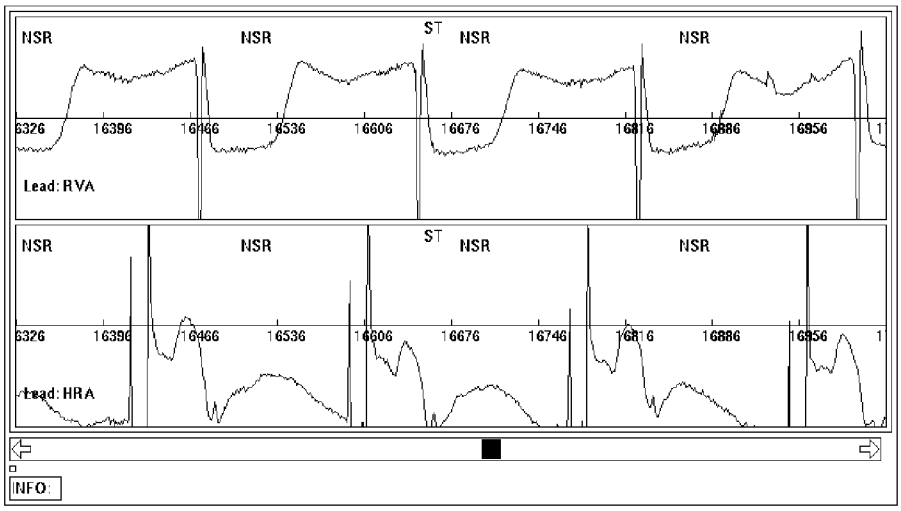

Figure 3.7: Sinus tachycardia (2PR < RR).

3.3.3.3 Normal Sinus Rhythm

The normal sequence of conduction is that the depolarisation of the SA node conducts through the atrium to the ventricles. This causes a P wave to occur first, then a delay while the signal propagates through the AV node, then an R wave occurs. A heuristic method of identifying this sequence of events is if the RR interval is more than twice as long as the PR interval (2PR < RR). This relation is used in MATIC to classify NSR and sinus tachycardia. The ST in Figure 3.7 is recognised correctly as NSR (a superclass which includes NSR and ST) because MATIC considers it to have normal timing, *i.e.* atrial depolarisations cause the ventricular depolarisations.

3.3.3.4 Supraventricular Tachycardia

Tachycardias which originate in the atrium are characterised by a fast atrial rate and thus SVT is identified by the PP interval being less than 280 ms.

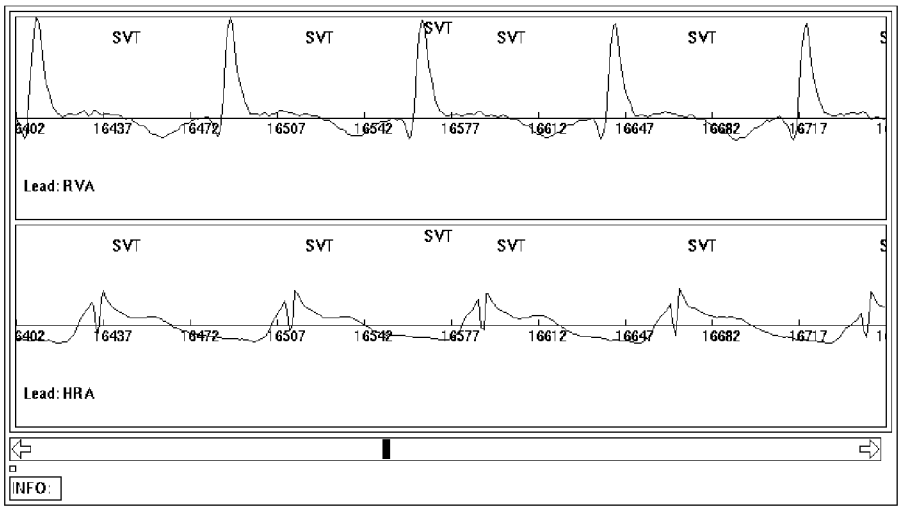

Figure 3.8: Supraventricular tachycardia rhythm (2PR > RR).

In addition, all rhythms that are not identified by the decision tree are classified as SVT. This condition classifies some true SVTs correctly, as well as catching any rhythms which do not fit the NSR, VT and VF criteria, enabling these uncertain beats to be logged by the ICD device. In Figure 3.8, the PR interval is very long and is not classified as an NSR (2PR > RR) and so the default classification of SVT is used.

3.3.4 Neural Network Morphology Classifier

Figure 3.9 shows a normal rhythm and the RVA morphology change which occurs when 1:1 retrograde VT is induced. Note that the MATIC timing algorithm would classify this VT as an SVT since 2PR > RR.

To solve this problem, a neural network is used to identify VT which is not dissociated but does have a different QRS morphology to the patient's NSR. The model used is a three layer perceptron (see Figure 3.10) as described in Appendix A and Equation A.1.

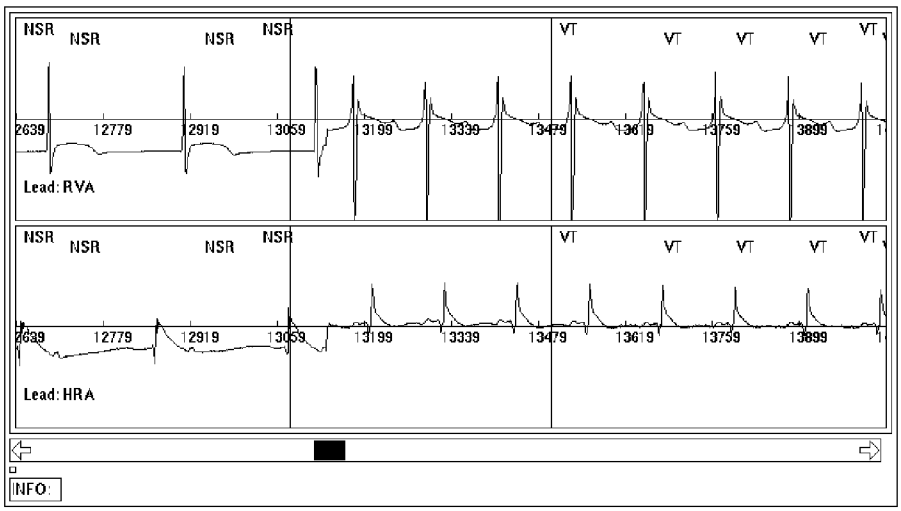

Figure 3.9: Ventricular tachycardia with 1:1 retrograde conduction. Note that a morphology change occurs and the MATIC timing algorithm cannot distinguish this from an SVT.

In a study by Vandepol *et al.* of 86 clinical VT patients [VFS+80], 57 patients had inducible sustained VT. Of these, 91% had morphologies similar to their clinical VTs. Of the 23 with induced nonsustained VT, 78% had similar morphologies. The morphology classifier used in MATIC assumes that the morphology of the arrhythmia induced during electrophysiological studies is the same as the patient's clinical morphology, and Vandepol gives evidence that the assertion is true in the majority of patients.

The network operates in parallel with the timing logic and the output of the neural network is a single number indicating the degree with which the input signal matches the stored patterns. The VT morphologies can be obtained through EPS, via data logging facilities of the ICD, exercise tests or long term intracardiac recordings. Those used in this experiment were all obtained through EPS.

Inputs to the neural network were 10 sampled points (sampling rate 125 Hz) from the RVA channel centered about the R wave of the QRS complex (using

the QRS detector). Originally, 20 inputs were used at the 250 Hz data rate, however, the same 80 ms window size at 125 Hz (*i.e.* 10 inputs to the neural network) was found to produce the same results. Halving of the number of input neurons nearly halves the computational complexity of the neural network. Five hidden units were used as this is the smallest number which produced the correct results. The notation *(a,b,c)* will be used to describe a neural network with a input neurons, b hidden neurons and c output neurons. Therefore the network size used in MATIC is (10,5,1).

The use of morphology for a patient requires 4 NSR and 4 VT 1:1 QRS complexes from that patient (obtained during the configuration procedure described earlier). The training set is formed by making the desired output of the NSR complexes equal to some constant α and the desired output of the VT 1:1 complexes equal to β. The weights used by the MATIC morphology classifier are then obtained by training the neural network on this data set. When ICEG signals are applied to the neural network, a VT 1:1 morphology is present when the output becomes greater than a third constant γ. In all of the experiments that have been conducted, $\alpha < \gamma < \beta$.

Although MATIC has only been used to recognise a specific abnormal morphology, it could also be trained to identify *any* abnormal morphology by training the network to become β during sinus rhythm. MATIC then makes a VT classification if the network output becomes less than γ. This is similar to the approach used in the Bin Area method and the Correlation Waveform Analysis method [LJDM88, TJD90a]. It is felt that matching on a specific VT pattern provides greater specificity than the "any abnormal morphology" method which will produce false positives on morphology changes which are not the intended VT (*e.g.* paroxysmal bundle branch block).

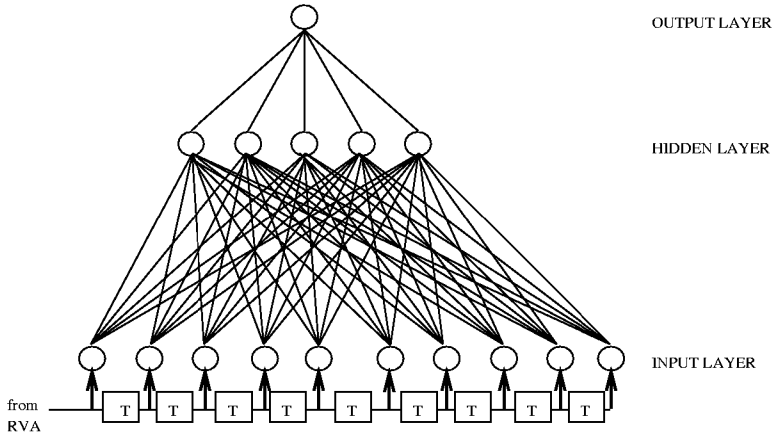

Figure 3.10: Three layer perceptron architecture.

3.3.5 Arbitration Logic

The arbitration logic combines the output of the timing and morphology classifiers to produce a single output class. The arbitration logic is a simple priority encoder and can be described as

$$CLASS = \begin{cases} VF & \text{if timing logic output is VF} \\ VT & \text{if neural network output} > \gamma \\ & \text{AND morphology is enabled} \\ \textit{timing logic output} & \text{otherwise.} \end{cases}$$

The priority arbitration scheme used is the simplest possible but was found to be adequate for classifying all the ICEGs that were available. More sophisticated schemes would be able to provide greater specificity, for example, if the VT 1:1

is known to be always greater than 100 bpm, it could be

$$CLASS = \begin{cases} VF & \text{if timing logic output is VF} \\ VT & \text{if neural network output} > \gamma \\ & \text{AND morphology is enabled} \\ & \text{AND RR} < 600 \text{ ms} \\ \textit{timing logic output} & \text{otherwise.} \end{cases}$$

3.3.6 X out of Y Detector

The output of the arbitration logic is then passed to an "X out of Y" detector. This outputs a final classification only if 5 of the last 6 arbitration logic output classes made by the system are the same, thus improving the certainty of the decision. If this criterion is not met, MATIC does not produce any output (corresponding to a *not sure* result). It is possible that a particular rhythm such as bigeminy can cause the output of the arbitration logic to produce alternating outputs and thus the X out of Y detector will never produce an output. This problem has not yet been addressed as such data was not available at the time of the study. However, it should not be difficult to use simple sequence matching techniques in place of the X out of Y detector in order to recognise such conditions.

The X out of Y function serves to average the classifier decisions over time, removing incorrect classifications due to ectopic beats, fusion beats and artifacts. Note that this technique delays classification for at least 5 QRS complexes from the onset of the tachycardia.

3.4 Performance Considerations

Clearly, the major computational requirements of the MATIC algorithm lie in the morphology classifier, the timing classifier and other logic having only modest computational requirements. In this section, the actual cost of computing the

morphology classification in MATIC is estimated, and then a system architecture for implementing MATIC using a low power morphology coprocessor is presented.

3.4.1 Morphology Classification on a Microcontroller

In a conventional ICD, a microcontroller is used to handle the classification task. In this section, the question of how much would it cost to implement the MATIC morphology classifier on such a microcontroller is addressed.

The Motorola MC68HC711D3 (HC11) [Inc90] was selected as a typical microcontroller which might be used in an ICD. This selection was based on the following features which enable MATIC to be implemented quite efficiently

- The HC11 is implemented in static CMOS so it will have low power consumption.
- The HC11 has single instruction multiply.
- The HC11 has a single instruction 16 bit add.

It is suspected that microcontrollers in existing devices would not have the last two features since present devices would not benefit much from those instructions.

In order to implement a (10,5,1) neural network, 55 multiplies and 55 additions would be required. On the HC11, the inner loop for the multiply–accumulate of the neural network can be expressed as

```
ldaa    0,y     ;  5 load the neuron value from prev layer
ldab    0,x     ;  4 load the synapse value
mul             ; 10 do the multiplication
addd    sum     ;  5 accumulate the result
stdd    sum     ;  4 save the result
iny             ;  4 index the next neuron
inx             ;  3 index the next synapse
```

The number in the comments is the number of cycles required on the HC11 and the total cycles required is 35. A maximum supply current for the HC11 is 15 mA at 5 V with a 2 MHz clock. No 3 V figure for this chip was available, and it will be assumed that it will draw the same at 3 V. If the chip is assumed to draw power only when making a classification, it need only operate at the heart rate, and this will be assumed to be 1 Hz. The average power consumption can now be estimated as

$$\begin{aligned} P &= VI \\ &= 3\ V \times \frac{35 \text{ cycles} \times 55 \text{ synapses}}{2,000,000 \text{ Hz}} \times 15 \text{ mA} \times 1 \text{ Hz} \\ &= 43\ \mu\text{W} \end{aligned}$$

This value does not include overheads required to perform analogue to digital conversion of the input ICEG, assumes that the HC11 draws no power when it is not being used, omits code for initialisation, loops, the neuron nonlinearity, power required by external RAM devices *etc.* On the other hand, it is expected that the microcontroller inside an ICD would be better optimised for low power dissipation than the HC11.

The power consumption figure of 43 μW for a microcontroller implementation of a neural network should be compared with that of the Kakadu neural network chip described later in Section 6.5.2.2. In that section, it is shown that Kakadu can operate with a power consumption of less than 25 nW.

3.4.2 System Architecture

An algorithm for arrhythmia classification of ICEGs is not useful in an ICD if it cannot be implemented within the constraints imposed by the system. If the morphology classifier of the previous section was added to an ICD, it would require an additional 0.75 Ah of battery life over 6 years. Since a typical ICD battery is 4.1 Ah (see Section 2.5), this would add approximately 20% to the power consumption – an unacceptably large amount. In this section, the architecture

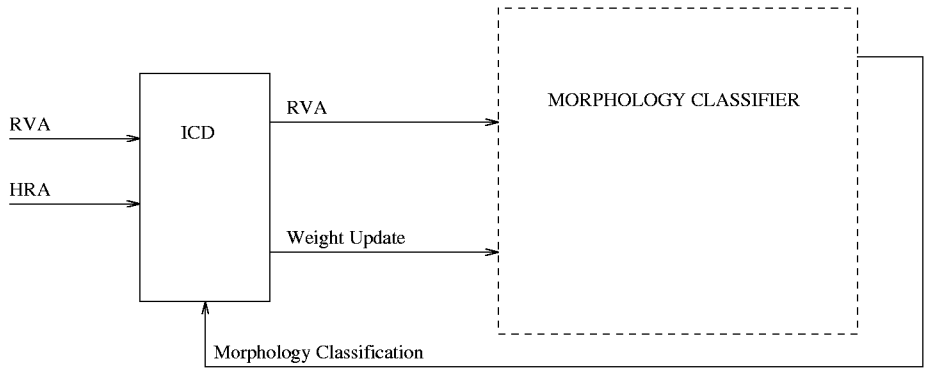

Figure 3.11: Block diagram of the system architecture of a MATIC implementation in an ICD.

of an implementation of MATIC which would have much lower consumption is detailed.

An assumption will be made in the following discussion that it is possible to design a custom VLSI neural network chip which has much lower power consumption than that of the microcontroller of the previous section. In Chapters 5 and 6, this assertion will be shown to be true, and in fact, it is possible to improve on the power consumption of the microcontroller neural network by several orders of magnitude.

The architecture is one of a coprocessor model. An existing ICD is augmented by a morphology chip which performs the computationally expensive task of morphology classification. With this architecture, the programmability of the microprocessor in the ICD can be used to implement the timing portion of MATIC in a flexible fashion and parameters can be changed in a patient dependent manner if necessary. A block diagram of this architecture is shown in Figure 3.11.

The microprocessor inside the ICD device orchestrates the classification process, implementing all of the MATIC algorithm except for the morphology classifier. The morphology classifier is implemented in a neural network chip which operates in parallel with the microprocessor. The morphology classifier obtains an input from the RVA channel and passes this through an analogue shift register to obtain a window of samples. These form the input to the neural network and a classifier using the structure of Figure 3.10 is used so morphologies can be analyzed by the neural network. The microprocessor reads the output of the morphology classifier and processes this information according to the arbitration logic described in Section 3.3.5, followed by the X out of Y detector of Section 3.3.6 whose output is the final MATIC classification.

3.5 Discussion

The timing part of the MATIC algorithm is similar in principle to the decision tree classification of two channels by Arzbaecher *et al.* [ABJ+84]. Although there is much evidence to support a sudden onset detector [WM86], it was not included in MATIC because all the arrhythmias in the database were induced via EPS and thus normal transitions are not represented.

Morphology and timing in a two lead system was used by Lin *et al.* (described in Section 2.4.8). Lin used correlation waveform analysis for template matching and performed this on both atrial and ventricular channels. The template was created from NSR rhythms and a morphology change was said to have occurred if the current QRS did not correlate well with the NSR. It is felt that greater specificity is obtained by the method used in MATIC whereby both the NSR and abnormal complexes are stored.

As with all ICEG classification systems, there are arrhythmias on which MATIC will fail. The aim in developing MATIC was not to try to produce

a perfect classifier, but to produce a simple system suitable for use in an implantable device, which overcomes some of the limitations of classifiers used in present ICD devices. Some situations in which MATIC might fail include

- Any form of AV block
- Bigeminy and trigeminy
- VT 1:1 with morphology different to that induced during EPS or exercise tests
- VT 1:1 with similar morphology to NSR
- Paroxysmal bundle branch block

Each of these conditions could cause misclassification because of MATIC's simplistic view of timing and morphological considerations during classification. In these cases, it may be possible to adjust the parameters in the timing decision tree or change the arbitration and post processing logic to suit a particular patient, thus using MATIC as a patient dependent classifier.

3.6 Summary

The design of the MATIC tachyarrhythmia classifier for ICEGs was presented in this chapter. This classifier follows the approach of mimicking a human expert, making assessments based on both timing and morphology features of the signal. It was shown that implementing the morphology classifier using the microprocessor in an ICD would consume approximately 20% of the power of the entire ICD. For this reason, an architecture was presented which uses the microprocessor inside the ICD to implement the timing part of MATIC in a flexible manner and uses a VLSI implementation of the morphology classifier to maintain low

power consumption. This architecture enables one to keep the benefits of a computationally expensive algorithm yet consume little power. The design of such a morphology classifier is presented in the next chapter.

Chapter 4

Bourke Prototype

4.1 Introduction

The morphology recognition ability of MATIC described in the previous chapter makes it both powerful and flexible because such logic can be tuned to recognise a range of arrhythmias which could otherwise not have been identified. Patient dependent morphology capture could be used not only to detect such problems as ventricular tachycardia and paroxysmal bundle branch block, but it can also be used by the ICD to identify complexes which have a morphology change so that they can be stored and examined later by a cardiologist.

The software implementation of a neural network morphology detector is computationally expensive and even on existing computers, simulations of artificial neural networks require large amounts of computing time and hence power. Fortunately, the inherent parallelism and simplicity of the processing elements within an artificial neural network make it ideally suited to implementation in VLSI.

The chapter begins with a description of previous work on the implementation of ANNs in VLSI, concentrating on the issue of synapse design since most of the area of a neural network chip consists of these elements. This is followed by a discussion of the design considerations for a low power neural network classifier

	Analogue	**Digital**
Precision	Low	High
Area	Small	Large
Speed	Fast	Slow
Power	Low	High
Flexibility	Poor	Good
Clocking	Asynchronous	Synchronous
Design Tools	Poor	Good
Noise Immunity	Poor	Good
Drift Immunity	Low	High

Table 4.1: Comparison between digital and analogue implementations.

chip which would be suitable for implementing the morphology recognition part of the MATIC algorithm. The detailed design of a prototype neural network, called Bourke, is then described. Bourke was a test chip which included a small neural network and a bucket brigade device. Since Bourke was small in size, it was possible to verify its design by analogue simulation before fabrication and the techniques used to perform this simulation are also described. The chapter concludes with a presentation of the results obtained by applying the fabricated chip to the XOR problem.

4.2 Previous work on VLSI Artificial Neural Network

4.2.1 Analogue Versus Digital

In designing a low power neural network chip, several important architectural issues immediately come to mind. The first question is whether to implement the network as a digital or analogue design. Analogue techniques have the desirable features of small area, low power and high speed. However, they suffer from low precision and low noise immunity. The digital approach offers good noise and drift immunity, arbitrarily high precision arithmetic, a greater degree of chip programmability and a more conservative design strategy (see Table 4.1).

An analogue approach was selected for two reasons. Firstly, a lower power consumption was expected and in addition, an analogue to digital conversion of the ICEG waveform is avoided since the ICEG signal can be processed entirely in the analogue domain.

4.2.2 Previous Analogue VLSI ANN Implementations

Neural networks are notorious for being computationally expensive, however, the natural parallelism and simple building blocks of ANNs make them an ideal candidate for VLSI implementation. The main stumbling block encountered when implementing an analogue neural network lies in the choice of the analogue weight storage technique.

Horio *et al.* [HN92] have proposed the following set of "necessary conditions" required of an analogue memory in a neural network

1. The analogue information in a memory should be maintained for a sufficiently long time (storage).

2. The memory should have enough resolution for analogue computation (resolution).

3. The information in a memory must be linearly controllable (programmability).

4. The information should be easily, continuously and asynchronously updated and accessed without destruction of the stored information or interruption of the network processing (disruption).

5. The memory should be as small as possible (size).

6. Read and write operations should be fast (speed).

As the analogue storage device that satisfies all of these conditions has not yet been invented [HN92], with the above considerations in mind, and adding the

	Capacitor	**EEPROM**	**MDAC**
Storage	Volatile	Nonvolatile	Nonvolatile
Resolution	10 bits	4–8 bits	6 bits
Programmability	Linear	Exponential	Linear
Disruption	Ok	Ok	Ok
Size	33.4 syn/mm^2	$>$ 150 syn/mm^2	15.6 syn/mm^2
Speed	Fast	Slow	Fast

Table 4.2: Comparison of weight storage techniques for analogue synapses. The size entry is the number of synapses on the chip divided by the area. When more than one design is cited, the figure represents an average. For the EEPROM, the die size is not known and a 64mm^2 die is assumed. For MDAC designs, the Graf *et al.* designs were not included in the "size" entry computation since their low analogue resolution distorts the result.

extra constraints of low power operation and a standard CMOS technology, a survey of previous work grouped by storage technique will be made. It may be useful to refer to Table 4.2, a summary of the techniques, while reading the following sections.

4.2.2.1 Charge on a Capacitor

Temporary weight storage can be achieved in standard CMOS by using the charge on a capacitor. Although the capacitor can be made very small in area, it must be periodically refreshed to replace charge which has leaked away. In its simplest form (see Figure 4.1), the synapse value is written by applying a voltage representing the desired synapse value to Vin and asserting WR.

Schwartz *et al.* [SHH89] have used a twin capacitor memory to implement a neural network chip having 1104 10 bit synapses on a 6×3.5 mm die. Schwartz uses the C_{store+} and C_{store-} capacitors of Figure 4.2 to store the synapse values and charge injection is used to program them by causing the transfer of small amounts of charge between a pair of transistors. To initialise the synapse, TA is turned on in order to charge the capacitors up to a convenient voltage V_{init} (typically VDD/2). The weight value can be increased by first switching on TC and TP, turning off TP, turning on TM and then slowly turning off TC.

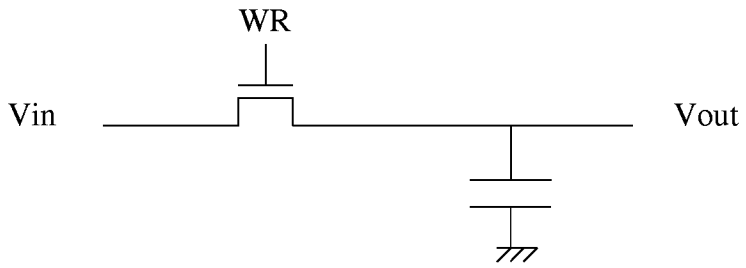

Figure 4.1: Capacitor weight storage.

This causes mobile charge in the channel of TC to diffuse into the minus node, lowering its voltage. Schwartz's capacitors were 2000 μm^2 in area and using this method, the leakage after 100 seconds was approximately 1% of the stored value.

In a design completed at the Jet Propulsion Laboratory [EDT89, ET90] capacitor storage refreshed from digital memory through a time–multiplexed DAC was used to implement 1024 11 bit synapses on an 8.2×8.8 mm die. Each capacitor was 2.4 pF and leaked at a rate of 3.3 mV/s at room temperature. The storage element is two 5550 μm^2 transistors which are organised in a differential fashion so that the first order leakage effects are cancelled.

The pulse stream method [MS88] implements synapses which modulate the pulse density of a stream of pulses sent to it by a neuron. It stores the synapse value dynamically on a capacitor. Unfortunately, for low power applications, too much switching occurs wasting precious power.

In terms of Horio's necessary conditions of an analogue memory, capacitors have many good points: they can be programmed with high resolution, they are linearly controllable, can be written using a simple DAC and can be made relatively small. However, because of charge leakage in the capacitor, the necessity

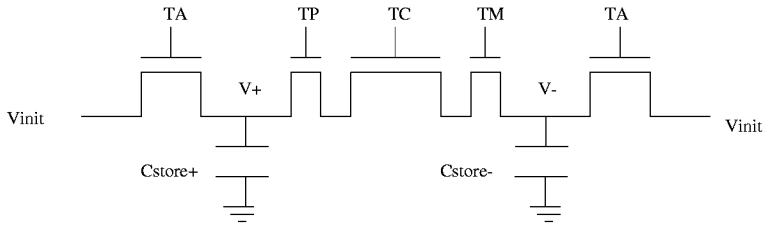

Figure 4.2: Capacitor weight storage of Schwartz *et al.* [SHH89].

of refreshing the weights adds complexity to the design and increases the power consumption of the overall system.

4.2.2.2 EEPROM

Floating gate technology such as that used for electrically erasable programmable read only memories (EEPROMs) is a true nonvolatile analogue storage technique and has been used for analogue storage in a synapse [Kra90, HTCB89, LSY91, DS92].

This technique uses charge stored on a floating gate, the floating gate being surrounded by a layer of silicon dioxide to prevent charge leakage. Electrons can be injected or removed from the floating gate through the insulating oxide via Fowler–Nordheim tunneling [Sze81]. In order to achieve this tunneling, a high electric field must be produced within the oxide. The amount of charge injected on the floating gate varies exponentially with the programming pulse width and programming pulse voltage.

EEPROMs are able to produce a stored charge with zero static power consumption and they can be close to the minimum feature size making them very

attractive for a low power storage scheme. However, the exponential programming characteristic makes programming difficult and it is often being performed in several steps. The device also has different write and erase characteristics and requires complicated control circuitry to program.

Holler *et al.* [HTCB89] have produced the "ETANN" neural network chip which is a 10240 synapse analogue neural network chip with 6 bit resolution. In 1 micron EEPROM technology, the synapse area was 2009 μm^2.

EEPROM synapses can be programmable to 8 bit accuracy [HTCB89, Car89] and are very small in size. These devices cannot be fabricated using a standard CMOS process and also suffer from a short term drift characteristic of the floating gate which is thought to be due to charge trapping in the oxides [DS92]. However, the major disadvantage of EEPROMs is that it was estimated that for weight storage of the order of 15 years, only 4 bit accuracy would be expected [HTCB89].

4.2.2.3 MDAC

Analogue weight storage can be achieved by storing weights in a digital form and then performing a digital to analogue conversion to obtain the analogue value. This method has the benefits of a digital interface, nonvolatile storage and devices can be fabricated with a standard CMOS process. A multiplying digital to analogue converter (MDAC) performs the synapse function that is desired.

Raffel *et al.* at MIT Lincoln Laboratory [RMB+87] have used a 5 bit multiplying digital to analogue converters to implement a matrix of 1024 synapses using an active area of 28 mm^2 in 3 μm CMOS.

Hollis *et al.* [HP90] have constructed a neural network chip with 44 7 bit synapses on a 2×3 mm die using the MDAC shown in Figure 4.3. The DAC is constructed from ratioed transistors and this serves to produce currents in powers of two which can be connected with the pass transistors B1 to B4. A power consumption of 9 μA per synapse was achieved with this design.

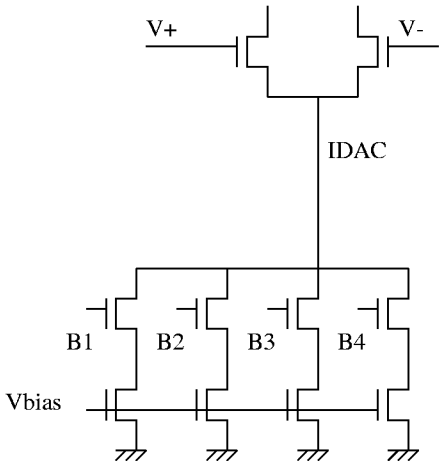

Figure 4.3: MDAC weight storage method of Hollis *et al.* [HP90].

A neural network chip which has 4416 synapses on a 6.7×6.7 mm die was designed by Graf *et al.* [GJH88]. Graf managed to accommodate a large number of synapses on a chip by using the trinary scheme shown in Figure 4.4. This has the advantage of only requiring binary multiplication which can be implemented in a very compact fashion. The disadvantage, of course, is that only a very limited weight resolution is available.

Another design by Graf *et al.* [GH90] implements 32,000 single bit connections on a 4.5×7 mm die. The chip is highly reconfigurable and can be programmed to implement single or multi–layer networks with binary or analogue connections. Although the synapses are single bit, analogue connections are possible because of a *multiplier and neuron concatenate* circuit. The design works from a 3 V or greater supply and has a current consumption of 1 μA per synapse.

The CCD technique [Chi90] has an input CCD which is multiplied by an array of charge domain MDACs. Chiang *et al.* have implemented a neural network chip with 144 MDACs which are time multiplexed to produce 2016 6 bit synapses on a 7×7 mm chip. A special CCD process is required to fabricate these devices.

The MDAC technique has good weight retention since digital storage is used

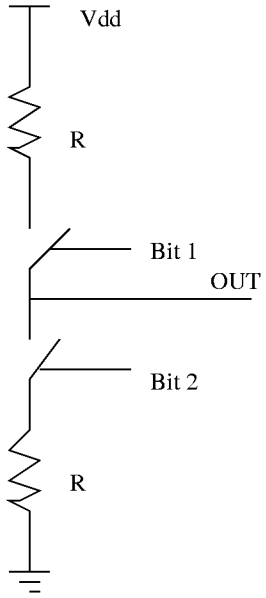

Figure 4.4: Ternary weight storage method of Graf *et al.* [GJH88].

and the weights are easily interfaced to other digital systems. One disadvantage, however, is that the area required by MDAC synapses is possibly the largest of the three technologies discussed.

4.3 Analogue Neural Network Design Concerns

The problems encountered when implementing a low power artificial neural network in analogue VLSI are numerous. Two restrictions fundamental to the choice of implementation technique for use in ICDs are that the chip must have low power consumption and also that the technology must be standard CMOS.

In the design of the neural network chip, it was obvious that it must be built from two basic cells, synapses and neurons. This lead to the following three design concerns

- How can we implement the synapse?
- How can we implement a low power neuron?

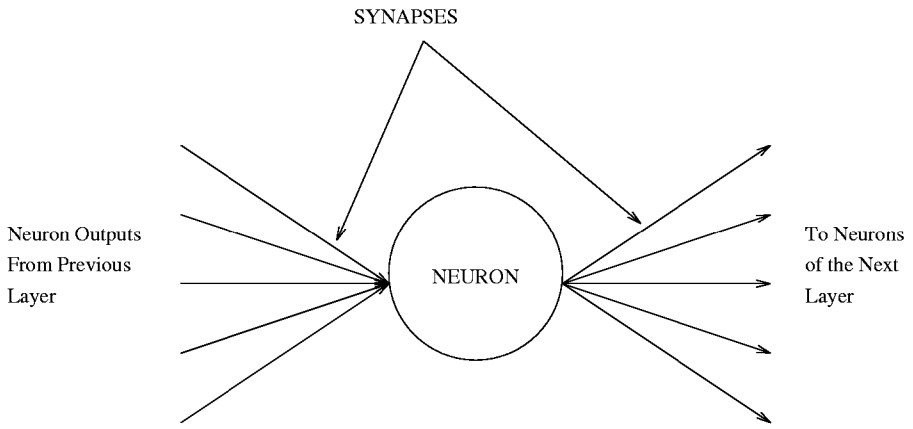

Figure 4.5: Neuron and synapse operation. Neurons are connected together by synapses.

- How can device mismatch, which is especially poor when low bias currents are used, be overcome?

The first question has the greatest influence on the architecture of the chip because synapses form the majority of the chip and thus their implementation most affects the chip's performance and size. These issues are discussed in the following sections.

4.3.1 Synapses

As shown in Figure 4.5, a neural network consists of neurons interconnected via synapses. Each neuron takes the sum of all of the synapse outputs which connect to it, and must produce a single value output.

It is logical to implement synapses as transconductance devices and neurons as transimpedances. This is because the output of synapses must be summed and this can be done using Kirchoff's current law. The output of neurons must fanout to the inputs of all synapses in the next layer and so a voltage representation of this is the most suitable.

A synapse must perform two functions. It must store weights corresponding to the synapse value, and it must perform multiplication of its input with the stored weight. A Gilbert multiplier is used and this has the added advantage of being a transconductance device.

For the analogue weight storage, an MDAC approach (see Section 4.2.2.3) was selected as an analogue storage device because it had the following advantages

- Can be fabricated in standard CMOS.
- Operation at very low bias currents is possible.
- Has a digital interface for weight update.
- Has analoguc inputs.
- The digital storage registers are static CMOS and hence dissipate negligible power.

4.3.2 Neuron

Transfer functions with nonlinear squashing characteristics (such as tanh) are normally employed for neurons. In a low power system, where the neuron input current can be of the order of ten nanoamps, a high impedance of the order of 1 MΩ is required. This is hard to implement in standard MOS because diffusion and polysilicon do not have the high resistance necessary, and an active circuit with the desired transfer characteristic is hard to design. If on–chip neurons are used, a method of measuring the activation of at least the output neurons is required for training, and this requires buffers to drive the signals off–chip.

A possible solution to this problem is to implement the neurons using off–chip resistors. Resistors have very low power consumption, allow all of the neuron activities to be monitored and precise control of the neuron's gain can be achieved by changing the resistance.

However, a resistor has a linear characteristic which, at first glance, appears unsuitable. This problem was addressed by implementing the nonlinear characteristic required by the neural network in the synapse instead of the neuron. Using this technique, the nonlinearity of the Gilbert multiplier is used to an advantage and the resulting neural network transfer function is discussed in Section 4.5.1.

Possible disadvantages of using off–chip neurons are that since the currents must travel through pins, pin leakage may affect the circuit and also, for larger networks, the number of pins required may become excessive.

4.3.3 Device Mismatch

Device mismatch causes nominally identical transistors within a chip to have different gains and threshold voltages. It is caused by processing variations across the chip. At low currents, the dominant term is threshold voltage mismatch, and a current mirror typically has an error of $\pm 20\%$ [Mea89]. Furthermore, transistor matching cannot be readily simulated and so until familiarity with a foundry is attained, it is very hard to judge absolute matching characteristics of transistors.

Efforts were made to layout the current source and MDAC so as to minimise potential matching problems. The design strategy allowed for the reference bias current to be varied so that its influence on the MDAC linearity could be investigated. Fortunately, the inherent design of a neural network offers some degree of compensation for synapse nonlinearity through training.

4.4 Bourke Chip

The Bourke chip (see Figures 4.6 and 4.7) was the first analogue neural network chip designed for this thesis. Bourke contains a small (3,3,1) neural network as well as a test bucket brigade device (BBD) and some other test circuits. In this section, only the design and simulation of the neural network portion of the chip

will be described. The design and results of the bucket brigade portion appears in Appendix B.

Bourke implements an artificial neural network based on the three layer perceptron model [Lip87]. A block diagram of the chip is shown in Figure 4.8. The chip takes voltage inputs and passes them through the first array of synapses to produce three pairs of hidden layer currents. These differential currents are converted to voltages using resistors that are external to the chip. The same nodes are used as voltage inputs to the next layer which produces output currents which are again converted to voltage outputs by the third neuron layer.

The main blocks of the chip are two synapse arrays, a current source and weight addressing circuitry. The synapses' digital to analogue converters are binary weighted current sources controlled by digitally stored weights. A common current source is used to supply bias voltages to all DACs. The circuit can be operated over a wide range of bias current, the linearity of the DACs improving as the bias increases and hence a tradeoff between power dissipation and DAC linearity can be made.

Although inputs to the neural network are analogue, synapse values are written digitally. This enables configuration of the chip to be performed digitally but keeps the actual signal processing in the analogue domain. The synapse array appears as a 12 word RAM with a 6 bit word size. Synapses are addressed by row and column through a pair of multiplexed row and column shift registers.

Schematic capture was entered using the Daisy Systems Corp. program "ACE" and their "Apex" simulator (similar to Spice) was used for analogue simulation. "Magic" from the University of California at Berkeley was used for mask layout, and Magic's "Irsim" was used for digital simulation. Bourke was a full custom VLSI design and was fabricated using Orbit Semiconductor's 1.2 μm double metal, single poly nwell process on a 2.2 $\times$ 2.2 mm die.

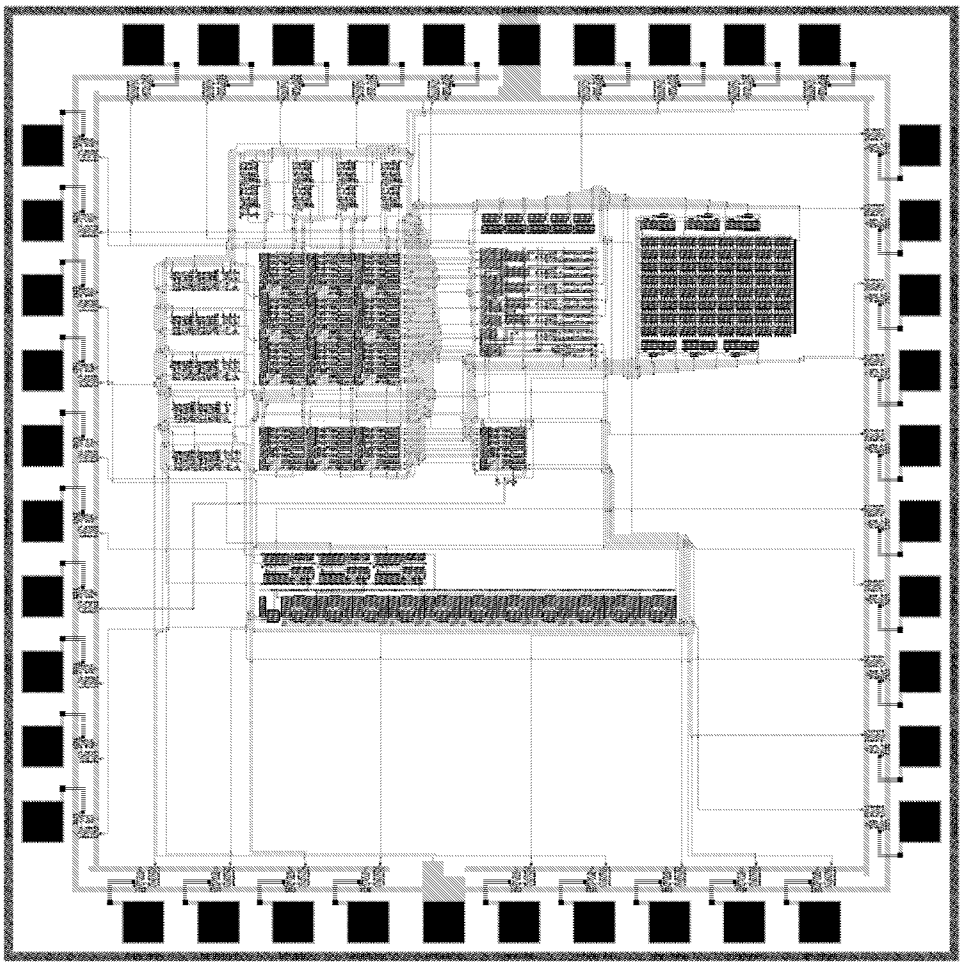

Figure 4.6: Bourke test chip.

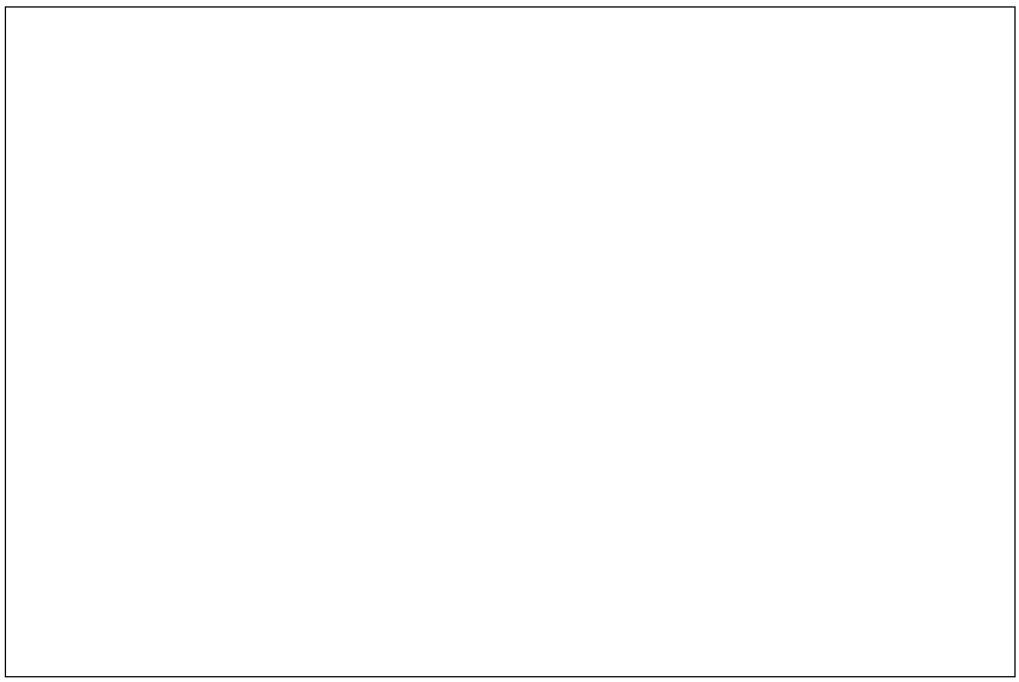

Figure 4.7: Photomicrograph of the Bourke chip.

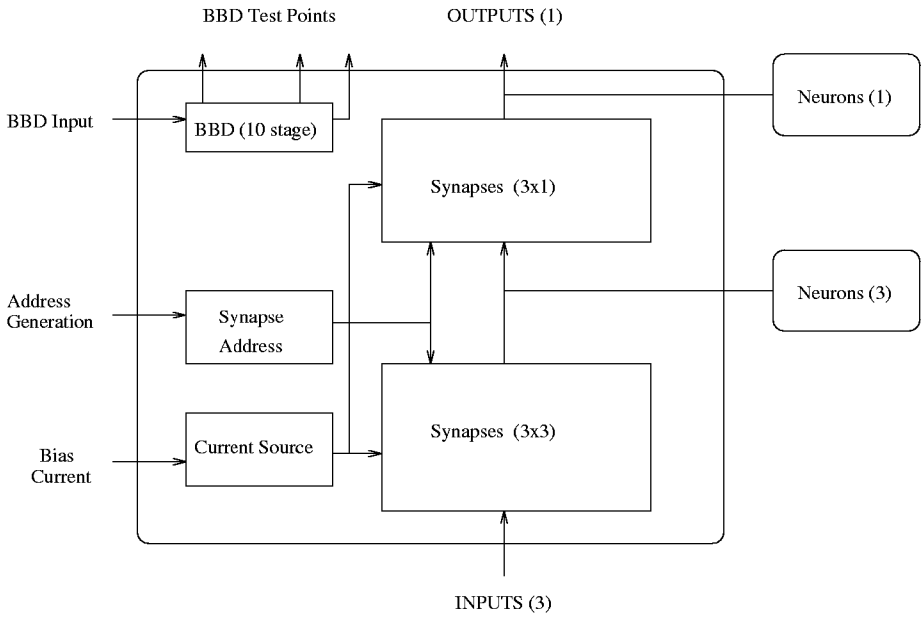

Figure 4.8: Block diagram of the Bourke chip.

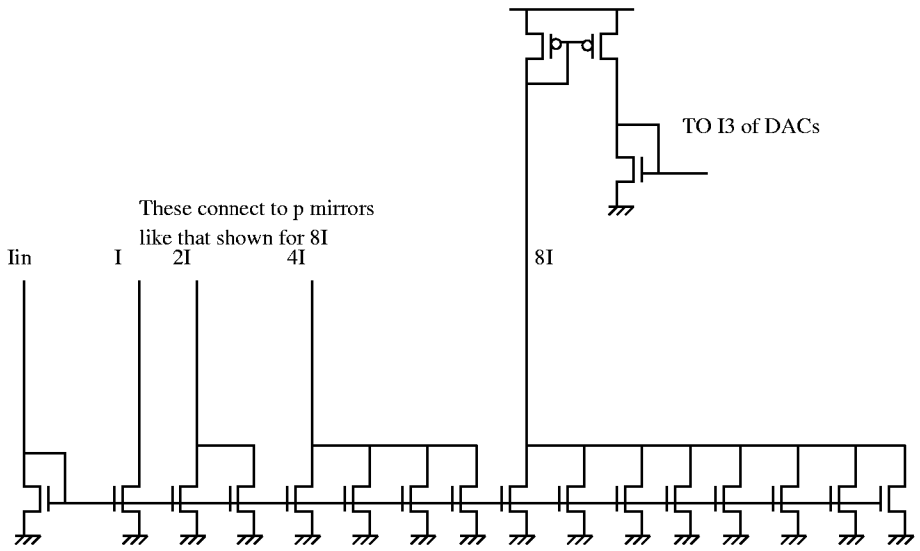

Figure 4.9: Current source circuitry (4 bits shown).

4.4.1 Current Source

The current source (Figure 4.9) is constructed by summing unit current sources and provides biases to enable the transistors inside the DACs to produce currents in powers of two.

For transistors with uncorrelated parameter variations, summing unit current sources improves the matching by a factor of $\sqrt{N}$. Correlated parameter variations such as changes in doping or oxide thickness are addressed by arranging the current sources in a common centroid configuration [BDST91]. Large (553 μm^2) transistors are used for the current source although smaller (81 μm^2) transistors are used inside the digital to analogue converters (DACs) in order to keep the total synapse area small.

The bias current is controlled by an off–chip current or voltage. Since all of the currents feeding the synapses are derived from this single input, the entire circuit can be switched off by making *Iin* equal to zero.

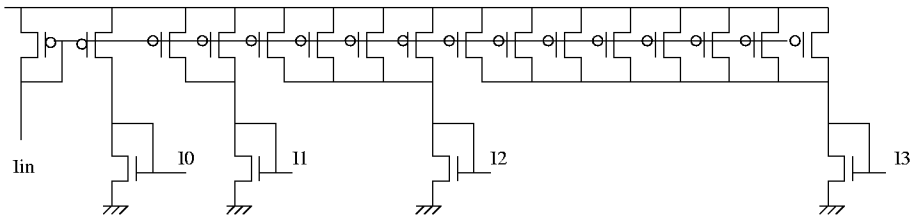

Figure 4.10: Improved current source circuitry which could be used in future designs (4 bits shown).

4.4.1.1 Improved Current Source

Although the current source just described was used in both the Bourke and Kakadu chips, it could be improved in future designs by replacing the n–type unit current source array with a p–type array as shown in Figure 4.10. This removes the need for an additional mirror and would improve the linearity of the current source as well as the obvious reduction in area.

4.4.2 Synapse

The synapse is composed from a weight storage register, a linear DAC and a Gilbert multiplier. The circuit diagram of the synapse is shown in Figure 4.11.

As synapses are the most numerous elements in a neural network chip, the size of the network that will fit in a given area is predominately determined by their dimensions. Although small synapses are required, the matching of critical transistors within the synapse is proportional to the square root of the transistor area and so these transistors should be made as large as possible. A compromise was reached by selecting moderate size transistors (81 μm^2) for the I0 to I4 mirrors within the synapse.

Storage of the synapse values is achieved using registers, the value of which are

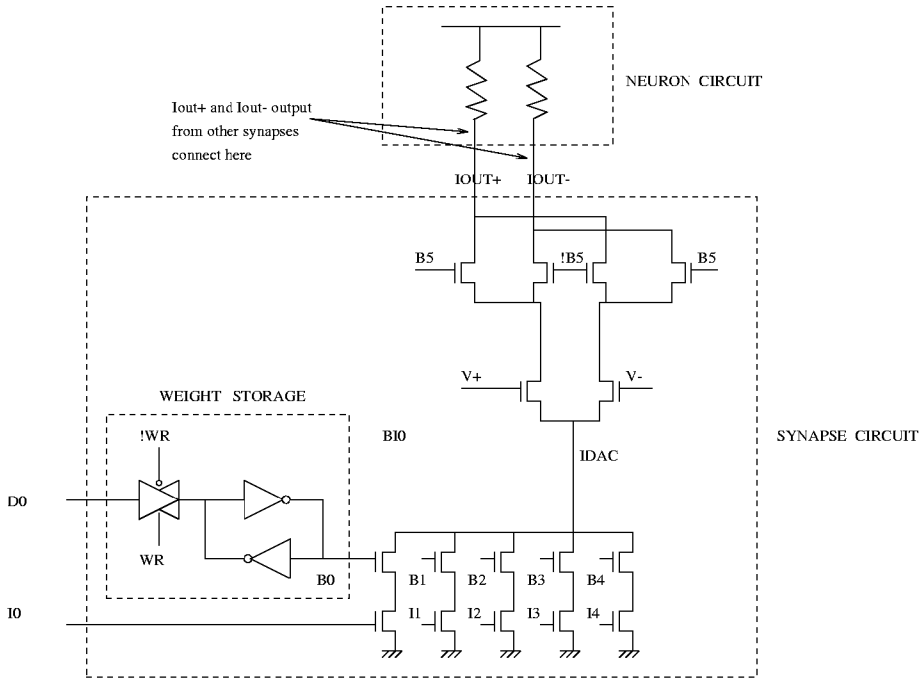

Figure 4.11: Synapse and neuron circuitry.

converted to analogue values via a DAC. This allows analogue signal processing techniques to be used whilst maintaining the advantages of digital weight storage.

To reduce synapse area, the registers were designed to be as narrow as possible since each register contains 6 flip–flops. Each flip–flop comprised of a transmission gate and a pair of cross–coupled inverters as shown in the weight storage block of Figure 4.11. The dimensions of the devices were carefully selected to allow the driving signal to overpower the inverters. The design has a minimum of transistors, a compact synapse layout and negligible static power dissipation.

The DAC is constructed through current summing. Each bit of the DAC is controlled by a pass transistor which can be turned on or off depending on the value stored in the (static) input flip–flop (B0–B4). I0–I4 are voltages taken from the current source which serves to provide bias currents in powers of two. The Bourke chip is programmed by writing values into the flip–flops of the DAC.

The DAC is connected to a Gilbert multiplier to form a synapse. The multiplier has a pair of voltage inputs, a pair of current inputs (from the DAC) and a pair of current outputs. The transfer function of this multiplier is given by the relation

$$I_{out+} - I_{out-} = \begin{cases} +I_{DAC} \tanh(\frac{\kappa(V_3 - V_4)}{2}) & \text{if B5} = 1 \\ -I_{DAC} \tanh(\frac{\kappa(V_3 - V_4)}{2}) & \text{if B5} = 0 \end{cases} \quad (4.1)$$

The multiplier is linear with the current input I_{DAC} and nonlinear to the neuron voltage inputs $V3$ and $V4$. This is the desired situation for if they were reversed, the tanh function would only serve to compress the range of weight values available and would not allow nonlinear problems to be solved. The DAC only produces positive values and current switching circuitry controlled by B5 enables the output to be changed in sign if a negative weight is desired. The area of a synapse is 106×113 μm and a layout is shown in Figure 4.12.

4.4.3 Neuron

The neuron is simply a large pull–up resistor (Figure 4.11). For all of the experiments conducted on both prototypes, 1.2 M resistors were used. This configuration allows the current sink output of the synapse to produce a voltage proportional to its output, and two resistors are required to produce a differential voltage output (Figure 4.11). The resistors are provided off–chip in order to allow easy control of the impedance and transfer characteristics of the neuron. These neurons also serve as convenient test points for the chip. Static RAM processes offer a high resistance polysilicon which would enable such neurons to be placed on chip.

4.5 Bourke Training

A full analogue simulation of the Bourke chip (using Apex) was trained on the XOR problem prior to the chip being sent to fabrication by training a range of

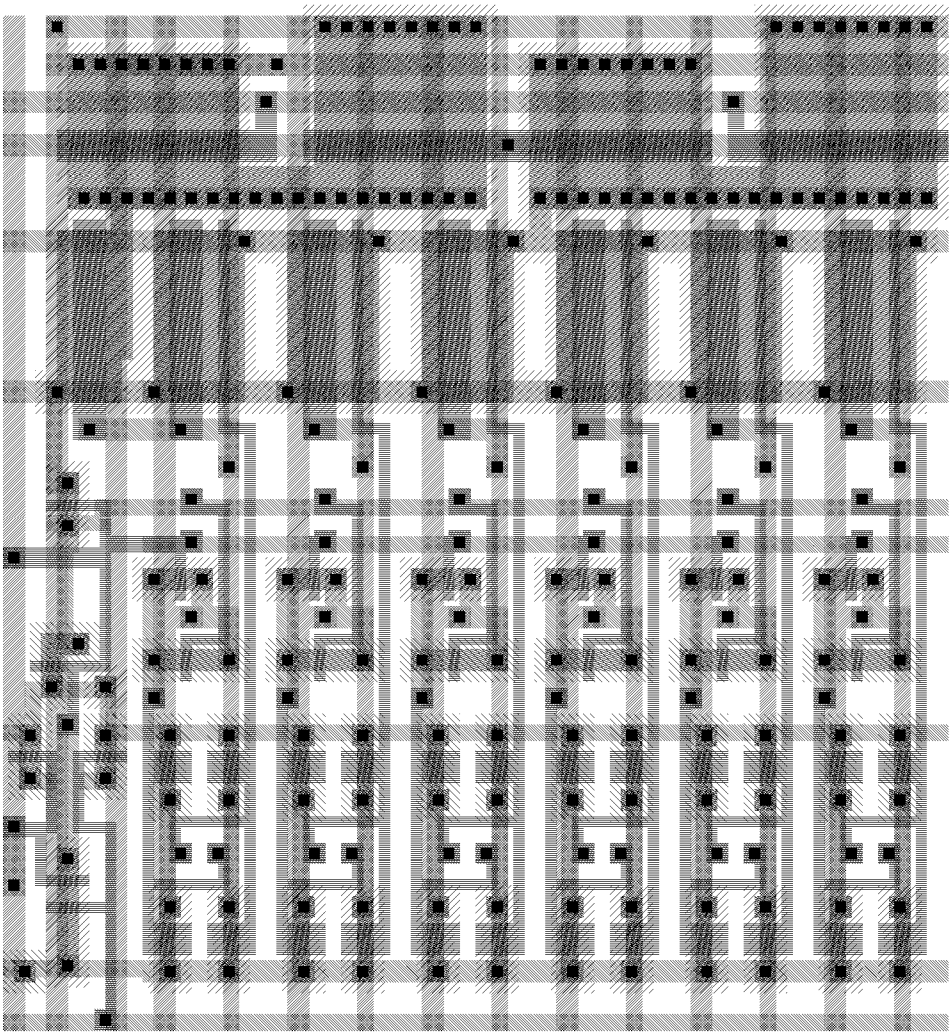

Figure 4.12: Synapse layout.

increasingly sophisticated models of the chip. The weights obtained from training simpler models were used as starting values for more sophisticated models. This method of hierarchical training was successfully used to reduce the number of passes of full analogue simulations required.

In the following sections, synapse modeling using mathematical models is first detailed, then the method of training simulations is described. Next an example of training Bourke on the XOR problem using full analogue simulation is presented and the section concludes with the results of training the actual Bourke chip to perform XOR.

4.5.1 Synapse Modeling

The synapses serve to multiply the inputs by a programmable weight, and all synapse outputs in a row are summed using Kirchoff's current law. The transfer function of the Bourke chip can be described by the following equations

$$u_i = \sum_{j=1}^{N_l} w_{ij} \tanh(\frac{\kappa a_j}{2}) \tag{4.2}$$

$$a_i = \alpha u_i \tag{4.3}$$

where w_{ij} represent the synapse connections to neuron i, u_i is the summed output of the synapses, a_i is the neuron output, α is the neuron gain (a 1.2M resistor gives a value of 1.2×10^6), κ is a constant, l denotes the lth layer ($0 \leq l \leq L-1$), L = total number of layers, N_l = number of neuron units at the lth level and i is the neuron number ($1 \leq i \leq N_l$).

This is different to the standard three layer perceptron model [RM86] which is

$$u_i = \sum_{j=1}^{N_l} w_{ij} a_j \tag{4.4}$$

$$a_i = \tanh(u_i) \tag{4.5}$$

By placing the nonlinearity inside the summation, the tanh characteristic of

a Gilbert multiplier can be used to implement this function. As shown in Section 5.7, this does not affect the neural network's ability to solve highly nonlinear problems such as XOR and parity problems.

4.5.2 Limited Precision Training

The training of a neural network such as Bourke is a difficult problem, the main difficulty lying in the fact that Bourke has both limited precision weights and a transfer function which is not precisely known.

The effect of the limited precision weights is that the output is restricted in terms of both range and resolution and so only problems which can be mapped within the restricted synapse range can be solved. The finite precision also makes it difficult for training algorithms to compute finite difference approximations to derivatives since the accuracy of this depends on the resolution with which the weight can be updated. In addition, the transfer function of the Bourke chip is nonideal due to leakage, mismatch *etc.*

In order to simulate the neural network design, a hierarchy of increasingly sophisticated models was used to represent the circuit. These models in increasing order of sophistication (and decreasing order of speed) were

1. A mathematical model based on Equation 4.1.
2. A mathematical model with quantised weights.
3. A full analogue simulation from the extracted layout using Apex.

The resulting weights from training the simpler models were used as starting values for the next model. This technique allowed the network to be mostly trained on the simple model and then optimised again using the more sophisticated but slower model to obtain the next result. Full analogue simulation takes approximately 15 minutes for each feedforward cycle on a SUN Microsystems Sparcstation 1+, and so the number of these evaluations must be made as small as possible, otherwise training would take several weeks of computer time.

The nonideal characteristics which occur due to device mismatch and noise mean that the exact transfer function of the Bourke chip is not known. The widely used backpropagation technique could not be used for this reason since gradients cannot be computed. Techniques which approximate the derivative of the chip using finite difference methods were not used as they increase the cost of an iteration by a factor equal to the number of weights in the network. Statistical methods such as simulated annealing are even slower and hence not suitable. The techniques chosen were the Nelder–Mead [NM65] and Hooke–Jeeves [HJ61] methods of nonlinear optimisation which do not require computation of the derivatives.

The optimisation task to be performed is to minimise

$$f(w) = \sum_{p} (\mathrm{ff}(w, p) - o(p))^2 \tag{4.6}$$

where p are the input patterns, ff is the neural network feedforward function, w are the weights and $o(p)$ is the desired output for pattern p.

The Nelder–Mead method was first used to optimise Equation 4.6, ff being described by Equation 4.3. The weights obtained were used as a starting condition for the next optimisation step which minimised the dynamic range of the weights. The Hooke–Jeeves method of optimisation was used for this task as it was found to be more successful. In order to minimise the weight range, the optimisation problem was reformulated to be

$$f(w) = 10,000 \times errors(w, p) + \frac{max(|w|)}{min(|w|)} \tag{4.7}$$

where $errors$ is the number of errors in classification, $max(|w|)$ is the maximum value of the absolute value of all the weights and $min(|w|)$ is the minimum value of the absolute value of all the weights. Minimising this equation thus minimises the number of bits if it can be achieved without sacrificing the classification result. Note that the number of errors is computed by using a margin value m which is selected by the user. An error is said to occur if $|\mathrm{ff}(w, p) - o(p)| > m$.

The optimised quantised weights were then used as the initial values for an Apex simulation of the extracted chip which minimises Equation 4.6 using the Hooke–Jeeves method. To compute the ff function for this, software was written to automatically construct an Apex simulation file, run Apex and to extract the node voltages representing the neuron activation from the Apex output file.

4.5.3 Hierarchical Training Example: XOR

An Apex simulation of the Bourke chip was tested by training an XOR network. Two inputs plus a bias input were used on the (3,3,1) network of Bourke.

A bias current of 5 nA was selected for the training experiment. This value was attained by simulating many networks using a mathematical model of the neural network, this value achieving reliable results for most simulations.

After training with the mathematical model and floating point weights, the following results were achieved after 2360 function evaluations [1] where "I" denotes the input, "O" the desired output, "N" the neural network output, `RANGE` $= max(|w|)/min(|w|)$, `ERR` is the mean squared error and all the numbers represent voltages (in volts) at the neuron outputs.

```
I 0.000000 0.000000 0.200000 ;
O -0.100000 ; N -0.099998 ;

I 0.000000 0.200000 0.200000 ;
O 0.100000 ; N 0.099978 ;

I 0.200000 0.000000 0.200000 ;
O 0.100000 ; N 0.100008 ;

I 0.200000 0.200000 0.200000 ;
```

[1] A function evaluation is considered to be an application of the ff function – in order to compute the error function for XOR, 4 feedforward evaluations are required.

```
O -0.100000 ; N -0.100063 ;

ERR= 1.13e-09 RANGE= 445
```

These weights were used as the starting point for the optimisation of Equation 4.7 which aims to reduce the dynamic range required by the synapses. After 1632 function evaluations the following results (input vectors omitted) were obtained

```
O -0.100000 ; N -0.050289 ;
O 0.100000 ; N 0.086844 ;
O 0.100000 ; N 0.058402 ;
O -0.100000 ; N -0.090538 ;

ERR= 0.001116 RANGE= 62
```

The result of the two preceding optimisations is a set of weights which can be used as a starting point for the optimisation of the full analogue simulation. The quality of the starting point can be seen by using the analogue simulation with the previously computed weights

```
O -0.100000 ; N -0.065000 ;
O 0.100000 ;  N 0.097000 ;
O 0.100000 ;  N 0.075000 ;
O -0.100000 ; N -0.097000 ;

ERR= 0.000467 RANGE= 31
```

The optimisation using a full analogue simulation was then performed and after 414 evaluations the result was

```
O -0.100000 ; N -0.123
O 0.100000 ; N 0.108
```

```
O 0.100000 ; N 0.102
O -0.100000 ; N -0.102

ERR= 0.0001502 RANGE= 31
```

This result shows that the analogue simulation of the chip can indeed be trained to perform the XOR mapping and the ability to train a full analogue simulation of the Bourke chip before it was sent to be fabricated gave confidence in the correctness of the chip design. Note that this model does not deal with device mismatch.

4.5.4 Bourke Chip Characterisation

Prior to any training tests on the Bourke chip, basic chip functionality and characterisation tests were performed. The characterisation tests consisted of measuring the synapse output as the bias, voltage input and weight inputs were varied. Results of these tests are not included in this chapter since in the next chapter, the same tests as applied to the Kakadu chip will be described.

4.5.5 Training of Bourke Chip

The Bourke chip was trained under control of a custom test jig interfaced to an IBM PC 386SX clone. A PC–LabCard "Multi–Lab Card" was used to provide 12 bit D/A, A/D and digital IO control of the Bourke chip. A bias current of 5 nA was used.

Attempts to train the Bourke chip on XOR from random weights using Hooke–Jeeves and Nelder–Mead optimisation were unsuccessful although many attempts at many different biases were tried. Training was only achieved after using the weights from the analogue simulation of the previous section as the starting weight values for the chip training. The outputs after training (740 evaluations) were

```
O -0.100000 ; N -0.153809 ;
O 0.100000 ; N 0.134277 ;
O 0.100000 ; N 0.119629 ;
O -0.100000 ; N -0.090332 ;

ERR= 0.001137 RANGE= 31
```

This result indicated that the Bourke chip had been correctly trained to solve the XOR problem.

4.6 Summary

This chapter began with a summary of previous implementations of artificial neural networks in VLSI. Following this, the design concerns for a neural network chip suitable for the low power classification of ICEG signals was described. The development and simulation of the Bourke neural network prototype was also detailed in this chapter.

Bourke implemented a small (3,3,1) neural network which employs multiplying digital to analogue converters for the synapses and off–chip resistors for the neurons. Using a hierarchical training technique, an analogue simulation of the chip was successfully trained on XOR before fabrication. The Bourke chip was used to test the concept of the architecture and was successfully trained on the XOR problem.

In the next chapter, the Kakadu chip which used the same cells and architecture as Bourke to implement a (10,6,4) network is described. Results of performance tests on its synapses and its ability to perform classification tasks are detailed.

Chapter 5

Kakadu Prototype

5.1 Introduction

Following the successful testing of Bourke on the XOR problem, a more ambitious chip, called Kakadu was designed. Kakadu has 10 input neurons, 6 hidden neurons and 4 output neurons and used the same synapses and neurons as Bourke. Kakadu was designed to be a general purpose chip, applicable to many problems.

In this chapter, the design of the Kakadu chip is detailed and the Kakadu chip performance is assessed by analyzing the measured synapse transfer function, MDAC linearity, power consumption and performance on some benchmark classification problems.

The problem of training a chip such as Kakadu is by no means trivial and the two questions that were encountered was how to control the chip and what algorithm to use to train the chip? In order to control signal levels on the Kakadu chip, a general purpose chip tester called Jiggle was designed. In order to select a training algorithm suitable for the Kakadu chip, a comparison between the combined search algorithm (CSA) [XJ92], Hooke–Jeeves optimisation [HJ61] and Nelder–Mead optimisation [NM65] was made. This comparison, as well as a description of the Jiggle chip tester, are also described in this chapter.

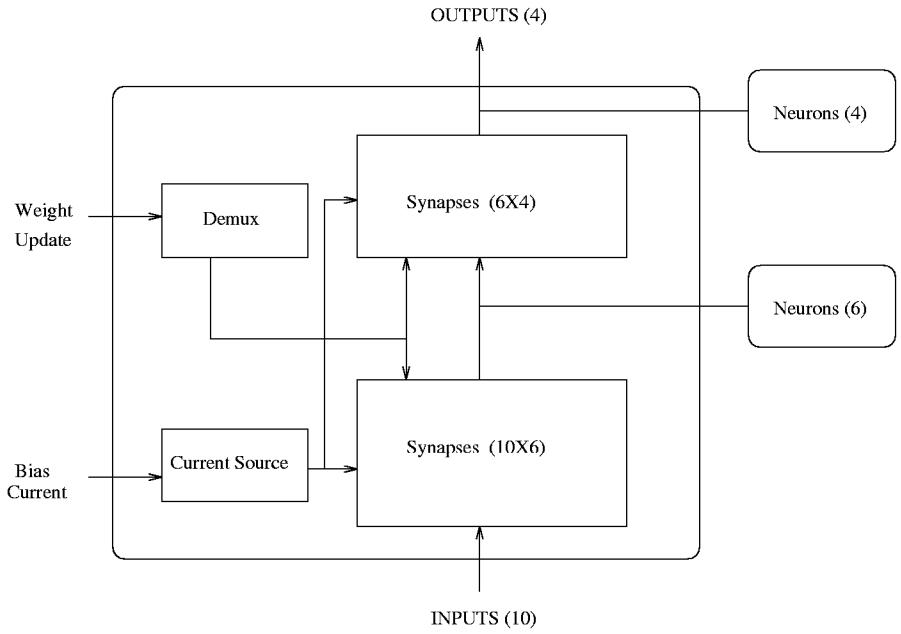

Figure 5.1: Block diagram of the Kakadu chip.

5.2 Kakadu Chip

It was determined that the largest network with the same architecture as Bourke which would fit in a 40 pin package was (10,6,4) in size. In order to achieve this, many of the digital signals were multiplexed and all synapse addressing and writing performed in a bit serial fashion. A block diagram of the Kakadu chip is in Figure 5.1.

A floorplan of the chip is shown in Figure 5.2, a plot in Figure 5.3 and a photomicrograph showing the main synapse blocks, row shift registers and the current source is shown in Figure 5.4. As can be seen from the plot and photomicrograph, the synapse arrays dominate the chip area.

Although Kakadu is a (10,6,4) neural network, it can implement any smaller network by setting unused synapses to zero. Setting a synapse to zero means

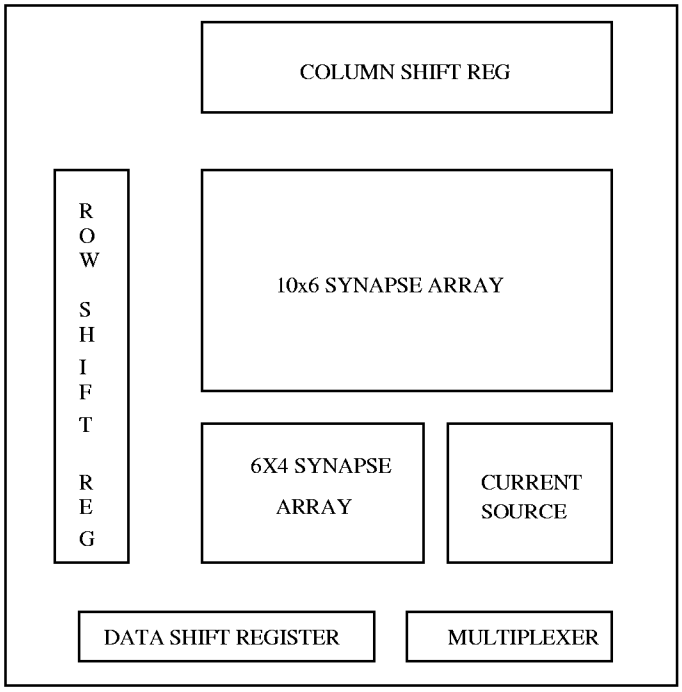

Figure 5.2: Floorplan of the Kakadu chip.

that no current can flow through that synapse and so no power is consumed by such synapses.

A summary of the major chip features of Kakadu is given in Table 5.1. Although the synapse circuit has differential inputs, in order to save pins, all 10 inputs to the chip are single ended and referenced to a single pin.

Unlike Bourke, Kakadu could not be simulated from the extracted layout since the design was too large for the Apex analogue simulator (5950 transistors). The digital weight writing circuitry was tested through digital simulation using Irsim and a netlist comparison, using the "Wombat" program from the University of California at Berkeley, was made to compare the extracted netlist with the schematic diagram.

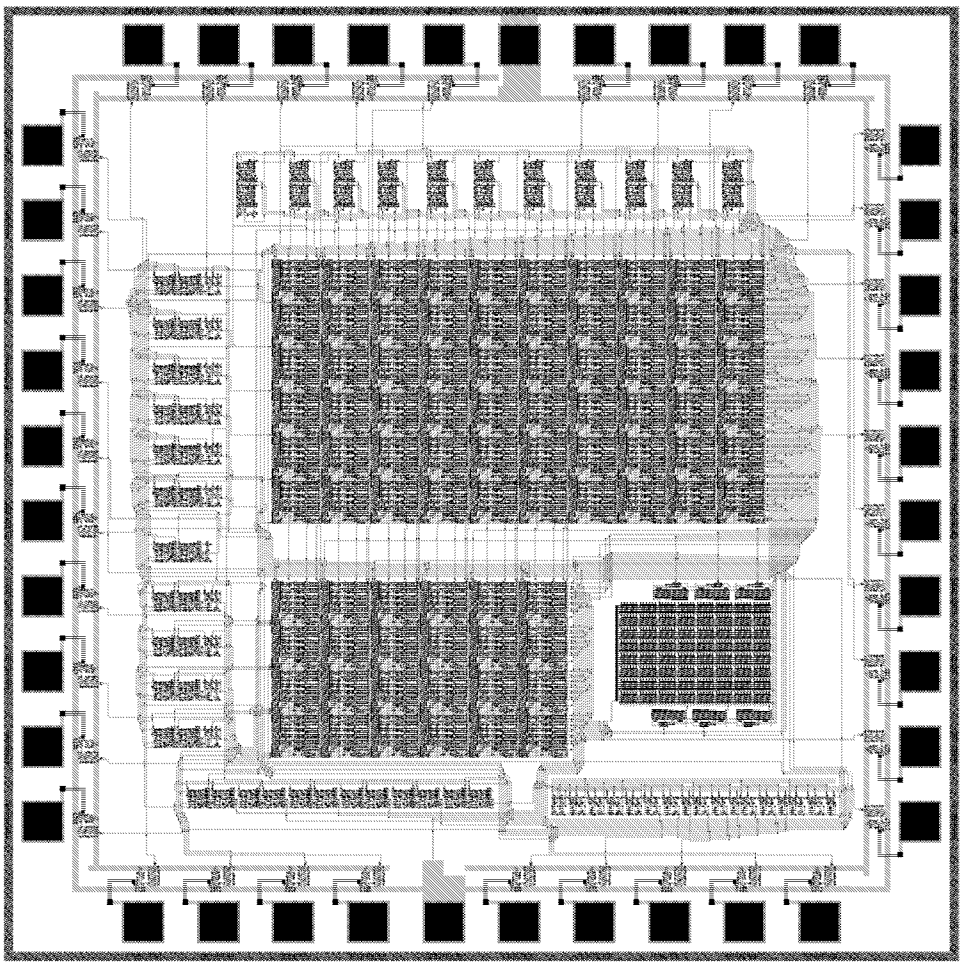

Figure 5.3: Plot of Kakadu chip.

Technology	1.2 μm double metal, single poly nwell
Chip Size	2.2×2.2 mm
Transistors	5950
Synapses	84×6 bit MDAC
Power Supply	3 V
Power Consumption (typical)	20 μW
Power Consumption with zero bias	not measurable (< 100 pA)
Propagation Delay	30 μS (0.2 V swing into 2 pF)

Table 5.1: Kakadu chip summary.

Figure 5.4: Photomicrograph of the Kakadu chip.

5.3 Jiggle Test System

5.3.1 Introduction

Jiggle is a general purpose chip tester which was designed specifically for the training and testing of low power analogue integrated circuits [LV92]. Commercial analogue chip testers are too expensive to justify for small projects, and multifunction analogue/digital interface cards, such as the PC–LabCard "Multi–Lab Card" used to test Bourke in Section 4.5.5, have too few channels to test the Kakadu chip. Jiggle has the following features

- 40 digital IO channels
- 40 analogue IO channels
- 2×12 bit analogue to digital converters
- 40×12 bit digital to analogue converters

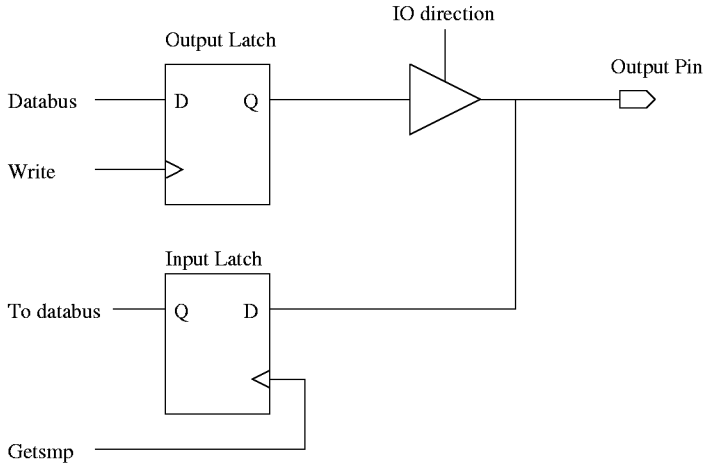

Figure 5.5: Jiggle digital IO block diagram.

- sample and hold circuitry on each analogue pin so that all can be sampled simultaneously
- software configuration of analogue and digital IO direction
- VME bus interface
- UNIX library for jiggle IO

Digital IO in Jiggle is achieved in the obvious manner, and is illustrated in Figure 5.5. The data direction is configured on the tristate buffer, and the output latch can be set to the desired binary level by writing to the output register. When configured as an input, a "Getsmp" command will sample all analogue and digital channels simultaneously and the state of all of the Jiggle inputs can then be read back at a later time.

Analogue IO is illustrated in Figure 5.6 and is similar in principle to the digital IO. A relay is used to configure the pin as either an input or an output. If it is an output, the switch is closed and the output of the 12 bit DAC drives the pin. Regardless of the IO direction, pins also drive a sample and hold amplifier which has an input impedance of 50MΩ. This high input impedance is necessary when

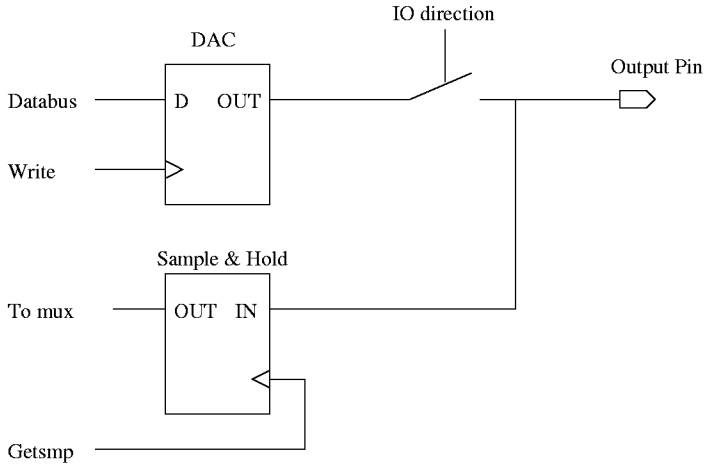

Figure 5.6: Jiggle analogue IO block diagram.

measuring output devices with low drive capability so that it does not affect the circuit being tested. The output of each pin's sample and hold is multiplexed so that two 12 bit analogue to digital converters can be shared over the 40 analogue channels.

5.3.2 Kakadu Test Setup

A block diagram of the Kakadu based test system is shown in Figure 5.7. The Jiggle chip tester has a VMEbus interface, and a SBUS to VMEbus interface card is used to allow a Sun 4 Sparcstation IPC to control the Jiggle card which in turn is connected to the Kakadu chip.

The Jiggle chip tester allows all pins of the Kakadu chip to be controlled from software using either digital or analogue input/output pins. The Kakadu chip is housed on a separate daughter board and contains some level shifting circuitry to convert the 5 V signals of Jiggle to the 3 V signals required by Kakadu. A photograph of this setup is shown in Figure 5.8.

All of the experiments conducted on Kakadu were controlled using Jiggle. The Jiggle chip tester allowed the Kakadu chip to be configured with bias voltages,

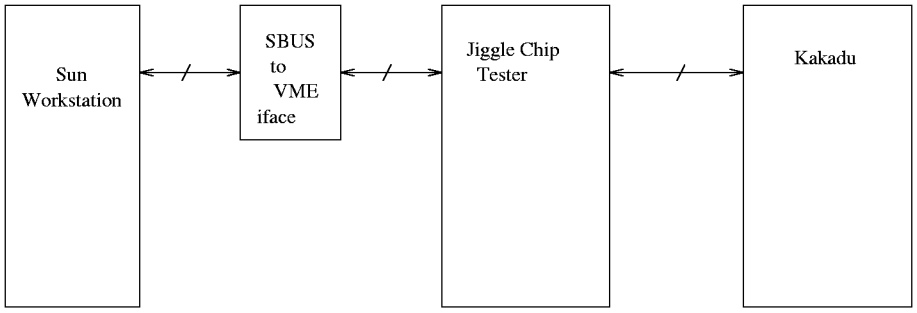

Figure 5.7: Block diagram of the Jiggle based test system.

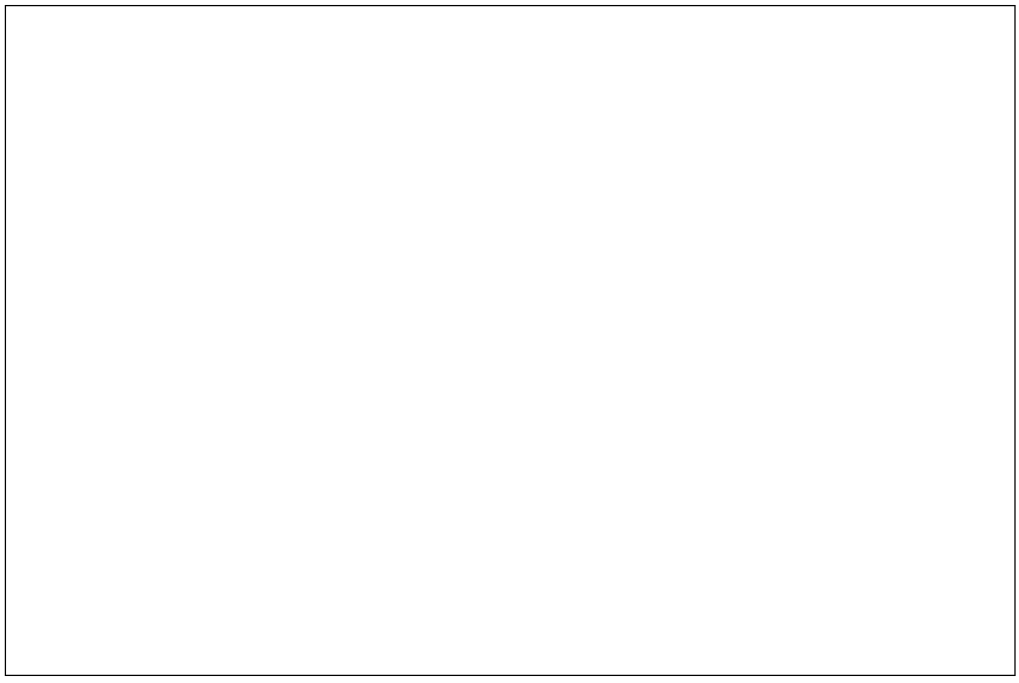

Figure 5.8: Photograph of the Kakadu test system.

weights and input via software control. Outputs were measured either using the 12 bit analogue to digital converters (ADCs) in Jiggle or on conventional test equipment.

5.4 Synapse Transfer Function

5.4.1 Voltage Inputs

From Equation 4.1 and Ohm's law, the synapse followed by a neuron has a transfer function described by

$$V_{out} = \begin{cases} +RI_{DAC}\tanh(\frac{\kappa(V_3-V_4)}{2}) & \text{if B5} = 1 \\ -RI_{DAC}\tanh(\frac{\kappa(V_3-V_4)}{2}) & \text{if B5} = 0 \end{cases} \tag{5.1}$$

where $R = 1.2 \times 10^6$. The synapse transfer function can be obtained by keeping I_{DAC} constant and varying V_3 and V_4. A curve fit was used to find $\kappa = 26.0719$ in this equation, and a plot of the measured and expected synapse transfer function can be seen in Figure 5.9. There is close agreement between the expected and measured values.

5.4.2 MDAC Linearity Test

The Gilbert multiplier used in the Kakadu MDAC has a transfer function described by Equation 4.1. The output of the DAC is equal to $I_{out+} - I_{out-}$ and so for fixed input voltages this equation can be simplified to be

$$I_{out} = kI_{DAC} \tag{5.2}$$

where k is a constant and

$$I_{DAC} = \begin{cases} +\sum_{k=0}^{4} 2^k B_i k & \text{if B5} = 1 \\ -\sum_{k=0}^{4} 2^k B_i k & \text{if B5} = 0 \end{cases} \tag{5.3}$$

Plots of the (measured) MDAC linearity for various bias currents are shown in Figures 5.10 and 5.11.

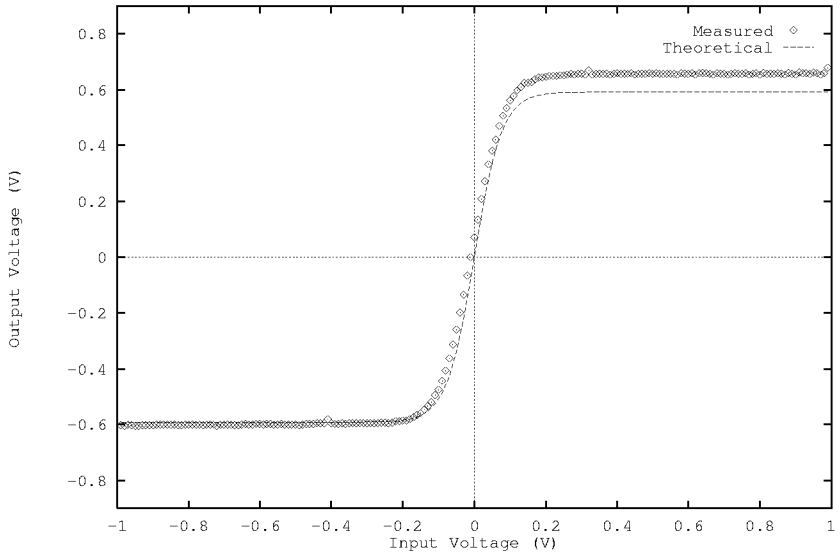

Figure 5.9: Synapse transfer function (measured and expected).

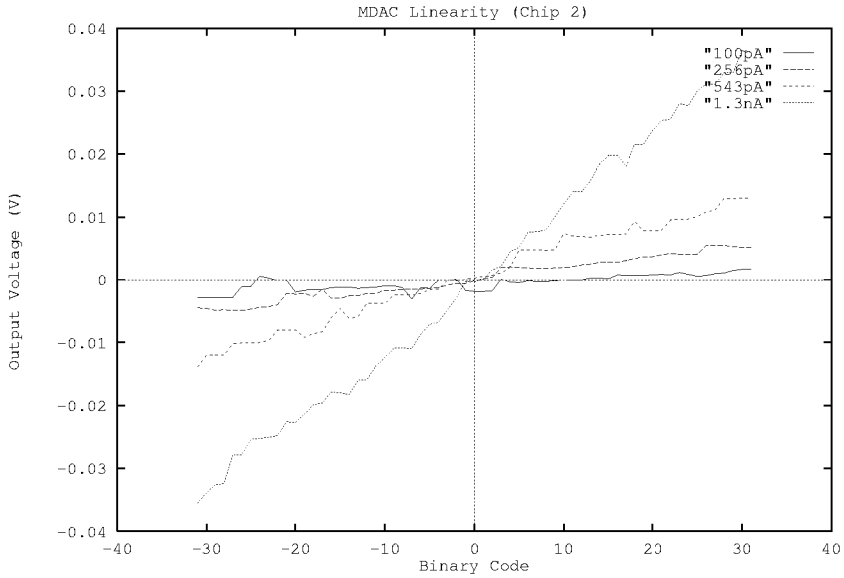

Figure 5.10: Measured MDAC linearity (low bias currents).

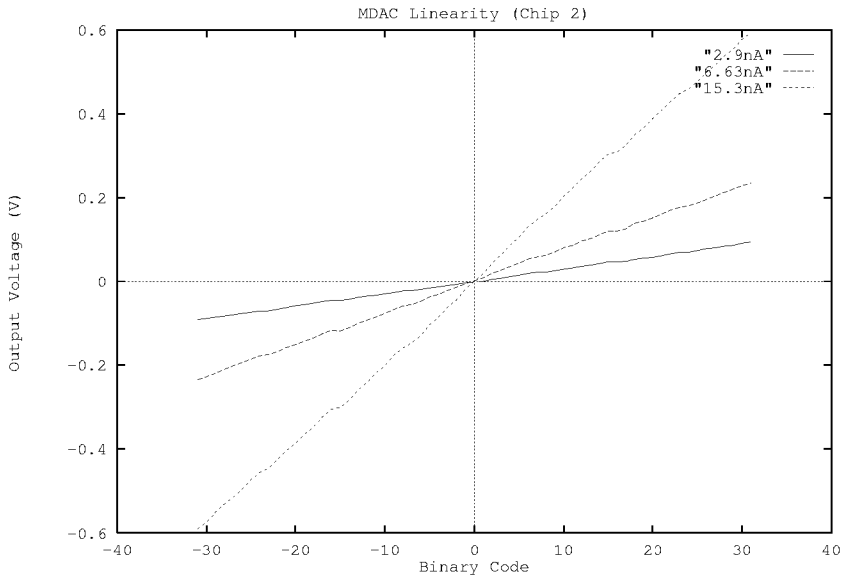

Figure 5.11: Measured MDAC linearity (high bias currents).

As the bias current increases, the DAC linearity can be seen to improve and a tradeoff between these can be made. Although monotonicity was not achieved until the bias current reached 15.3 nA, a bias current of 6.63 nA was selected for the operation of the experiments. This choice of bias current had a single point where it was not monotonic but it was expected that such behaviour would not affect the operation of a neural network since training can compensate for such effects.

5.5 Propagation Delay

The propagation delay of the Kakadu chip is determined by the time taken to charge up the neuron capacitance. This is highly dependent on the bias current, synapse values and the capacitance of the output.

In order to measure a typical propagation delay figure for Kakadu, the chip was biased at 6.63 nA and the time from a change in the inputs until the output

reaches 90% of the final value (2pF load) was measured. This delay was typically 30 μS.

5.6 Power Consumption

It is useful to be able to estimate the power consumption of the Kakadu chip. This is a function which is linear with the weight values since I_{DAC} in Figure 4.11 is the current drawn for that particular synapse. The current consumption I_{KAKADU} can be approximated by the formula

$$I_{KAKADU} = a + b\sum_{i=0}^{N}|w_i| \quad (\mu A) \tag{5.4}$$

where a and b are constants, w_i is the ith weight, i indexes through all of the weights in the chip and I_{KAKADU} is the current consumption in μA.

The power consumption for a number of different synapse values was measured after the outputs had settled to the final output value. This is the static consumption and includes the chip plus the off–chip neuron dissipation. A Philips/Fluke PM2525 multimeter with a resolution of 100 pA was used for these measurements.

The chip was biased at 6.63 nA and a least squares fit was made to the data, the result being $a = 0.842, b = 0.00736$. Figure 5.12 shows the measured current dissipation of the chip and the curve fit of Equation 5.4 to this data.

It can be seen that the linear fit quite closely approximates all of the points and this confirms that this is a valid technique for estimating the power consumption of the Kakadu chip. If the weights are not known, an upper bound for a particular architecture size can be determined (if operated at the same bias) using the formula

$$I_{MAXKAKADU} = 0.842 + n * 31 * 0.00736 \quad (\mu A) \tag{5.5}$$

where n is the total number of synapses in the neural network.

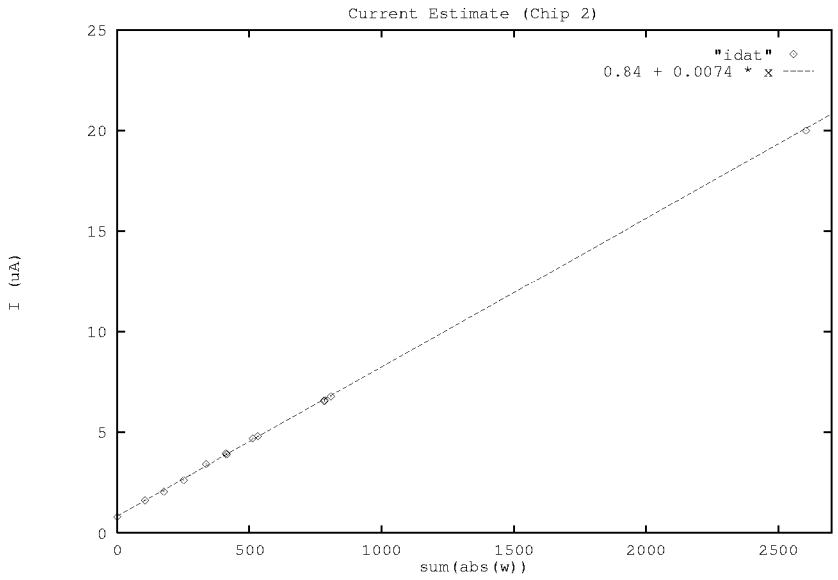

Figure 5.12: Current consumption curve fit (bias current = 6.63 nA).

An additional experiment was conducted with the weights all set to the maximum value, and the bias current reduced to zero. The current consumption of the chip in this configuration could not be measured with the PM2525, the reading being less than 100 pA. In this configuration, essentially no current, other than leakage, is used by Kakadu.

5.7 Neural Network Training

Training of analogue neural network chips is much harder than their digital counterparts. Mismatch of transistors, imperfect transistor models and noise means that a mathematical formula for the neural network transfer function cannot be attained and thus a formula for the gradient also cannot be reliably computed.

Such effects were not a major problem in the Bourke chip since the number of synapses was small and a very good set of starting weights could be obtained

through analogue simulation. However, the Bourke chip could not be trained successfully from random starting weights using either Nelder–Mead or Hooke–Jeeves optimisation techniques. For a chip such as Kakadu, where analogue simulation is not possible due to its size, a more reliable optimisation algorithm is required.

Several techniques were used to train the Kakadu chip from random starting weights by minimising Equation 4.6 described in Section 4.5.2. Nelder–Mead [NM65], Hooke–Jeeves [HJ61], and the combined search algorithm (CSA) [XJ92] were all used successfully in the training of Kakadu on different problems. An experiment was conducted in an attempt to find the most reliable training algorithm of the three.

In the following sections, the CSA algorithm will firstly be described, and then the comparison experiment between algorithms will be discussed.

5.7.1 Combined Search Algorithm

The combined search algorithm [XJ92] employs two minimisation strategies, namely modified weight perturbation and random search. Modified weight perturbation is a local search and the random search algorithm is a non–local search technique. CSA can be described by the following pseudocode

```
while not converged
{
    /* modified weight perturbation */
    for i = 1 to 10
    {
        for each weight w
        {
            wsave = w;
            w = w + DELTA;     /* DELTA is usually set to 1 */
```

```
            evaluate error;
            if error has not improved
                w = wsave;
        }
    }
    /* random search algorithm */
    for i = 1 to 30
    {
        for each weight w
        {
            wsave = w;
            w = uniformly distributed random number;
            evaluate error;
            if error has not improved
                w = wsave;
        }
    }
}
```

The CSA algorithm is very simple and the results obtained are surprisingly good, convergence being very fast for small problems. Although CSA has been successfully used to train Kakadu on problems with more than 50 synapses, it is expected that performance would degrade rapidly for larger problems.

5.7.2 Comparison between Training Algorithms

The algorithms were tested by attempting to train a quantised mathematical model of Kakadu derived from Equation 4.3 on four different problems. These problems were xor (Section 5.8.1), par3 (Section 5.8.2), par4 (Section 5.8.3) and morph (Section 5.8.5).

Problem	*Nelder–Mead*		
	Number Trained	Iterations (mean)	Iterations (SD)
xor	1	8348	–
par3	3	28253	9560
par4	0	–	–
morph	4	23554	9066
Problem	*Hooke–Jeeves*		
xor	1	1044	–
par3	0	–	–
par4	0	–	–
morph	3	18698	8289
Problem	*CSA*		
xor	1	14584	–
par3	8	9781	13959
par4	0	–	–
morph	8	24644	1658

Table 5.2: Training experiment results (8 trials per problem).

The training started from random weights, and was considered successful if the outputs of the neural network could be trained to within a margin value of the desired values. This test was conducted 8 times for each algorithm and training set, and the results are summarised in Table 5.2.

From the table, it is easily seen that CSA is the only algorithm of the three that can reliably train the Kakadu architecture, and it was able to train all problems except for par4 and xor in all of the trials in the experiment. The method which was finally used to train the par4 problem is discussed in Section 5.8.3.

5.8 Neural Network Training Examples

The Kakadu chip was trained on a number of benchmark neural network classification problems using the combined search algorithm and the Jiggle chip tester. Training time for these problems was usually between 5 and 50 minutes and the main bottleneck was Kakadu's serial weight update interface. All of the experiments were conducted using a 3 V supply and the performance and power consumption results are presented in the following sections.

Input (Volts)			*Desired Output*	*Output (Volts)*
0.2	0.0	0.0	0.0	0.031
0.2	0.2	0.0	0.2	0.215
0.2	0.0	0.2	0.2	0.173
0.2	0.2	0.2	0.0	0.032

Table 5.3: Results of applying Kakadu to the XOR problem (6.9 μW).

5.8.1 XOR

XOR has been a benchmark problem for neural networks because it is a simple yet highly nonlinear problem. The minimum network size which can solve this problem is (3,2,1) with one input being a bias. To make Kakadu behave like a smaller network, the weight values for the unconnected synapses are set to zero. Kakadu was successfully trained on this problem, results of this test being shown in Table 5.3. The bias current used was 6.63 nA. The chip was considered successfully trained if the difference between the measured and desired output less than a particular margin. For the XOR function, this margin was set to be 0.08 V.

For XOR, the power dissipated was 6.9 μW at 3 V. The same problem has been successfully trained with bias currents down to 3.5 nA. For bias currents lower than this, training could not be achieved. It was decided that further tests would be conducted with a bias of 6.63 nA to be sure that sufficient DAC linearity was achieved.

5.8.2 PARITY (3 BIT)

Three bit parity is another nonlinear benchmark problem which can be thought of as XOR in three dimensions. The function and results of the problem are tabulated in Table 5.4. Kakadu was successfully trained using a (4,3,1) network with a margin set to 0.08 V and a bias current of 6.63 nA. The quiescent power consumption for this problem was 9.0 μW.

Input (Volts)				*Desired Output*	*Output (Volts)*
0.2	-0.1	-0.1	-0.1	-0.1	-0.103
0.2	-0.1	-0.1	+0.1	+0.1	+0.952
0.2	-0.1	+0.1	-0.1	+0.1	+0.103
0.2	-0.1	+0.1	+0.1	-0.1	-0.098
0.2	+0.1	-0.1	-0.1	+0.1	+0.103
0.2	+0.1	-0.1	+0.1	-0.1	-0.073
0.2	+0.1	+0.1	-0.1	-0.1	-0.090
0.2	+0.1	+0.1	+0.1	+0.1	+0.105

Table 5.4: Results of applying Kakadu to the parity 3 problem (9.0 μW).

5.8.3 PARITY (4 BIT)

The 4 bit parity problem was very difficult to train on a (5,4,1) network and although many attempts were made at training the chip using CSA, none were successful.

Training was finally achieved by using CSA on a mathematical model of the chip to provide a starting point for further training using the chip. The mathematical model was one which used Equation 4.3, and the weights were allowed to be floating point values within the maximum range of the synapse values (*i.e.* $-31 \leq w \leq 31$). This modified weight range had the following benefits

- enabled training to be performed in the absence of quantisation effects
- makes Equation 4.6 continuous in this range
- allows CSA to be used with a DELTA less than 1

When the chip was trained with CSA using the starting values thus obtained, it converged very quickly to a solution for this problem.

The results of the training on the chip are shown in Table 5.5. The margin was set to 0.08 V and the bias current was 6.63 nA. The quiescent power consumption for this problem was 15.6 μW.

Input (Volts)					*Desired Output*	*Output (Volts)*
0.2	-0.1	-0.1	-0.1	-0.1	-0.1	-0.071
0.2	-0.1	-0.1	-0.1	+0.1	+0.1	+0.071
0.2	-0.1	-0.1	+0.1	-0.1	+0.1	+0.066
0.2	-0.1	-0.1	+0.1	+0.1	-0.1	-0.085
0.2	-0.1	+0.1	-0.1	-0.1	+0.1	+0.061
0.2	-0.1	+0.1	-0.1	+0.1	-0.1	-0.073
0.2	-0.1	+0.1	+0.1	-0.1	-0.1	-0.059
0.2	-0.1	+0.1	+0.1	+0.1	+0.1	+0.071
0.2	+0.1	-0.1	-0.1	-0.1	+0.1	+0.141
0.2	+0.1	-0.1	-0.1	+0.1	-0.1	-0.117
0.2	+0.1	-0.1	+0.1	-0.1	-0.1	-0.103
0.2	+0.1	-0.1	+0.1	+0.1	+0.1	+0.051
0.2	+0.1	+0.1	-0.1	-0.1	-0.1	-0.056
0.2	+0.1	+0.1	-0.1	+0.1	+0.1	+0.022
0.2	+0.1	+0.1	+0.1	-0.1	+0.1	+0.085
0.2	+0.1	+0.1	+0.1	+0.1	-0.1	-0.024

Table 5.5: Results of applying Kakadu to the parity 4 problem (15.6 μW).

5.8.4 Character Recognition

Figure 5.13 shows a simple character recognition problem to which Kakadu was applied. A (10,6,4) network was divided into a bias unit and a 3×3 pixel array. The network was trained (bias current 4.4 nA) on the characters '0', '1', '7' and '+', each output being assigned to one character. Only four characters were used because there are only four outputs on the Kakadu chip. After training, one or more bits in each character was corrupted and the network output passed through a "winner take all" decision to determine the network's classification of the corrupted character. The results of this experiment (shown in Tables 5.6 and 5.13) show that Kakadu was able to correctly classify patterns that it had not been trained on. Kakadu draws 22.5 μW during this test.

5.8.5 ICEG Morphology Classification

The Kakadu chip was designed primarily to classify ICEG waveforms based on morphology to aid in the identification of cardiac arrhythmias. This final example

Input			*Output*				*Class*
1 1 1	1 1	1 1 1	-0.12	-0.11	-0.09	+0.10	0
1 1 1	1 1 1	1 1 1	-0.10	-0.04	-0.16	+0.09	0
	1 1 1		-0.10	-0.10	+0.10	-0.10	1
1	1 1 1		-0.05	-0.09	+0.00	-0.13	1
1 1	1 1	1	-0.10	+0.09	-0.10	-0.10	7
1 1	1 1 1	1	-0.16	+0.03	-0.10	-0.02	7
1	1 1 1	1	+0.10	-0.10	-0.09	-0.09	+
1	1 1		+0.06	-0.08	-0.02	-0.15	+

Table 5.6: Results of applying Kakadu to the character recognition problem (22.5 μW). In the "Input" column a '1' denotes an input voltage of 0.1 V and a blank is -0.1 V. An additional bias input of 0.2 V was used.

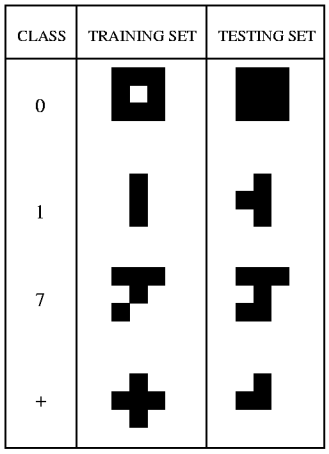

Figure 5.13: Character recognition example.

demonstrates Kakadu's ability to identify abnormal heart signals based on the shape of the signal.

A VT 1:1 patient was selected from an ICEG database and Kakadu was trained on 4 NSR and 4 VT 1:1 complexes (bias current 6.63 nA). The patterns were constructed by taking 10 samples centered about each QRS. The test set comprised 68 VT and 447 NSR rhythms from the same patient. On this problem, 100% correct classification was achieved. The average power consumption for this problem was 20 μW.

5.9 Summary

The Kakadu chip implements a (10,6,4) neural network using the same neuron and synapse design as Bourke. In this chapter, the design and testing of the Kakadu chip was detailed. A description of the Jiggle chip tester was presented as well as a comparison between the Hooke–Jeeves, Nelder–Mead and combined

search algorithm training algorithms. Of the three training algorithms, the combined search algorithm was found to be the most reliable.

The measured and expected synapse transfer functions of the Kakadu test chip were compared and found to be in close agreement. Linearity of the DAC improved with increasing bias current. The DAC was found to be 6 bit monotonic at a bias current of 15.3 nA although the chip has solved the XOR problem with a bias as low as 3.5 nA. A bias current of 6.63 nA was selected as an operating point for neural network training experiments to ensure good DAC linearity without overly increasing the power dissipation of the chip.

The Kakadu chip was trained using the combined search algorithm on XOR, 3 bit parity, 4 bit parity, ICEG morphology classification and character recognition problems. For the 4 bit parity problem, the combined search algorithm could not find a solution from a random starting point. However, when a starting point was obtained from a mathematical model of the chip, a solution was quickly found.

Also developed in this chapter was a linear model of the current consumption of the Kakadu chip was a function of the synapse values. This model was found to agree very well with the data and enables accurate estimates of the chip's power consumption to be made. The maximum measured power dissipation for all of the problems described in this chapter was 20 μW.

Chapter 6

System Performance

6.1 Introduction

In this chapter, the performance of MATIC is evaluated using a large database of ICEG signals taken from 67 patients. In order to provide a basis for comparison, a simple threshold detector (STD) similar to the classification algorithm used in the Telectronics Guardian 4210 is first applied to the database and the classification performance noted.

The STD classifier is then compared with MATIC using a three layer perceptron network with floating point weights (FPNN). This system shows the performance of the MATIC algorithm with a "perfect" network which does not suffer from problems of noise, matching and limited precision weights. Misclassified complexes are analysed and the causes of these misclassifications are justified. The importance of the morphology classifier and the ability of a neural network to store multiple morphologies are also demonstrated.

A MATIC system which employs the Kakadu chip is then described and the classification performance on the ICEG database is presented. These results are compared with those obtained with the FPNN. The chapter concludes with measured power consumption figures of the Kakadu chip obtained while performing morphology classification.

6.2 ICEG Database

MATIC was tested on a database of intracardiac electrogram (ICEG) signals containing 12483 QRS complexes recorded from 67 patients during electrophysiological studies. The patients had a wide variety of arrhythmias including sinus tachycardia (ST), normal sinus rhythm, normal sinus rhythm with bundle branch block, sinus tachycardia with bundle branch block, atrial fibrillation (AF), various supraventricular tachycardias, ventricular tachycardia, ventricular tachycardia with 1:1 retrograde conduction (VT 1:1) and ventricular fibrillation (see Table 6.1). Within these arrhythmias, there were also numerous ventricular ectopic beats, fusion beats, noise and other artifacts.

The tachycardias were initially classified manually into subclasses by a human and the data was labelled with this information. The computer classification was not available to the human during this process. Only regions of stable rhythms were labelled, regions with unstable rhythms due to the programmed stimulation were not used in the study. Multiple rhythms (usually NSR and a tachycardia) were labelled in most patients and a total of 122 classified "regions" were obtained in this manner from the 67 patients used in this study. The labelled data were then confirmed by other human experts to double check the classifications.

Since the MATIC algorithm produces classifications which correspond to ICD therapy, it only produces 4 different "superclasses", namely NSR, SVT, VT and VF (see Section 3.2.1.4). The mapping used to convert from a subclass to a superclass is shown in the first two columns of Table 6.1.

6.3 Simple Threshold Detector

In order to compare the performance of MATIC with a classifier similar to one currently in use, a classification system based on the algorithm employed in the Telectronics Guardian ATP 4210 [Ltd89] was used (see Section 2.5 for a description of the Guardian device). This is a simple threshold detector (STD)

with the RR interval (measured in the RVA channel) being compared to fixed thresholds. The "sudden onset" detector used in the Guardian 4210 was not employed as all tachycardias in the database were initiated through programmed stimulation. The default 4210 classification parameters were used –

$$CLASS = \begin{cases} VF & \text{if } RR < 250 \text{ ms} \\ VT & \text{if } 250 \text{ ms} \leq RR < 400 \text{ ms} \\ NSR & \text{otherwise} \end{cases}$$

Output of this threshold detector was then fed to an "8 out of 10" post processor (as done in the 4210).

The results of the STD classifier are shown in the "confusion matrix" of Table 6.1. The STD system output classes appear in the columns of the matrix and human classifications appear in the rows. Correct classifications appear on the diagonal of the matrix and are in bold type. The subclass classification made by the human is mapped to an appropriate superclass (as shown in the first two columns of the table) so that the performance of STD for various tachycardias can be determined. As an example, the second row of Table 6.1 shows that the database held 10 regions of subclass "ST" (superclass "NSR"). STD produced 1882 outputs for these (ST) regions, and of these, 1786 were NSR and 96 were VT.

It can be clearly seen that the classifier does not produce reliable classification. An overall classification performance of 75.9% was achieved. If the SVT patients in the database are omitted (current ICDs are not used in patients susceptible to SVTs), the revised correct classification rate of the STD classifier becomes 86.6%.

As expected, sinus tachycardia and supraventricular tachycardias produce VT false positives, and the lack of atrial information in the STD logic means that certain slower VT rhythms are classified as NSR.

Note that the threshold parameters used in a real STD classifier are normally set in a patient dependent manner and one would expect a real ICD to give much

Subclass	Superclass	Regions	NSR	SVT	VT	VF
NSR	NSR	64	**5032**	0	4	0
ST	NSR	10	**1786**	0	96	0
SVT	SVT	9	191	**0**	860	0
AT	SVT	2	51	**0**	8	0
AF	SVT	3	123	**0**	1	0
VT	VT	9	134	0	**226**	0
VT 1:1	VT	10	737	0	**351**	110
VF	VF	6	6	0	0	**157**
VTF	VF	9	6	0	79	**31**

Table 6.1: Subclass confusion matrix of STD classifier. Note that a classification rate of 75.9% is achieved for all data and a rate of 86.6% for non–SVT data. The rows represent the human classified subclass and the columns represent the computer classification. Human subclass classifications are collapsed into superclasses in the rows according to the "Superclass" column. The "Regions" column is the total number of stable rhythms for that subclass. Correct classifications are in bold type.

better performance than that of Table 6.1. However, since only sinus rhythm and a tachycardia were available for most patients in our database, setting a patient dependent threshold would not give a true indication of the classifier's ability (since there would only be two different rhythms).

6.4 MATIC with FPNN

In order to provide a means whereby the hardware neural network could be compared with a "perfect" neural network, the MATIC algorithm was first tested using a three layer perceptron neural network with floating point weights. This floating point neural network (FPNN) has the following differences to Kakadu

- Uses 64 bit IEEE floating point weights instead of the 6 bit values of Kakadu.
- Implements a three layer perceptron (Equation 4.5) instead of the linear neuron with nonlinear synapse model (Equation 4.3).
- Does not suffer from noise or drift.

Subclass	Superclass	Regions	NSR	SVT	VT	VF
NSR	NSR	64	**5605**	4	2	0
ST	NSR	10	**1535**	24	2	1
SVT	SVT	9	0	**1022**	0	0
AT	SVT	2	0	**52**	0	0
AF	SVT	3	0	**165**	0	0
VT	VT	9	0	0	**332**	0
VT 1:1	VT	10	2	0	**1253**	0
VF	VF	6	0	0	2	**196**
VTF	VF	9	0	2	0	**116**

Table 6.2: Subclass confusion matrix of MATIC classifier. Note that 99.6% of classifications are correct.

For the classification experiments using the FPNN, the parameters of the neural network morphology classifier (as described in Section 3.3.4) were $\alpha = 0.0$, $\beta = 1.0$ and $\gamma = 0.8$.

6.4.1 FPNN Results

The MATIC algorithm with FPNN morphology classifier was used to classify the ICEG database and the results are shown in Table 6.2; 99.6% correct classification was achieved.

It can easily be seen that MATIC performs better than the STD classifier for all arrhythmia subclasses. This success is mostly due to the extra information which can be obtained from the additional atrial lead.

6.4.1.1 Misclassifications

In total, 10315 outputs were produced by MATIC from the 12483 input QRS complexes (an output is generated only if the X out of Y criterion is satisfied so the number of outputs is always less than number of QRS complexes). This also causes different classifiers to produce a different number of outputs even though the same database is used.

A total of 39 QRS complexes were incorrectly classified and these were caused by the following

- Edge Effect (4 complexes). Upon onset of an arrhythmia, latency effects in the X out of Y detector can cause the first complex to be incorrectly classified if the last X complexes were the same. This is not a serious problem as in practice it does not matter if correct classification is delayed by a single QRS complex (see Figure 6.1 for an example).

- Bad QRS Detection (11 complexes). The QRS detector used was very good, however there were some cases where it made false detections. Spikes in the waveform could cause false detections or missed detections, causing MATIC to produce an incorrect classification.

- MATIC Errors (24 complexes). These misclassifications are due to inadequacies in the MATIC system. *All of these 24 complexes came from a single patient* with sinus tachycardia and were classified as SVT. In this patient, a 240 ms AV conduction time can be observed (see Figure 6.2), and the patient has 1° AV block. MATIC will misclassify this condition as SVT since ($2PR > RR$).

6.4.2 Performance of MATIC without Morphology

In order to assess the usefulness of the morphology classifier, the MATIC algorithm with morphology disabled was applied to the VT 1:1 patients. The confusion matrix of this experiment is shown in Table 6.3. Of the 2358 classifications made, 1045 were incorrect (55.7% correct classification). MATIC (with morphology disabled) did not correctly classify any VT 1:1 rhythms since the timing logic requires AV dissociation for a VT classification (see Section 3.3.3).

6.4.3 Multiple Morphologies

An additional experiment was conducted to demonstrate the ability of the neural network classifier to store multiple morphologies. The 8 training samples from

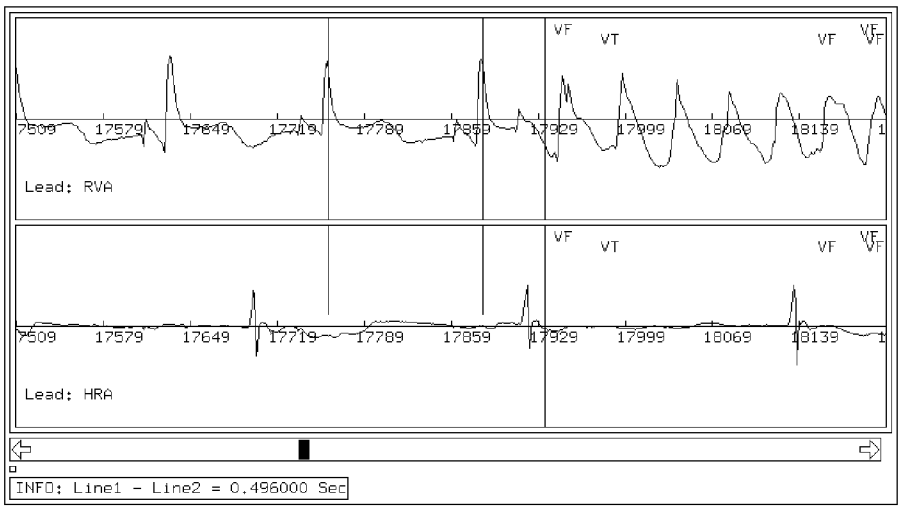

Figure 6.1: ICEG misclassified because of the edge effect. The VF is classified as a VT since the 5 previous QRS complexes (out of 6) were VT, causing the X out of Y detector to assume that it is VT.

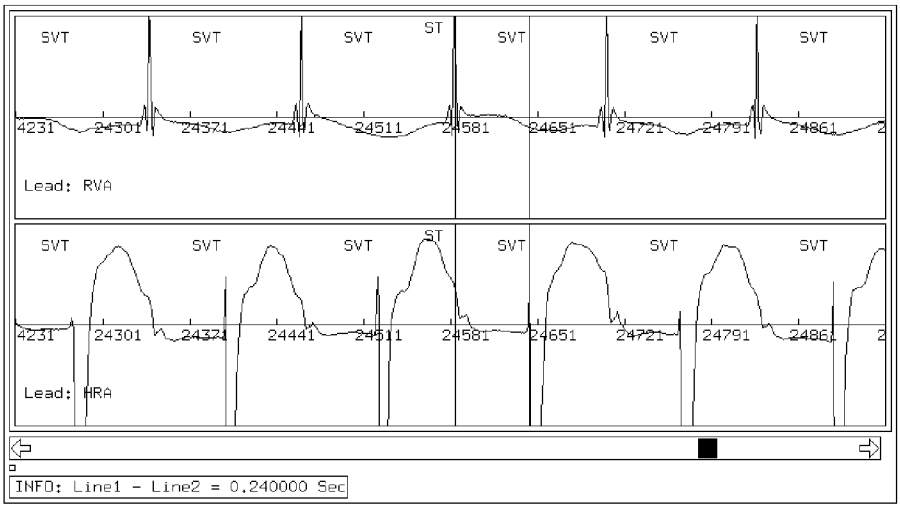

Figure 6.2: Misclassification due to 1° heart block (PR = 244 ms). MATIC misclassifies this ST as SVT because ($2PR > RR$).

Subclass	Superclass	Regions	NSR	SVT	VT	VF
NSR	NSR	10	**1313**	0	0	0
VT 1:1	VT	10	127	894	**0**	24

Table 6.3: Subclass confusion matrix of MATIC classifier for VT 1:1 data with morphology detection disabled. 55.7% are correctly classified. Note that all VT 1:1 rhythms are incorrectly classified.

Subclass	Superclass	Regions	NSR	SVT	VT	VF
NSR	NSR	10	**1252**	0	1	0
VT 1:1	VT	10	0	25	**982**	0

Table 6.4: Subclass confusion matrix of MATIC classifier for VT 1:1 data using a single set of weights, demonstrating the network's ability to store multiple patterns. 98.8% are correctly classified.

each of the 10 patients with VT 1:1 were used to form a training set of 80 QRS complexes. A larger network consisting of 10 input units, 8 hidden units and 1 output unit was found to produce the best results. After training, the resulting network was used to classify all of the VT 1:1 patients. Thus the network was required to store 20 different morphologies (NSR and VT 1:1 for each of the 10 patients). The results, shown in Table 6.4, show a degraded performance compared with a separate classifier for each patient with 26 misclassified complexes. However, 24 of the 25 VT 1:1 rhythms misclassified as SVT were from a single patient whose NSR morphology is very similar to another patient's VT 1:1 morphology. From this result, we observe that multiple patterns can be stored by the neural network, but if NSR and VT 1:1 patterns in the training set are very similar, incorrect results may occur. It should be noted that the same problem will exist for any template matching algorithm and even for humans.

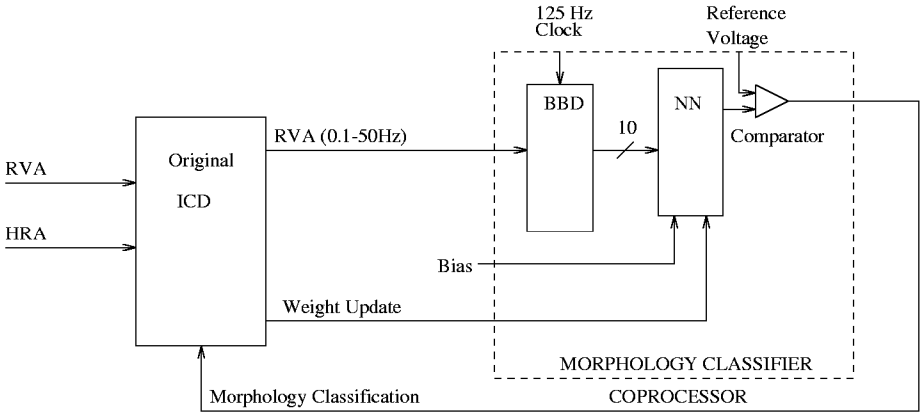

Figure 6.3: Block diagram of MATIC classifier system in an ICD.

6.5 MATIC Implemented using Kakadu

A system architecture for implementing the MATIC algorithm using a neural network coprocessor was presented in Section 3.4.2. In this section, the implementation of this architecture using the Kakadu chip as the coprocessor is described.

6.5.1 System Architecture

A slightly more detailed block diagram of Figure 3.11 which employs a bucket brigade device and a neural network chip is shown in Figure 6.3.

In this architecture, the microprocessor inside an ICD is used to implement all of the timing algorithm of MATIC except for the morphology classifier. The morphology classifier is implemented in a neural network chip operating in parallel with the ICD. The morphology classifier obtains a 0.1–50 Hz band limited RVA input from the ICD and passes this through an analogue delay line to obtain

a window of previous samples. These form the input to the neural network and thus a classifier the same as Figure 3.10 is implemented and morphologies can be analyzed by the neural network. The output of the neural network is passed through a comparator producing a digital output which can be read by the ICD.

6.5.1.1 Analogue Delay Line

A charge transfer device can be used to produce a sampled analogue delay line, charge transfer techniques being both power efficient as well as small in size. There are two different devices which are suitable for this purpose, the charge coupled device (CCD) and the bucket brigade device (BBD).

A CCD is an array of closely spaced MOS diodes [Sze81]. By applying the appropriate clocking sequence to a CCD, packets of charge can be stored and moved across a semiconductor substrate. The CCD offers the advantages of lower noise and smaller area over a BBD. However, CCDs require a special fabrication process, generally require a more complicated clocking sequence, and are more difficult to implement. To avoid the difficulties of implementing and fabricating a CCD, and considering that only an extremely short analogue delay line was required (10 stages), a BBD was chosen.

The design and test of a bucket brigade device which was fabricated on the Bourke chip is described in Appendix B. At the 125 Hz operating frequency, the prototype BBD circuit was found to have a charge transfer inefficiency of 0.35% and an estimated power consumption of 40 nW (see Section B.5.1). The addition of a BBD would not change the power consumption of a chip such as Kakadu since it consumes many orders of magnitude less power than that of the synapses.

In order to ensure that the neural network output is read at the correct time, the ICD device must read the output of the neural network when the QRS is complex is centered within the window. In the case of 10 inputs, the ICD device

should read the output of the comparator 5 samples after the occurrence of an R wave.

6.5.1.2 Test System

A test system using Kakadu and Jiggle to serve as the morphology classifier for MATIC was developed to evaluate the performance of the system described in the previous section.

When compared with the morphology classifier block shown in Figure 6.3, Kakadu implements the NN block of the morphology classifier and the Jiggle chip tester replaces the other circuitry in the morphology classifier as well as the ICD.

The Jiggle based system is completely software controllable and thus facilitates experimentation. In particular, replacing the BBD with 10 DACs from Jiggle removes the real–time constraint of the system and so data can be processed in a batch environment. This simplifies debugging of the system and makes experimentation easier.

Since the linear range of the synapse in Kakadu is less than 200 mV (Figure 5.9), 0.35% of this value is 0.7 mV — less than the resolution of the Jiggle test jig (which is 1.2 mV). Thus the CTI of this BBD would have an insignificant effect on the operation of the neural network morphology classifier in MATIC, especially considering the good generalisation properties of a neural network.

6.5.2 Kakadu Results

The MATIC algorithm with Kakadu biased at 6.63 nA was applied to the ICEG database. The results are tabulated in Table 6.5; 99.3% correct classification was achieved. The results of classifying only the VT 1:1 patients (the neural network is not used for other patients) are shown in Table 6.6.

In order to achieve the results just presented, the morphology parameters α, β and γ (Section 3.3.4) were selected by trial and error in a patient dependent

Subclass	Superclass	NSR	SVT	VT	VF
NSR	NSR	**5406**	4	20	0
ST	NSR	**1535**	24	2	1
SVT	SVT	0	**1022**	0	0
AT	SVT	0	**52**	0	0
AF	SVT	0	**165**	0	0
VT	VT	0	0	**332**	0
VT 1:1	VT	0	16	**974**	0
VF	VF	0	0	2	**196**
VTF	VF	0	2	0	**116**

Table 6.5: Subclass confusion matrix of MATIC classifier using the Kakadu chip. Note that 99.3% of classifications are correct.

Sublass	Superclass	NSR	SVT	VT	VF
NSR	NSR	**1114**	0	18	0
VT 1:1	VT	0	16	**974**	1

Table 6.6: Subclass confusion matrix of MATIC classifier using Kakadu. 98.4% are correctly classified.

fashion. The actual values used are shown in Table 6.7. Patient dependent morphology parameters were not necessary for the FPNN but were required for Kakadu mainly to compensate for the low resolution of the synapse values (6 bits). For the FPNN, synapses can take on floating point values and so arbitrary output levels can be achieved.

Patient	α	β	γ
1	0.0	1.0	0.8
2	0.0	1.0	0.8
3	0.0	0.5	0.4
4	0.0	0.2	0.1
5	0.0	0.5	0.4
6	-0.1	0.1	0.0
7	0.0	0.5	0.4
8	0.0	0.5	0.4
9	0.0	1.0	0.8
10	0.0	1.0	0.8

Table 6.7: Morphology parameters of Kakadu chip for the 10 VT 1:1 patients.

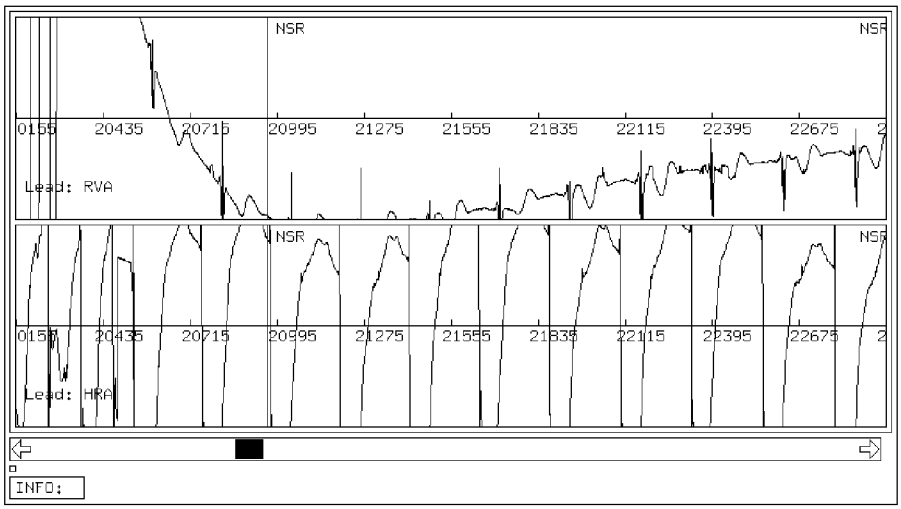

Figure 6.4: Patient misclassified because of large baseline drift.

6.5.2.1 Misclassified Morphologies

It is interesting to note that nearly all of the morphologies misclassified by Kakadu came from two patients. These patients were classified correctly by the FPNN.

In the first patient (Figure 6.4) a low frequency drift of the baseline signal occurred. This was caused by the amplifiers of the recording system being overloaded following stimulation therapy. In the FPNN, this does not present problems since it can tolerate a large input range. In Kakadu, however, inputs which are outside the linear range of the synapses (see synapse transfer function in Figure 5.9) are compressed and clipped. This causes all inputs above and below the linear range to appear the same to the neural network. A small baseline drift can be tolerated, but for this particular patient, the drift was too large for Kakadu. This is not a potentially dangerous situation since such large drifts were recorded only because of the data collection system recovering from the effects of applying stimulation. For this reason ICDs ignore the first few classifications following shock therapy. High pass filtering can also minimise this effect.

In the second patient (Figure 6.5), the morphology change between NSR and

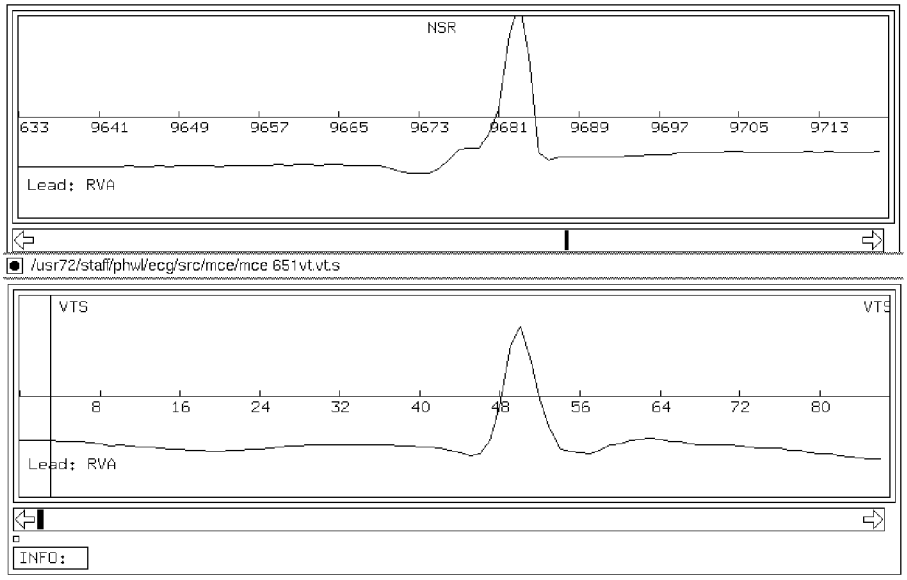

Figure 6.5: Patient misclassified because of similar NSR/VT 1:1 morphology. The top ICEG is a QRS from the RVA channel of the patient's NSR and the bottom is one from the same patient's VT 1:1.

VT 1:1 was not great enough for Kakadu to detect, although the FPNN could detect the change. This is again caused by the saturating input characteristic of the Kakadu network. The morphology change seen between NSR and VT 1:1 is a higher R wave. The Kakadu network cannot distinguish between the two morphologies since it compresses the input and so they appear very similar. It would not be recommended that morphology criterion alone be used to classify patients with such a small change between their NSR and VT 1:1 morphologies, even if a FPNN (which achieves good results) was employed since minor morphology changes such as caused by bundle branch block or drug therapy, could easily cause misclassifications.

Patient	Power
1	18.3 μW
2	16.5 μW
3	12.6 μW
4	13.8 μW
5	20.1 μW
6	21.6 μW
7	9.0 μW
8	15.6 μW
9	13.5 μW
10	24.0 μW

Table 6.8: Power consumption of Kakadu chip for the 10 VT 1:1 patients.

6.5.2.2 Kakadu Power Consumption

The power consumed by Kakadu when classifying the 10 VT 1:1 patients is shown in Table 6.8. All patients required less than 25 μW.

Since the Kakadu chip draws practically no current ($<$ 100 pA) when the bias transistor is grounded (see Table 5.1), a further reduction in the power consumption of Kakadu whilst being used for MATIC can be achieved. The propagation delay of the chip is approximately 30 μS (see Table 5.1), if a very conservative value of 1000 μS is allowed for the outputs to settle and a heart rate of 1 Hz is assumed, the Kakadu chip need only be biased 1/1000 th of the time, reducing the average current consumption of the system from 25 μW to a maximum of 25 nW.

6.6 Summary

MATIC classifies patients based on timing and morphological features. For 57 of the 67 patients, morphology analysis was not required because timing based classification was conclusive. For these cases, patient independent classification was achieved. Ten patients had VT with 1:1 retrograde conduction which could not be classified on the basis of timing alone, so patient dependent morphological considerations were employed to obtain reliable classification. A total correct

classification rate of 99.6% was achieved by MATIC using a floating point neural network, compared to the 75.9% achieved by a simple threshold detector. MATIC is simple enough to realise on a low power integrated circuit yet achieved near perfect classification results on a large database of ICEG signals.

MATIC implemented using Kakadu was applied to the same database of signals, suffering a slight degradation in performance down to 99.3%. Two patients that could be classified using a FPNN morphology classifier could not be classified using Kakadu. In each case, this was due to the compression of the input signal by the first synapse layer. The Kakadu neural network chip could discriminate between normal and abnormal morphologies whilst consuming a maximum of 25 microwatts continuous power. This figure could be further reduced to less than 25 nW by turning off the bias of the chip when it is not used.

Chapter 7

Conclusion

The aim of this thesis has been to develop a system for classifying tachyarrhythmias which is suitable for use in an implantable cardioverter defibrillator (ICD). The main difficulty encountered was that an ICD imposes a very restrictive power budget and so computationally expensive classification algorithms could not be implemented.

The first step taken towards addressing the problem was the development of the MATIC algorithm. This algorithm and an architecture for its implementation was described in Chapter 3. MATIC is a tachyarrhythmia classifier which follows a classification process similar to that used by cardiologists. Classifications are performed by combining timing and morphological criteria to produce a system which can make very reliable classifications on a wide range of arrhythmias. The computationally expensive part of the MATIC algorithm is the neural network morphology classifier and the implementation of this using the microprocessor of an ICD was shown to be prohibitively expensive. To overcome this problem, an architecture which employs a low power neural network coprocessor chip was developed.

The design and implementation of two prototype low power analogue neural network chips was undertaken in order to produce a morphology classifier suitable for use with the MATIC algorithm. The first chip, called Bourke, was

designed to test the building block circuits and is described in Chapter 4. Bourke implemented a bucket brigade and a small (3,3,1) neural network and it was successfully tested on the XOR problem.

The second chip, called Kakadu, is described in Chapter 5 and was based on the same circuit elements as Bourke. Kakadu implemented a (10,6,4) feedforward neural network. The synaptic elements in Kakadu are multiplying digital to analogue converters (MDACs), and the nonlinearity of the multiplier in the MDAC is used to provide the nonlinearity of the neural network. Neurons are implemented as off chip resistors and have a linear transfer function. The low power consumption of the Kakadu chip is achieved by operating the MDACs at low bias currents, typically 6.63 nA. Kakadu was successfully tested on a number of classification problems and had a typical power dissipation of tens of microwatts.

In Chapter 6, the performance of the MATIC algorithm was tested on a large database of intracardiac electrograms. In order to provide a basis for comparison, a classifier similar to that used in an existing ICD, the Telectronics Guardian ATP 4210, was tested on this database and achieved 75.9% correct classification. MATIC achieved 99.6% correct classification with a floating point neural network implemented in software. When the Kakadu chip was used to replace the floating point neural network, the classification performance dropped to 99.3%. This small decrease was attributable to two patients that could not be classified reliably. In both cases, the problem was due the compression effect of the input synapses of Kakadu. This limitation would only affect the morphology classifier performance in very marginal cases, and in such cases, it would not be recommended that any morphology classifier be used. When performing classification, the Kakadu chip dissipated less than 25 microwatts of continuous power. This figure could be further reduced to less than 25 nW by turning off the bias of the Kakadu chip in the period between QRS complexes.

7.1 Future Work

7.1.1 MATIC

The MATIC algorithm is, of course, not the last word in arrhythmia classification. Future work will further improve on the present classifiers and it is hoped that ideas from this thesis may inspire advances in this area.

The construction of a test ICD which incorporates the Kakadu chip and MATIC algorithm could be made by modifying an existing ICD. Such a system could be used in animal trials to take the realisation of an ICD device with a neural network morphology classifier one step further. Of course, the final goal would be to produce a new ICD device which incorporates algorithms like MATIC to perform morphology classification that could be used in humans. The increased reliability of improved algorithms like MATIC would make ICD therapy available to a larger group of patients and have an improved mortality rate in recipients, thus moving towards reducing the incidence of sudden cardiac death mortality.

ICEG signals may not be the best sensors for recognising tachyarrhythmias. Biosensors which monitor right heart pressure, impedance, stroke volume, temperature, oxygen content, cardiac output and pH are being investigated [CL91]. Some are able to more easily and clearly identify the hemodynamically significant tachyarrhythmias. By incorporating additional sensors and algorithms to analyze them in an ICD, classification performance can only improve.

7.1.2 Low Power Neural Network Chips

A large part of this thesis was devoted to the design of a low power analogue neural network chip. The power consumption of the Kakadu chip is dominated by the synapse design. The synapse design, in turn was constrained by the problem of producing a nonvolatile analogue storage device in standard CMOS. This is

one of the major problems facing VLSI neural network design, and if a method can be found to achieve small, high resolution and low power analogue storage, much more powerful neural network chips could be developed.

As the popularity and application of artificial neural networks classifiers increases, the use of low power neural network chips like Kakadu will also increase. Neural networks have been successfully applied to many fields of research including telecommunications, speech recognition, character recognition, image processing *etc.*. However, neural network research is still in its infancy and there are still very few actual hardware devices that incorporate neural network technology. As research on neural network applications progresses, any battery or solar powered system requiring computationally expensive nonlinear mapping could incorporate low power neural network chips to reduce power consumption. Some possible applications include handwriting input devices, speech recognition devices, toys and of course, implantable medical devices.

7.2 Closing Remarks

The aim of this thesis was to solve a difficult classification problem given a very limited power budget. The approach taken was to combine two classifiers: one based on a decision tree which implemented rules which were well understood and a second which used a neural network to perform classification based on examples. The neural network was well suited to the morphology recognition problem and by virtue of its architecture of simple processing elements, mapped well to analogue VLSI. The approach of using decision trees plus neural networks seems a particularly simple and elegant method of producing low power classifiers and this approach should be suitable for many other applications besides tachyarrhythmia classification.

Appendix A

Artificial Neural Networks

A.1 Introduction

An artificial neural network (ANN) is an architecture for computing inspired by biological neural systems. ANNs consist of a large number of simple computational elements (neurons) operating in parallel. The neurons are connected through programmable synaptic connections (also called weights) which modify the strength of the connections between neurons.

The connection topology of the neurons and weights form the architecture of the neural network. The architecture used in this thesis is a three layer perceptron, meaning that the neurons are organised into three layers (input layer, hidden layer and output layer), with any particular neuron being connected to all neurons in the layer above (see Figure A.1). The network operates in a strictly feedforward manner with the signal propagating from input layer through the hidden layer to the output layer via the weights (represented by lines which connect between neurons).

A neuron has a single input and a single output. The input is the weighted sum of the outputs of all neurons in the previous layer and the transfer function is usually a squashing function such as the sigmoidal function $f(x) = 1.0/(1+e^{-x})$ or $f(x) = \tanh(x)$. Neurons in the input layer are slightly different to those in the

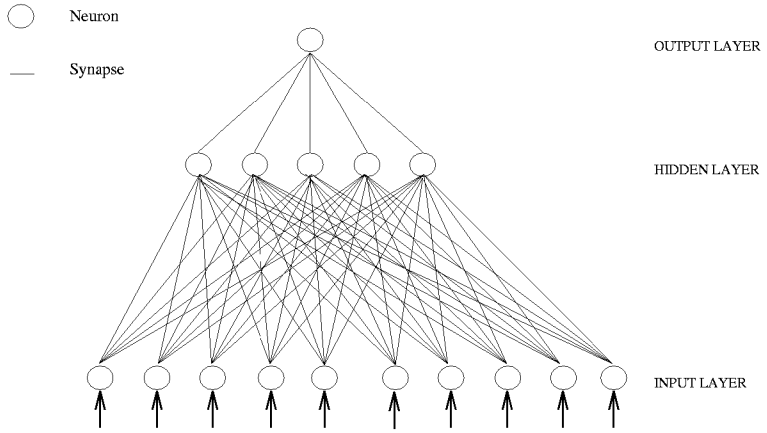

Figure A.1: Three layer perceptron architecture.

other layers as they have the identity transfer function $f(x) = x$. Mathematically, the output of a neuron in layer $l+1$ is given by

$$u_i(l+1) = \sum_{j=1}^{N_l} w_{ij} a_j(l) \quad \text{(A.1)}$$

$$a_i(l+1) = f(u_i(l+1)) \quad \text{(A.2)}$$

where l denotes the lth layer ($0 \leq l \leq L-1$), L = total number of layers, N_l = number of neuron units at the lth level, i is the neuron number ($1 \leq i \leq N_l$) and f is the activation function.

A three layer perceptron can, by virtue of the non–linearity of the neurons, form arbitrarily complex nonlinear mappings [Cyb89, HSW89]. Without the non–linearity of the neurons, only linear mappings can be achieved.

A.2 Training

In order to train an artificial neural network, a set of a number of input and output vector pairs is supplied by the user to provide examples for the training.

The input vectors are applied to the network, the network output is compared with the training set output and the weights of the neural network are updated to minimise this difference.

Training is commonly achieved using the backpropagation algorithm [Lip87]. Backpropagation simply performs minimisation of the multidimensional error function

$$e(w) = (\sum_{p=1}^{P}(d_p - a_p(w))^2)/P \tag{A.3}$$

where p iterates through the input patterns of the training set, P is the number of training patterns, w is a vector of weight values, d_p is the desired output, and a_p is the output of the neural network when the pth training pattern is applied to the inputs. Backpropagation uses the method of gradient descent, weights being updated by the formula

$$\Delta w_{ij} = -\eta \frac{\partial e(w)}{\partial w_{ij}} \tag{A.4}$$

where Δw_{ij} is the change to be applied to the weights ($w_{ij} \leftarrow w_{ij} + \Delta w_{ij}$), $\partial e/\partial w_{ij}$ is the partial derivative of e with respect to w_{ij}, and η is a constant.

By substituting Equations A.1, A.2 and A.3 in Equation A.4, and taking the partial derivatives, we obtain the backpropagation formulae

$$\Delta w_{ij} = \eta \delta_i(l+1) f'(u_i(l+1)) a_j(l) \tag{A.5}$$

$$\delta_i(l) = \begin{cases} \sum_{j=1}^{N_{l+1}} \delta_j(l+1) w_{ji} & \text{if } l \neq L \\ d_i(l) - a_i(l) & \text{if } l = L \end{cases} \tag{A.6}$$

where all the variables are as defined earlier.

For η sufficiently small the error function of the new w will always be less than or equal to the error function for the previous value of w and hence we can reduce the error until we reach a minima. This minima, in general, is not the global minimum, but for a wide class of problems the algorithm is able to find a satisfactory solution. The derivation of the backpropagation formula is detailed in Rumelhart and McClelland [RM86].

After training, when a supposedly optimal weight configuration is found, input patterns which were not in the training set are applied to the network. The neural network is said to *generalise* in these cases with the intent of producing sensible outputs for previously unseen inputs.

Appendix B

Bucket Brigade Devices

B.1 Introduction

A bucket brigade device is an analogue shift register which operates by transferring charges between capacitors which are connected together by MOS transistors. The signal is represented by a charge deficit and this is transferred between a series of storage capacitors by applying the appropriate switching sequence.

B.2 Simple Bucket Brigade

The simple BBD uses capacitors isolated via a MOSFET. The circuit shown in Figure B.1 is driven by the nonoverlapping clocks ϕ_1 and ϕ_2. This sequence causes deficits of charge to move from node S to node D.

B.2.1 Operation

Suppose that the signal is represented by the amount of charge Q and a deficit of charge equal to this value is stored at the source node S of Figure B.1. Assume also that the voltage at the destination node D is $V_{dd}-V_t$ where V_t is the threshold voltage. When the ϕ_2 clock is brought to V_{dd}, the voltage at D changes to $2V_{dd} - V_t$. An amount of charge equal to Q flows from D to S, changing the

voltage at S to $V_{dd} - V_t$ and transferring the charge deficit from S to D. Since $\Delta Q = C\Delta V$, the charge deficit can be measured as a voltage on nodes $V(S)$ and $V(D)$.

Similarly, during ϕ_1, a charge deficit is transferred from D to the next storage capacitor and so on. Thus by providing nonoverlapping clocks for ϕ_1 and ϕ_2, charge deficits representing the signal are shifted along the bucket brigade device.

B.2.2 Charge Transfer Inefficiency

The total charge transfer inefficiency (CTI) of a BBD is the sum of the intrinsic CTI and the drain conductance CTI. The intrinsic CTI is a high frequency effect caused by the fact that charging up the storage capacitors takes a finite amount of time, so if the BBD is clocked too fast, the CTI will increase.

The low frequency CTI of the simple BBD is caused by the output drain conductance of the transistors. As the charge transfer proceeds, the controlling transistor's region of operation moves from being in saturation to being subthreshold since the source and drain voltages tend to equalise. As the output conductance of the FET is finite, the subthreshold current is a function of the drain voltage (*i.e.* $i = i(v_D(t))$) causing a different current flow for different signals.

The expression for the low frequency CTI, α_{lf}, was derived by Berglund and Thornber [BT73] and can be expressed as

$$\alpha_{lf} \approx \frac{g_r C_S}{g_m C_D} \tag{B.1}$$

where g_m is the forward conductance of the transfer FET, g_r is the reverse conductance of the transfer FET, C_S is the source node capacitance and C_D is the drain node capacitance.

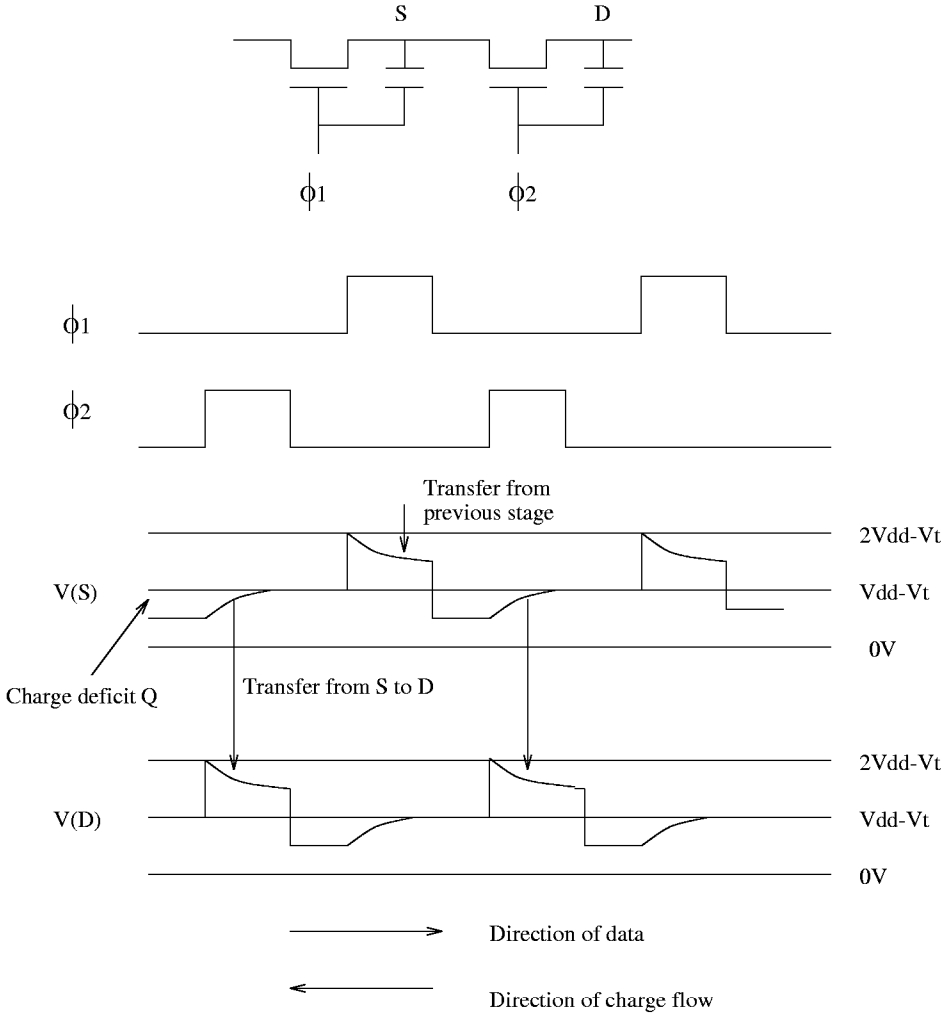

Figure B.1: Bucket brigade waveforms.

B.2.3 Frequency Response

The frequency response of a BBD circuit can be viewed in terms of incomplete charge transfer and is given by the following formula derived by Berglund [Ber71]

$$H(f) = [(\frac{1-2\alpha-2\beta}{1-2\alpha e^{-j2\pi f/f_s}})e^{-j2\pi f/f_s}]^n \tag{B.2}$$

where α is the low frequency CTI, β is the transfer attenuation per stage of the BBD, n is the length of the delay line and f_s is the sampling frequency.

B.3 Tetrode Bucket Brigade

The low frequency charge transfer inefficiency (CTI) of the simple BBD is limited by the output drain conductance of the BBD. The tetrode configuration shown in Figure B.2 reduces the effect of finite output conductance by using an extra transistor with gate connected to a voltage $V_{bb} = V_{dd} - 0.2$ V which makes the voltage swing at A and B smaller. This causes the low frequency CTI to be reduced to $(1 + \frac{C_S g_d}{C_B g_m})(\frac{C_B g_d}{C_S g_m})$ where $g_m = -\partial i/\partial v_S$ (forward conductance), $g_d = -\partial i/\partial v_D$ (reverse conductance), C_S is the storage capacitor capacitance, C_B is the parasitic capacitance between the two transistors, and C_g is the gate capacitance of the isolating transistor.

B.4 Test Circuit Design

When designing the bucket brigade circuit, it was not known whether the simple bucket brigade would have a sufficiently low CTI for the purposes of ICEG morphology classification. For this reason, a tetrode circuit was used, noting that the tetrode BBD can be made into a simple BBD by tying V_{BB} to V_{dd}. Table B.1 demonstrates how the two configurations can be achieved.

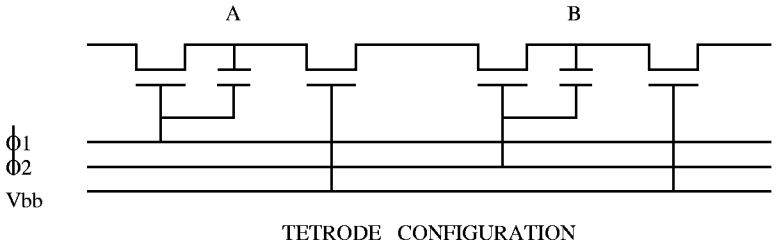

Figure B.2: Tetrode clocking scheme.

Device	V_{BB}	ϕ_1	ϕ_2
Simple BBD	V_{dd}	Clocked $0 - V_{dd}$	Clocked $0 - V_{dd}$
Tetrode BBD	$Vdd - 0.2\ V$	Clocked $0 - V_{dd}$	Clocked $0 - V_{dd}$

Table B.1: Clocking of BBDs on test chip.

B.4.1 Storage Capacitor Size

The size of the storage capacitor determines the signal to noise ratio of the BBD. As the data will be clocked into this BBD at a rate of 125 Hz, the noise at low clock frequencies is the only one of interest. In this case, the noise voltage is [Hen90]

$$E[v_n^2] = \frac{m}{2}\frac{kT}{C} \tag{B.3}$$

where m is a measured value of approximately 2.0.

In order to achieve the desired signal to noise ratio of 78 dB for a maximum input range of 0.5 V at body temperature (310 K), one can calculate from Equation B.3 that the capacitance must be 1 pF.

B.4.2 Transistor Size

The low frequency CTI of the BBD is related to the ratio g_d/g_m (Equation B.1) and Scott and Chamberlain [SC80] have shown that $g_d/g_m \propto 1/L$. Thus we can reduce the low frequency CTI by increasing the length of the transistors. However, increasing the length reduces the drive capability of the transistor and this in turn increases the intrinsic CTI. As a compromise between these two effects, a moderate length of 9.6 μm was chosen.

Secondly, the widths of the transfer and isolating transistors must be chosen. The widths do not have any bearing on the drain conductance CTI, however, Henderson [Hen90] shows that the intrinsic CTI is

$$\epsilon_i = (1 + \frac{1}{C_S}\frac{k_p}{2}\frac{Z_S Z_{BB}}{Z_S + Z_{BB}} t(V_{BB} - V_t - V_S(0)))^{-2} \tag{B.4}$$

where Z_S and Z_{BB} are the width to length ratios of the transfer and isolating transistors respectively and the optimal ratio is 1:1.

In deciding on the absolute value for the width, increasing the width will reduce the intrinsic CTI. However it will also increase the parasitics. The optimal absolute value of the width is that which reduces the intrinsic CTI to the same order as the drain conductance CTI. For these reasons, widths of 48.6 μm were selected.

B.5 Results

A 10 stage test BBD having the circuit of Figure B.2 was fabricated on the Bourke chip (see Figure 4.6). The size was chosen since that is the same size delay line required by MATIC.

The chip's CTI was calculated by measuring the impulse response of the BBD chip (in simple BBD mode), computing the frequency response, and curve fitting this measured frequency response to Equation B.2. Measurements were performed using a Hewlett Packard HP3561A Dynamic Signal Analyzer, and the

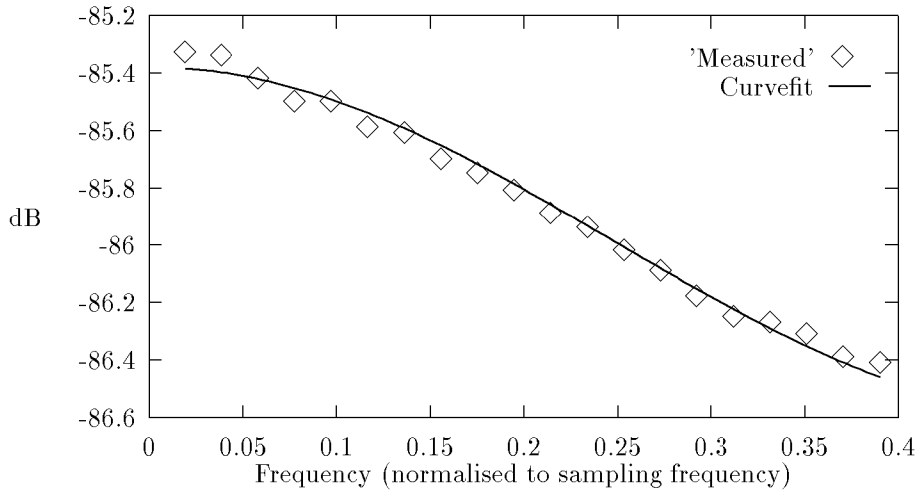

Figure B.3: BBD frequency response. Dotted line represents theoretical frequency response for $\alpha_{lf} = 0.35\%$.

circuit was operated from 3 V at 125 Hz. The curve fit is shown in Figure B.3 and indicates a low frequency CTI of 0.35%.

The Bourke chip was only tested as a simple BBD since this was found to have an adequately low CTI. Although a lower CTI is expected in the tetrode configuration, extra power would be required to generate the tetrode bias voltage.

B.5.1 Power Consumption of the BBD

The power drawn by the BBD was very small and the clocking frequency very low. Since power is consumed only during switching, an accurate power measurement could not be obtained.

The dominant term in the power consumption of the BBD is that used to charge the capacitance of the ϕ_1 and ϕ_2 nodes of Figure B.2. Since the circuit consists of ten stages, each with two 1 pF capacitors, the approximate capacitance which must be charged is 20 pF. If we allow another 20 pF for the capacitance of the routing, the power consumed charging this capacitance at a frequency of

125 Hz is then

$$P = CV^2 f \tag{B.5}$$

$$= 40 \times 10^{-12} \times 3^2 \times 125 \tag{B.6}$$

$$= 45 \text{ nW} \tag{B.7}$$

This value is many orders of magnitude less than the power used in Kakadu (which is tens of microwatts) and so the BBD device will not greatly affect the total power consumed by a neural network chip such as Kakadu.

B.6 Summary

Equations for the charge transfer inefficiency (CTI) and frequency response of the simple BBD were derived and used to design a BBD with a signal to noise ratio of 78 dB at body temperature. A 10 stage simple BBD was implemented and tested on the Bourke chip. This BBD was found to have a CTI of 0.35% from a 3 V supply at 125 Hz and power consumption of the BBD was estimated to be 45 nW.

Bibliography

[AB89] A.G. Andreou and K.A. Boahen. Synthetic neural circuits using current-domain signal representations. *Neural Computation*, pages 489–501, 1989.

[ABJ+84] R. Arzbaecher, T. Bump, J. Jenkins, K. Glick, F. Munkenbeck, J. Brown, and N. Nandhakumar. Automatic tachyarrhythmia recognition. *PACE*, pages 541–546, May–June 1984.

[And90] A.G. Andreou. Synthetic neural systems using current-mode circuits, May 1990.

[BDST91] C.A.A. Bastiaansen, Wouter J. Groeneveld D., H.J. Schouwenaars, and H.A.H. Termeer. A 10–b 40mhz 0.8μm CMOS current–output D/A converter. *IEEE Journal of Solid-State Circuits*, 26(7):917–921, July 1991.

[Ber71] C.N. Berglund. Analog performance limitations of charge–transfer dynamic shift registers. *IEEE Journal of Solid-State Circuits*, SC–6(6):391 394, December 1971.

[BGRL+92] U. Birgersdotter-Green, M. Rosenqvist, F.W. Lindermans, L. Ryden, and Kjell Radegran. Holter documented sudden death in a patient with an implanted defibrillator. *PACE*, pages 1008–1014, July 1992.

[BT73] C.N. Berglund and K.K. Thornber. Incomplete transfer in charge–transfer devices. *IEEE Journal of Solid-State Circuits*, SC–8:108–116, April 1973.

[Car89] L.R. Carley. Trimming analogue circuits using floating–gate analogue MOS memory. *IEEE Journal of Solid-State Circuits*, SC 25(6):1569–1575, December 1989.

[Chi90] A.M. Chiang. A CCD programmable signal processor. *IEEE Journal of Solid-State Circuits*, 25(6):1510–1517, December 1990.

[CJ90] Z. Chi and M. Jabri. Neural networks for QRS classification. In *Proceedings of 2nd IASTED International Symposium on Signal Processing and its Applications*, pages 858–861, Gold Coast, Australia, August 1990.

[CJ92] Z.R. Chi and M.A. Jabri. Identification of supraventricular and ventricular arrhythmias using a combination of three neural networks. In *Proceedings of Computers in Cardiology 1991*, pages 169–172, Venice, Italy, 1992. IEEE Computer Society Press.

[CL91] T.J. Cohen and B. Liem. Biosensor applications to antitachycardia devices. *PACE*, 14:322–327, February, Part II 1991.

[Cyb89] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2:303–314, 1989.

[dBC89] M.A. de Belder and A.J. Camm. Devices for tachycardia termination. *American Journal of Cardiology*, 64:70J–74J, December 1989.

[DBW90] R. Dzwonczyk, C.G. Brown, and H.A. Werman. The median frequency of the ECG during ventricular fibrillation: Its use in an algorithm for estimating the duration of cardiac arrest. *IEEE Transactions of Biomedical Engineering*, 37(6):640–646, June 1990.

[DS92] D.A. Durfee and P.S. Shoucair. Comparison of floating gate neural network memory cells in standard VLSI CMOS technology. *IEEE Transactions on Neural Networks*, 3(3):347–353, May 1992.

[DTJ91] L.A. Dicarlo, R.D. Throne, and J.M. Jenkins. A time–domain analysis of intracardiac clcctrograms for arrhythmia dctcction. *PACE*, February:329–336, 1991.

[Dub89] D.B Dubin. *Rapid Interpretation of EKG's: A Programmed Course.* Cover Publishing, 4th edition, 1989.

[EDT89] S. Eberhardt, T. Duong, and A. Thakoor. Design of parallel hardware neural network systems from custom analog VLSI "building block" chips. In *IEEE INNS International Joint Conference On Neural Networks*, volume 2, 1989.

[ET90] S. Eberhardt and A. Thakoor. A learning neural network from analog VLSI building blocks. In *Proceedings of the First Australian Conference on Neural Networks*, page 123, Sydney, Australia, 1990.

[FLS84] T. Fahraeus, C. Lassvik, and C. Sonnhag. Tachycardias initiated by automatic antitachycardia pacemakers. *PACE*, pages 1049–1054, November 1984.

[FYN91] S. Farrugia, H. Yee, and P. Nickolls. Neural network classification of intracardiac ECGs. In *International Joint Conference on Neural Networks*, pages 1278–1287, Singapore, Spring 1991.

[GH90] H.P. Graf and D. Henderson. A reconfigurable CMOS neural network. In *Proceedings of the IEEE International Solid–State Circuits Conference*, pages 144–145, 1990.

[Gil89] R.F. Gillum. Sudden coronary death in the United States. *Circulation*, 79:756–765, 1989.

[GJH88] H.P. Graf, L.D. Jackel, and W.E. Hubbard. VLSI implementation of a neural network model. *IEEE Computer Magazine*, 21(3):41–49, March 1988.

[Hen90] P.C.B. Henderson. *Analog Correlators in Standard MOS Technology.* University of Sydney PhD Thesis, 1990.

[HJ61] R. Hooke and T.A. Jeeves. Direct search solution of numerical and statistical problems. *JACM*, 8:212–229, 1961.

[HN92] Y. Horio and S. Nakamura. Analog memories for VLSI neurocomputing. In E. Sanchez-Sinencio and C. Lau, editors, *Artificial Neural Networks – Paradigms, Applications and Hardware Implementation*, pages 344–363. IEEE Press, 1992.

[HP90] P.W. Hollis and J.J. Paulos. Artificial neural networks using MOS analog multipliers. *IEEE Journal of Solid-State Circuits*, 25(3):849–855, June 1990.

[HSW89] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

[HTCB89] M. Holler, S. Tam, H. Castro, and R. Benson. An electrically trainable artificial neural network (ETANN) with 10240 "floating gate" synapses. In *IEEE INNS International Joint Conference On Neural Networks*, volume 2, 1989.

[Inc90] Motorola Inc. *MC68HC711D3 Technical Data.* Motorola, 1990.

[Jul88] D.G. Julian. *Cardiology.* Bailliere Tindall, 5th edition, 1988.

[JW84] M.E. Josephson and H.J.J. Wellens. *Tachycardias: Mechanisms Diagnosis Treatment.* Lea and Febiger, 1984.

[JWA79] J.M. Jenkins, D. Wu, and R.C. Arzbaecher. Computer diagnosis of supraventricular and ventricular arrhythmias. *Circulation*, pages 977–985, November 1979.

[Kra90] A. Kramer. Eeprom device as an analog storage element for neural networks. In *Proceedings of the First Australian Conference on Neural Networks*, page 131, Sydney, Australia, 1990.

[KW90] D.S. Khoury and B.L. Wilkoff. Tachycardia recognition algorithms for implantable systems. *IEEE Engineering in Medicine and Biology*, pages 40–42, June 1990.

[LDJ88] D. Lin, L.A. Dicarlo, and J.M. Jenkins. Identification of ventricular tachycardia using intracavitary electrograms: analysis of time and frequency domain patterns. *PACE*, pages 1592–1606, November 1988.

[Lee89] S.C. Lee. Using a translation-invariant neural network to diagnose heart arrhythmia. In *Proceedings of IEEE Engineering in Medicine and Biology Society 11th Annual International Conference*, pages 2025–2026, 1989.

[Lee90] S.C. Lee. Using a translation-invariant neural network to diagnose heart arrhythmia. In D.S. Touretzky, editor, *Advances in Neural Information Processing Systems 2*, pages 240–247. Morgan Kaufann Publishers, 1990.

[LHMM76] A. Langer, M.S. Heilman, M.M. Mower, and M. Mirowski. Considerations in the development of the automatic implantable defibrillator. *Med Instrum*, 10:163–167, 1976.

[Lip87] R. Lippmann. An introduction to computing with neural nets. *IEEE ASSP Magazine*, pages 4–22, April 1987.

[LJDM88] D.P. Lin, J.M. Jenkins, L.A. Dicarlo, and R.S. MacDonald. Arrhythmia diagnosis using morphology and timing from atrial and ventricular leads. *Computers in Cardiology*, pages 159–162, 1988.

[LSY91] B.W. Lee, B.J. Sheu, and H. Yang. Analogue floating–gate synapses for general purpose VLSI neural computation. *IEEE Transactions on Circuits and Systems*, 38:654–658, June 1991.

[Ltd89] Telectronics Pty Ltd. *Guardian ATP 4210 – Clinician's Information.* 1989.

[LV92] P.H.W. Leong and J.G. Vasimalla. *JIGGLE User's Manual.* Sedal Technical Report PL-0892, Sydney University Electrical Engineering, 1992.

[MC83] H.J.L. Marriott and M.H.B Conover. *Advanced Concepts in Arrhythmias.* C.V. Mosby, 1983.

[Mea89] C. Mead. *Analog VLSI and Neural Systems.* Addison-Wesley Publishing Company, 1989.

[MF86] A.D. Mercando and S. Furman. Measurement of differences in timing and sequence between two ventricular electrodes as a means of tachycardia differentiation. *PACE*, November–December:1069–1078, 1986.

[MMK+81] J.D Maloney, M. Masterson, D. Khoury, R. Trohman, B. Wilkoff, T. Simmons, V. Morant, and L. Castle. Long term effectiveness of dual–demand sequential pacing for automatic conversion and prevention of refractory reenty supraventricular tachycardias (abstr). *American Journal of Cardiology*, 47:476, 1981.

[MMK+91] J. Maloney, M. Masterson, D. Khoury, R. Trohman, B. Wilkoff, T. Simmons, V. Morant, and L. Castle. Clinical performance of the

implantable cardioverter defibrillator: Electrocardiographic documentation of 101 spontaneous discharges. *PACE*, 14:280–285, February, Part II 1991.

[MS88] A.F. Murray and A.V.W. Smith. Asynchronous VLSI neural networks using pulse-stream arithmetic. *IEEE Journal of Solid-State Circuits*, 23(3):688–697, June 1988.

[NH77] M. Nygards and J. Hulting. Recognition of ventricular fibrillation utilizing the power spectrum of the ECG. *Computers in Cardiology*, pages 393–397, 1977.

[NM65] J.A. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, 7:308–313, 1965.

[PF88] F. Pannizo and S. Furman. Frequency spectra of ventricular tachycardia and sinus rhythm in human intracardiac electrograms–application to tachycardia detection for cardiac pacemakers. *IEEE Transactions on Biomedical Engineering*, 35(6):421–425, June 1988.

[RM86] D.E. Rumelhart and J.L. McClelland. *Parallel Distributed Processing: Explorations in the microstructure of cognition*, volume I. MIT Press, Cambridge, Massachusetts, 1986.

[RMB+87] J. Raffel, J. Mann, R. Berger, A. Soares, and S. Gilbert. A generic architecture for wafer–scale neuromorphic systems. In *Proceedings of IEEE International Conference on Neural Networks*, pages 501–513, San Diego, USA, June 1987.

[SC80] D.B. Scott and S.G. Chamberlain. Modeling and experimental simulation of the low–frequency transfer inefficiency in bucket brigade devices. *IEEE Transactions on Electron Devices*, ED–27(2):405–414, February 1980.

[SHH89] D.B. Schwartz, R.E. Howard, and W.E. Hubbard. A programmable analog neural network chip. *IEEE Journal of Solid-State Circuits*, 24(2):313–319, April 1989.

[SJSL88] C.D. Schuger, K. Jackson, R.T. Steinman, and M.H. Lehmann. Atrial sensing to augment ventricular tachycardia detection by the automatic implantable cardioverter defibrillator: A utility study. *PACE*, 11:1456–1464, October 1988.

[SS76] R.A.J. Spurrell and E. Sowton. Pacing techniques in the management of supraventricular tachycardias. part 2. an implantable atrial synchronous pacemaker with a short atrioventricular delay for the prevention of paroxysmal supraventricular tacycardias. *Journal of Electrocardiology*, 9:89–96, 1976.

[Sze81] S.M. Sze. *Physics of Semiconductor Devices, 2nd ed.* Wiley, 1981.

[TDJW90] R.D. Throne, L.A. Dicarlo, J.M. Jenkins, and S.A. Winston. Paroxysmal bundle branch block of supraventricular origin: A possible source of misdiagnosis in detecting ventricular tachycardia using time domain analyses of intraventricular electrograms. *PACE*, pages 453–468, April 1990.

[TDNC91] M.A. Tooley, D.W. Davies, A.W. Nathan, and A.J. Camm. Recognition of multiple tachyarrhythmias by rate–independent means using a small microcomputer. *PACE*, February:337–340, 1991.

[TJD90a] R.D. Throne, J.M. Jenkins, and L.A. Dicarlo. The bin area method: A computationally efficient technique for analysis of ventricular and atrial intracardiac electrograms. *PACE*, pages 1286–1297, October 1990.

[TJD90b] R.D. Throne, J.M. Jenkins, and L.A. Dicarlo. Evaluation of area–of–difference algorithms for detecting ventricular tachycardia using electrogram morphology. In *Proceedings of the Annual Conferences of the IEEE Engineering in Medicine and Biology Society*, volume 12–2, pages 0584–0585, 1990.

[Tro90] P. Troup. Implantable cardioverter–defibrillator devices. *Progress in Cardiology*, 3/2, 1990.

[Tro91] P.J. Troup. Programmed stimulation–guided therapy compared with implantable cardioverter defibrillator device therapy in the treatment of ventricular tachyarrhythmias. *PACE*, 14:267–272, February, Part II 1991.

[TVJ92] L. Toivonen, M. Viitasalo, and A.J. Jarvinen. The performance of the probability density function in differentiating supraventricular from ventricular rhythms. *PACE*, 15:726–730, 1992.

[VFS+80] C.J. Vandepol, A. Farshidi, S. Spielman, A.M. Greenspan, L.N. Horowitz, and M.E. Josephson. Incidence and clinical significance of induced ventricular tachycardia. *American Journal of Cardiology*, pages 725–731, April 1980.

[Win91] R.A. Winkle. State–of–the–art of the AICD. *PACE*, 14:961–966, May, Part II 1991.

[WLL88] D.F. Wong, H.W. Leong, and C.L. Liu. *Simulated Annealing for VLSI Design*. Kluwer, 1988.

[WM86] J. Warren and R.O. Martin. Clinical evaluation of automatic tachycardia diagnosis by an implanted device. *PACE*, 9:1079–1083, November–December 1986.

[Wyn91] C.R.C. Wyndham. *An Electrophysiological Approach to the Diagnosis of Arrhythmias.* Futura, 1991.

[XJ92] Y. Xie and M. Jabri. Training limited precision feedforward neural networks. In *Proceedings of Third Australian Conference on Neural Networks*, pages 68 71, 1992.

[ZT88] Y.S. Zhu and N.V. Thakor. Detection of ventricular fibrillation by sequential testing. *Computers in Cardiology*, pages 325–328, 1988.

Publications

Patents

- University of Sydney (P.H.W. Leong and M.A. Jabri), "A Method and System for Automatically Classifying Intracardiac Electrograms", *Complete Australian Patent Application 25253/92*, September 1992.

Chapters in Books

- M. Jabri, S. Pickard and P. Leong "Classification of Arrhythmias using Artificial Neural Networks", *invited to contribute to a book "Neural Network Applications" to be published by Kluwer (Europe), Alan F. Murray editor.*

Journals

- P.H.W. Leong and M. Jabri, "MATIC - An Intracardiac Tachycardia Classification System", Pacing & Clinical Electrophysiology (PACE), Sept 1992, USA.

- Philip Leong, "Current source scrounges parts", *EDN*, pp. 182, June 6, 1991.

- M. Jabri, S. Pickard, P. Leong and Y. Xie, "Algorithms and Implementation Issues in Analog Low Power Learning Neural Network Chips", *accepted for*

publication in the International Journal on VLSI Signal Processing, USA 1992.

Full Length Conference Papers

- P. H. W. Leong and M. A. Jabri, "Connection Topologies for Digital Neural Networks", *Proceedings of the Second Australian Conference on Neural Networks*, pp. 34–37, February 1991.

- M. Jabri, S. Pickard, P. Leong, B. Flower, G. Rigby, and P. Henderson "VLSI Implementation of Neural Networks with Application to Signal Processing", *Proceedings of IEEE's International Symposium on Circuits and Systems*, pp 1275-1278, Singapore 1991.

- P. H. W. Leong and M. A. Jabri, "Arrhythmia Classification Using Two Intracardiac Leads", *Proceedings of Computers in Cardiology 1991*, pp. 189–192, IEEE Computer Society Press, 1992.

- P. H. W. Leong and M. A. Jabri "An Analogue Low Power VLSI Neural Network", *Proceedings of the Third Australian Conference on Neural Networks*, pp 147–150, Canberra, Australia 1992

- S. Pickard, M. Jabri, P.H.W. Leong, B.G. Flower and P. Henderson "Low Power Analogue VLSI Implementation of A Feed-Forward Neural Network", *Proceedings of the Third Australian Conference on Neural Networks*, pp 88-91, Canberra, Australia 1992.

- P.H.W. Leong and M. Jabri "An Analogue Low Power VLSI Neural Network", *Proceedings of the International Joint Conference on Neural Networks*, Volume II-6, pp 78-683, Baltimore, USA 1992.

- M. Jabri, S. Pickard, P. Leong, B. Flower and Y. Xie "ANN Based Classification of Heart Defibrillators", *Advances in Neural Information Processing Systems 4*, pp 637–644, Morgan Kaufmann, 1992.

- P. H. W. Leong and M. A. Jabri "A VLSI Arrhythmia Classifier", *accepted for publication in the Fourth Australian Conference on Neural Networks*, Melbourne, Australia 1993

Glossary

AF abbreviation for atrial fibrillation

ANN abbreviation for artificial neural network

AV node abbreviation for atrioventricular node

arrhythmia any abnormal heart rhythm

atria singular form of atrium

atrial fibrillation supraventricular tachycardia causing twitching of the atria with an absence of atrial contractions

atrioventricular node part of the heart's conduction system which delays transmission of the action potential

atrium chanber in the heart which receives blood from the veins and pumps blood into the ventricles

BBB abbreviation for bundle branch block

BBD bucket brigade device

bigeminy aberrant beats which occur after every normal beat

bradycardia slow heart rate (< 60 beats per minute)

bundle branch block a delay in the ventricular conduction system

CSA abbreviation for combined search algorithm

cardioversion restoration of a heart's rhythm to normal by an electrical shock

EPS abbreviation for electrophysiological studies

ectopic aberrant

electrophysiological studies procedure whereby temporary catheters are inserted in the surface of the heart and tachyarrhythmias induced by stimulation so that the conduction sequence of the heart can be studied

HRA abbreviation for high right atrium

high right atrium location in the atria in which an ICD lead is often placed

ICD abbreviation for implantable cardioverter defibrillator

ICEG abbreviation for intracardiac electrogram

implantable cardioverter–defibrillator permanently implanted device which can deliver shock therapy in event of a tachycardia

intracardiac electrogram time recording of the potentials within the chambers of the heart

morphology shape of the intracardiac electrogram (usually refers to the shape of the QRS complex)

NSR abbreviation for normal sinus rhythm

normal sinus rhythm heart rhythm originating from the sinoatrial node

PP interval between successive P waves

PR interval between the last P and R waves

P wave deflection on an ICEG occurring when depolarisation of the atria occur

paroxysmal sudden onset

RR interval between successive R waves

R wave deflection on an ICEG occurring when depolarisation of the ventricles occur

RVA abbreviation for right ventricular apex

refractory period period after depolarisation in which no further depolarisation can occur

retrograde conduction conduction of the wave of depolarisation in a reverse direction

right ventricular apex location in the ventricle in which an ICD lead is often placed

SA node abbreviation for sinoatrial node

ST abbreviation for sinus tachycardia

STD abbreviation for simple threshold detector

SVT abbreviation for supraventricular tachycardia

sinoatrial node the heart's pacemaker

sinus tachycardia heart rhythm greater than 100 beats per minute but otherwise normal

supraventricular tachycardia tachycardia originating in the atria or atrioventricular node

tachycardia fast abnormal heart rhythm ($>$ 100 beats per minute)

tachyarrhythmia tachycardia

trigeminy ectopic beats which occur after every two normal beats

VF abbreviation for ventricular fibrillation

VT abbreviation for ventricular tachycardia

VT 1:1 abbreviation for ventricular tachycardia with 1:1 retrograde conduction

VTF abbreviation for fast ventricular tachycardia

ventricle bottom part of the heart which pumps arterial blood

ventricular fibrillation tachycardia causing twitching of the ventricular muscle and no cardiac output

ventricular tachycardia tachycardia originating in the ventricles