A LOW POWER TRAINABLE ANALOGUE NEURAL NETWORK CLASSIFIER CHIP

Philip H. W. Leong and Marwan A. Jabri

Sydney University Electrical Engineering, NSW 2006 Australia

1 ABSTRACT

This paper describes an analogue VLSI chip called "Kakadu" which implements a trainable (10,6,4) multilayer perceptron. Kakadu is a classifier designed for low power applications and has a typical power consumption of 20 μW. Kakadu has been tested on many classification problems including XOR, 4 bit parity, character recognition and arrhythmia classification.

2 INTRODUCTION

Artificial neural network (ANN) architectures use a parallel interconnection of simple processing elements in order to perform computation. ANNs have been successfully applied to many classification problems and several very low power implementations of analogue neural networks have been reported [1, 2].

An implantable cardioverter–defibrillator (ICD) is a permanently implanted device which monitors the heart and delivers shock therapy in the event of life–threatening arrhythmias. Current ICDs rely on very simple classification techniques since the more sophisticated techniques cannot be implemented in the strict power budget that must be met.

This paper describes an analogue neural network chip designed to meet these criteria. The chip, called Kakadu, is an implementation of a three layer perceptron which contains 84 synapses and has ultra low power consumption. Although designed for the purpose of arrhythmia classification, Kakadu is a reasonably general purpose classifier and classification results on benchmark problems will be described in this paper.

3 KAKADU

The Kakadu neural network chip is a (10,6,4) perceptron implemented using 6 bit multiplying digital to analogue converters as synapses and off–chip resistors as neurons. It was fabricated using the Orbit Semiconductor 1.2 μm double metal single poly process on a 2.2 × 2.2 mm die. A photomicrograph showing the main synapse blocks, row shift registers and the current source is shown in Figure 1.

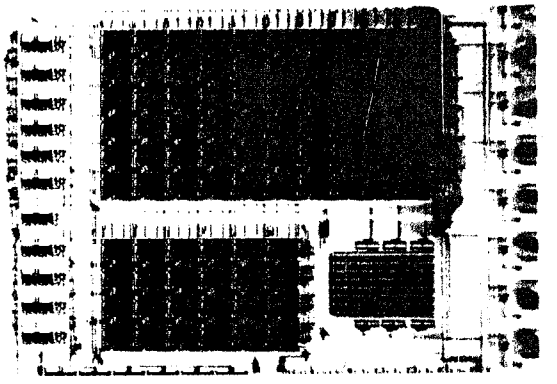

Figure 1: Photomicrograph of the Kakadu Test Chip.

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 Kakadu chip can be described by the following equations

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

$$a_i = \alpha u_i \quad (2)$$

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 is the total number of layers, N_l is the number of neuron units at the lth level and i is the neuron number ($1 \leq i \leq N_l$).

IEEE 1993 CUSTOM INTEGRATED CIRCUITS CONFERENCE

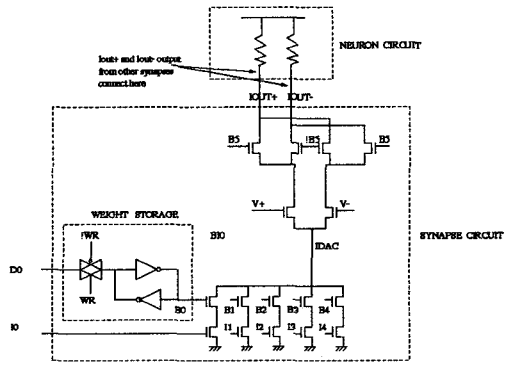

Figure 2: Synapse and Neuron Circuitry.

By placing the nonlinearity inside the summation, the tanh characteristic of a Gilbert multiplier can be used to implement this function.

4 TRAINING

The Kakadu chip was tested using the "Jiggle" test jig. Jiggle was designed by Sydney University Electrical Engineering and is a general purpose chip tester having 64 12 bit analogue input/output channels as well as 64 digital input/output channels. Jiggle connects to a VME bus, and the VME cage is interfaced to a Sun SPARCstation IPC via a Bit 3 Model 466 SBUS to VME converter. Jiggle allows arbitrary analogue or digital signals to be presented to the pins of the test chip and thus allows software control of the weight updating and training of the Kakadu chip. All results in this paper were obtained at a bias current of 6.63 nA and a supply voltage of 3 V.

The training approach was one of optimisation in which the mean squared error of all the patterns was reduced. The Combined Search Algorithm [3] (CSA) was used for all of the results in this paper. CSA employs two minimisation strategies namely modified weight perturbation [4] 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 + 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 = random number;
            evaluate error;
            if error has not improved
                w = wsave;
        }
    }
}
```

5 RESULTS

5.1 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 2 is the current drawn for that particular synapse. A number of current consumption measurements were made for different weight values and then a least squares fit was used to derive the current consumption formula

$$I_{KAKADU} = 0.842 + 0.00736 \sum_{i=0}^{N} |w_i| \quad (\mu A) \qquad (3)$$

where 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.

Figure 3 shows the measured current dissipation of the chip and the curve fit of Equation 3 to this data. Note that the maximum current consumption of this chip occurs when all the weights are set to the maximum value and this value is 20 μA.

5.2 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

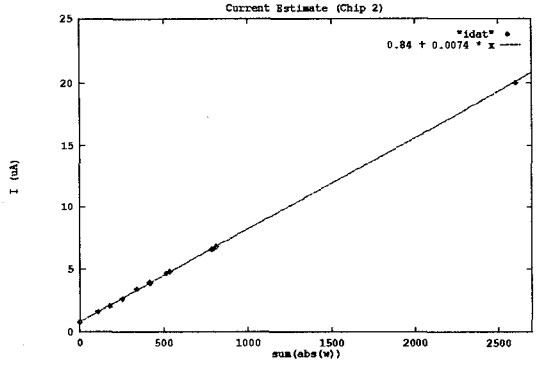

Figure 3: Current Consumption Curve fit (Bias Current = 6.63 nA)

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

test being shown in Table 1. 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.

5.3 Parity (4 Bit)

A highly nonlinear benchmark test often used for neural networks and training algorithms is the 4 bit parity problem. Given 4 bits of input, the ANN is trained to calculate whether the input is even or odd parity. This test was successfully trained using a (5,4,1) network. The margin was set to 0.08V. The quiescent power consumption for this problem was 15.6μW, and the results of this test are shown in Table 2.

5.4 Character Recognition

Kakadu was applied to a simple character recognition problem and the patterns and results are shown in Table 3. A (10,6,4) network was divided into a bias unit

Input					Desired Output	Output
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 2: Results of Applying Kakadu to the Parity 4 Problem (15.6 μW)

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	-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	-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 3: 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.

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 show that Kakadu was able to correctly classify patterns that it had not been trained on. Kakadu draws 22.5 μW during this test.

5.5 MATIC

The MATIC algorithm [5] classifies arrhythmias based on timing and morphological features. The timing features are classified using a decision tree and a neural network is used to recognise a certain class of arrhythmias called ventricular tachycardia with 1:1 retrograde conduction (VT 1:1) which can only distinguished by

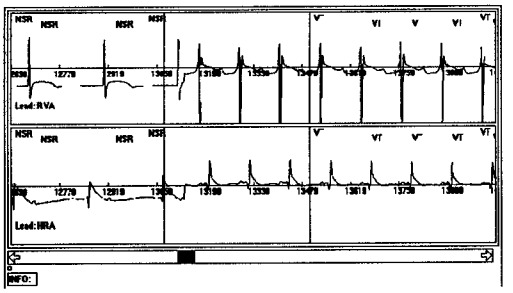

Figure 4: Ventricular Tachycardia with 1:1 retrograde conduction. Note how the morphology of the signal changes between normal rhythm (on the left) and VT 1:1 (on the right).

morphological features. An example of this situation can be seen in Figure 4, where the patient's arrhythmia changes from a normal rhythm to that of a VT 1:1, and a morphology change can be clearly seen.

The MATIC algorithm was tested on a database of 67 patients and the results of using Kakadu to perform morphology classification and a classification rate of 99.2% was achieved. A standard floating point software implementation of a neural network achieved a classification rate of 99.6% and a standard arrhythmia classification algorithm similar to those currently used in implantable defibrillators achieved 75.9% [6].

The power consumption of Kakadu for the 10 VT 1:1 patients is shown in Table 4. The maximum power consumption of the chip was 25 μW for the patients studied. The propagation delay of the Kakadu chip is approximately 30 μS, a normal heart rate is approximately 1 Hz, and Kakadu has negligible ($<$ 100 pA) power consumption at zero bias. If we allow a conservative value of 1000 μS for propagation, the average power consumption of the system can be reduced to less than 25 nW by turning off the bias of the chip when it is not being used.

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

Table 4: Power consumption and Morphology Parameters of Kakadu chip for the 10 VT 1:1 patients.

6 CONCLUSION

We have designed, fabricated and successfully tested a a trainable low power analogue VLSI neural network chip. This chip can solve a wide range of classification problems and has a typical power dissipation of tens of microwatts.

7 ACKNOWLEDGEMENTS

This work was supported in part by the Australian Department of Industry, Technology and Commerce and Telectronics Pacing Systems. Some of the patient data was supplied by Dr. Vince Paul, St. Georges Hospital, London.

References

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

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

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

[4] Marwan Jabri and Barry Flower. Weight perturbation: An optimal architecture and learning technique for analog vlsi feedforward and recurrent multilayer networks. *IEEE Transactions on Neural Networks*, 3(1):154–157, January 1992.

[5] P.H.W. Leong and M.A. Jabri. Matic – an intracardiac tachycardia classification system. *PACE*, September 1992.

[6] P.H.W. Leong and M.A. Jabri. A VLSI arrhythmia classifier. In *Proceedings of the Fourth Australian Conference on Neural Networks*, pages 41–44, Melbourne, Australia, 1993.