Synaptic Weight Generation in VLSI Stochastic Neural Networks.

J.G. Ortega, J.M. Quero, C.L. Janer and L.G. Franquelo

Escuela Superior de Ingenieros
Avda. Reina Mercedes s/n
Sevilla 41012 SPAIN

ABSTRACT

Fully parallel stochastic neural network implementations can be realized nowdays. However, in these implementations most of the silicon area is consumed in the stochastic pulse sequence generation circuits. In order to improve their efficiency in terms of consumed silicon area, new techniques must be developped. This is specially important in applications where a large number of synaptic weights are needed. In this paper we present a new approach that can significantly increase the efficiency of the technique that has been used up to now.

1. Introduction

Stochastic logic systems realize pseudoanalog operations using stochastically coded pulse sequences, [1], [2]. In stochastic systems, the terms that are to be processed are synchronous pulse sequences. Information is codified as the probability, at a given clock cycle, of the pulse taking "high" value. Stochastic pulse sequences are generated in such a way that all pulse streams are stochastically independent.

Consider now a set of n pulse streams whose probabilities, at a given clock cycle, of being at "high" level are p_1, p_2, ..., p_n. These probabilities are mutually independent. If these sequences are the inputs of a n-input AND gate, the probability of the gate output, at a given clock cycle, of being at "high" level is equal to $\prod_{i=1}^{i=n} p_i$. It is clear that the product operation is achieved by means of simple AND gates, that is, by a extremely low area consuming circuit.

Stochastic summation is a much more difficult operation to perform, specially if the terms to be added are signed. Two types of circuits have been described in the bibliography. One is the OR gate and the other is the up-down counter.

The up-down counter technique, although is widely used in neural network implementation, [3]-[5], has a very important drawback. Pulses coming from other neurons have to be multiplexed in time (i.e. *serialized*) leading to *high computation times.*

If two pulse sequences are the inputs of an OR gate and the pulse sequences to be added do not overlap, the output firing probability is equal to the addition of both firing probabilities. This OR-based add function is thus distorted by pulse overlap. In order to achieve a quasy linear behaviour pulse densities should stay very low, specially if many terms are to be added. This technique does not permit the integration of neurons with a very high number of synaptic connections as it would lead to extremely low maximum pulse density, [6]. It should be taken into account that the addition of two numbers that take values ranging from 0 to 1 may take a value bigger than one, which can not be represented by a probability.

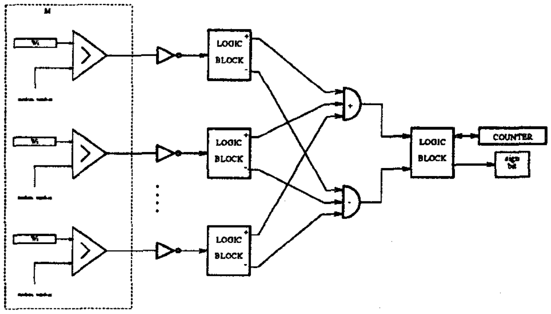

Fig. 1: Stochastic architecture.

In previous papers we have proposed a fully parallel stochastic computation architecture suitable for neural network implementation, [7], [8]. It circumvents one of the main drawbacks of stochastic computation architectures that have been used up to now: the absence of a space-efficient technique of adding weighted input signals in parallel. However still remains an important problem to be solved: to find a simple circuit that generates the stochastic signals.

0-7803-2768-3/95/$4.00 © 1995 IEEE

2. Stochastic Pulse Generation

A hardware implementation of a multilayer neural network based on this stochastic architecture was presented in a previous paper, [9]. This purely digital architecture is expandable, and the circuit was designed so that any multilayer network could be implemented by adding an appropriate number of I.C.s. However **most of the silicon area of this implementation is consumed in circuits involved in stochastic pulse generation of the synaptic weights** (block M of Fig.(1)).

In this implementation, stochastic streams of pulses are generated using a well known technique that has been broadly used and described in the technical literature: Digital codifications of these weights are digitally stored and then compared with uncorrelated random numbers producing uncorrelated stochastic signals, [2]-[5]. Therefore load registers, digital comparators and a pseudorandom generator are necessary.

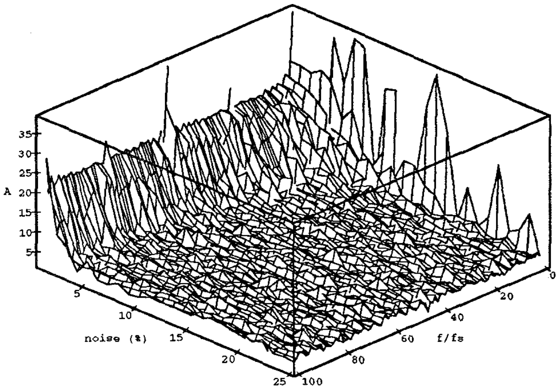

Fig. 2: Pulse sequence correlation.

To improve space efficiency we proposed a new technique to produce the random pulse sequences that codify the synaptic weights, [10]. The basic stochastic cell of this technique is a high frequency oscillator whose $\frac{T_{on}}{T}$ rate can be controlled, and a lower frequency sampling circuit. The $\frac{T_{on}}{T}$ rate is fixed equal to the synaptic weight that is meant to be stochastically codified. **If the oscillator's input capacitor has a small value, the oscillator will be very sensitive to noise**. Consequently, there is a certain degree of uncertainty about the voltage at oscilator's input, producing time uncertainty about the moment in which the oscilator switches from either the on state to the off state or from the off state to the on state. It follows that oscillator's phase cannot be predicted after it has switched several times. In Fig.(2) it is shown the autocorrelation of the pulse sequence, (A), as a fuction of the noise intensity and the oscillation-sampling frequencies ratio (f/fs). This figure has been obtained by simulation considering that the noise is white and gaussian. If the oscilator's output is sampled with a flip flop at a slow enough rate, the sampled signal will be random. The probability of this signal taking the "high" level will be equal to $\frac{T_{on}}{T}$.

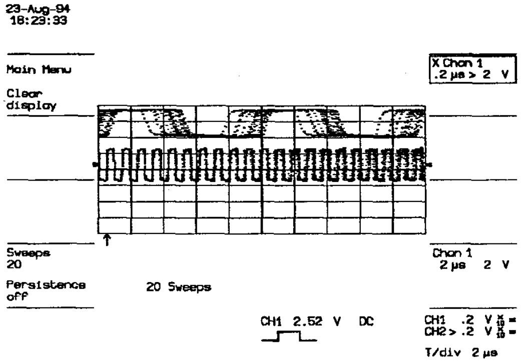

Fig. 3: Phase noise.

In order to produce complete spatial (between different pulse sequences) and time (in each pulse sequence) randomness, phase uncertainty must be greater or equal to 2π. Following these ideas, we designed test board using **discrete components**. Naturally the maximum oscillating frequency was rather modest, but the measured cross-correlation between different sampled oscillators and the autocorrelation were very encouraging. In Fig.(3) we show how noise acumulates producing phase uncertainty. We decided to face the **VLSI implementation** of the proposed circuit as we intend to develop applications including a large number of neurons and synaptic weights per neuron.

3. VLSI Implementation

The oscillator consists on five consecutive C-mos inverters. The output of the fifth inverter is conected to the first inverter's input, leading to an unstable circuit that oscilates at a high frequency. V_1 and V_2 are voltage signals that are used to fix the T_{on} and T_{off} values. The output is conected to a flip-flop that samples it. The flip-flop clock signal is suplied via an input digital pad. The whole circuit is plotted in Fig.(4).

The two voltages could be either internally fixed in applications where learning is not an esential feature, or could also be adjusted by additional hardware if learning is to be included in the considered application.

The overall design consists on:

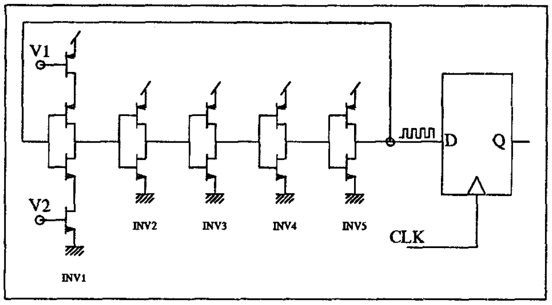

Fig. 4: Basic cell.

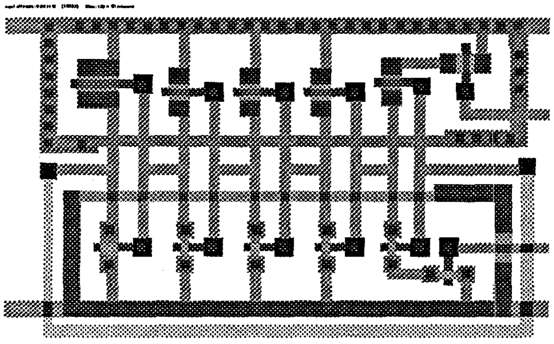

Fig. 5: Oscillator's layout.

1. An oscillator whose V_1 and V_2 voltages can be externally fixed. It permits to generate pulse streams in a wide range of densities. The size of this oscillator is 129 × 80 microns. The size of the flip-flop is 180 × 100microns. The oscillator's layout can be seen in Fig.(5).
2. Three oscillator whose V_1 and V_2 are fixed by a Mos transistor voltage divider.
3. Eight equall oscillators have been included in this design. In these oscillators V_1 voltage is fixed to 5V, yielding a 2.5ns T_{on} time. V_2 voltages are conected to eight analog pads so that the eight T_{off} times can be fixed externally.

All oscillator cells have guard rings in order to prevent, or, at least, minimize, coupling between the different circuits. The circuit has been designed using ES2 $1.5\mu m$ technology and the software package was MAGIC. The size of the whole circuit, including the sampling flip-flops, is 743×736 microns.

4. Experimental Results.

Fig. (6) shows the existing relationship between the control voltage V_2 and the $\frac{T_{off}}{T}$ ratio. V_1 has been fixed to 0V. Notice that $\frac{T_{off}}{T}$ takes only values ranging from 0 to 0.5 because 0.5 to 1 ratios can be obtained by means of an inverter gate.

In order to find out whether spatial proximity of the basic cells is related to high cross-correlation numbers, we have calculated the cross-correlation between the sampled stream of pulses of one of the basic cells and the rest of them. The obtained values are shown in Tab.(4). These calculations have been carried out three times (columns a, b and c). The sample frequency is 100kHz, the pulse sequence length is 1000 and the $\frac{T_{off}}{T}$ ratio is 0.5. It can be seen that stochastically independent pulse sequences can be obtained provided that the basic cells are not placed too closed from each other. The whole layout is shown in Fig.(7). The '0' cell is placed in the upper-left corner of the layout, '1', '2' and '3' cells are placed below. The '4', '5', '6' and '7' cells are placed at the right side of them. An increase of the sampling frequency does not lead to higher cross-correlation values. However the time-correlation numbers increase as the switching frequency becomes higher. In Fig. (8) it is shown the time-correlation of pulse sequences sampled from the '0' and '5' basic cells.

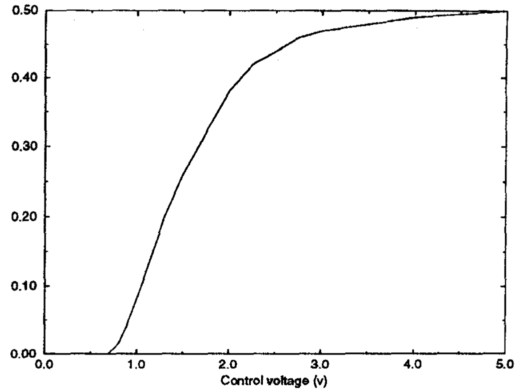

Fig. 6: $\frac{T_{off}}{T}$ regulation plot.

Osc.	a	b	c
0-1	-0.605882	-0.424706	-0.408235
0-2	-0.022353	-0.092941	-0.043529
0-3	-0.147059	-0.043529	-0.064706
0-4	0.012941	-0.029412	-0.036471
0-5	-0.015294	-0.041176	0.015294
0-6	-0.002353	0.001176	0.038824
0-7	-0.010588	0.010588	-0.005882

Table: 1: Cross-correlation results ($f_{sample} = 100kHz$).

In many stochastic applications time-uncorrelation is not an essential feature (if two streams are to be multiplied by means of an

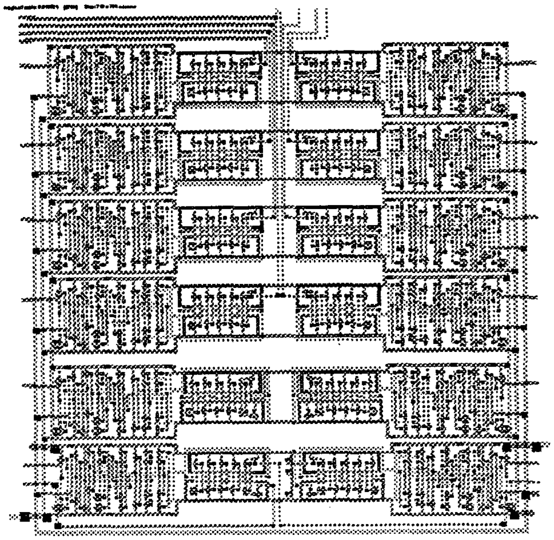

Fig. 7: Circuit layout

AND gate only spatial correlation is needed). In such cases this technique may be used leading to space-efficient implementations. We are currently improving the temporal statistical behavior of the pulse streams.

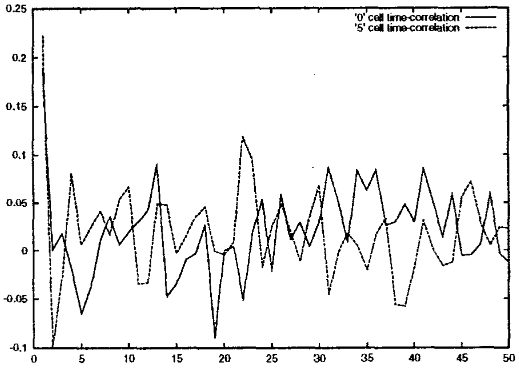

Fig. 8: Time-correlation results ($f_{sample} = 100kHz$).

5. Conclusions and Further Work

In this paper we have presented a new technique to generate stochastic pulse sequences that is suitable for VLSI implementation purposes. The basic cell consists on a high frequency oscillator and a sampling flip-flop. The consumed silicon area is very small, specially if it is compared with the strictly digital approach that has been considered up to now. The spatial correlation values of different pulse sequences are quite satisfactory. However the time-correlation behavior should be improved. In order to achieve this goal, we are going to include a noise source in the ring oscillator and use higher scales of integration which will lead to smaller input capacitors.

References

[1] B.R. Gaines. Stochastic Computing Systems. *Advances in Information Systems Science*, vol.2, pp. 37-172. 1969

[2] Y. Kondo and Y. Sawada. Functional Abilities of a Stochastic Logic Neural Networks *IEEE Trans. on Neural Networks*, vol.3, pp.434-443, 1992.

[3] D.E. Van den Bout and T.K. Miller III. A Digital Architecture Employing Stochaticism for the Simulation of Hopfield Neural Nets. *IEEE Trans. on Circuit and Systems*, vol.36, pp. 732-738. 1989

[4] W. Wike, D.E. Van den Bout and T.K. Miller III The VLSI Implementation of STONN. *IEEE Int. Joint Conf. on Neural Networks*, vol.2, pp.593-598, 1990.

[5] J.M. Quero, C.L. Janer and L.G. Franquelo. Constrained Hopfield Neural Network for Real-Time Predictive Control. *Proc. of the 1994 IEEE Int. Conf. on Industrial Electronics, Control and Instrumentation*, Iecon'94. Bologna, Sept. 1994.

[6] Alan F. Murray, Dante Del Corso and Lionel Tarassenko. Pulse-Stream VLSI Neural Networks Mixing Analog and Digital Techniques *IEEE Trans. on Neural Networks*, vol.2,no.2, pp.193-204, 1991.

[7] C.L. Janer, J.M. Quero and L.G. Franquelo. Fully Parallel Summation in a New Stochastic Neural Network Architecture. *IEEE Int. Conf. on Neural Networks*, San Francisco, pp. 1498-1503. 1993

[8] C.L. Janer, J.M. Quero, J. Ríos, J.G. Ortega and L.G. Franquelo. Design Criteria for Fully Parallel Stochastic Neural Network. *ICECS'94 El Cairo (Egypt)*, Dec., 1994.

[9] J.M. Quero, J.G. Ortega, C.L. Janer and L.G. Franquelo. VLSI Implementation of a fully parallel stochastic Neural Network. *IEEE Int. Conf. on Neural Networks*, Orlando (USA), July, 1994.

[10] J.G. Ortega, J.M.Quero, C.L. Janer and L.G .Franquelo. Interfaces to Stochastic Logic: Application to Stochastic Neural Network. *ICECS'94 El Cairo (Egypt)*, Dec., 1994.