Stochastic and PWM coding for an efficient implementation of Cellular Neural Networks

F.Colodro, A.Torralba, R.G.Carvajal and L.G.Franquelo
Dpto. de Ingeniería Electrónica,
Escuela Superior de Ingenieros
Avda. Reina Mercedes, s / n, SEVILLA–41012 (Spain)
Tlf: 34–5–4556849, Fax: 34–5–4629205
e–mail: pcolr@gte.esi.us.es

Abstract— **This paper present the application of Pulse Stream Techniques (PSTs) to the hardware implementation of a Cellular Neural Network. The time differential equation of this networks suggests that the dynamic of one neuron status can be emulated by adding discretized packets of charge to a capacitor. This task can be carried out by driving a current source with a pulse stream signal.**

I. Introduction

Different strategies of design have been used for integrated electronic implementation of Neural Networks (NNs). Based on digital or / and analog technologies, many arquitectures have been described in the literature. The advantages and drawbacks of analog and digital computing are well–known. To overcome these drawbacks, *Pulse Stream Techniques* (PSTs) have been proposed [1]. PSTs take the advantages of both, digital and analog implementations, as the information is carried by digital signals and analog circuitry is used to control them.

The most common PSTs are Stochastic Logic (SL) and Pulse Width Modulation (PWM). In SL, a signal is represented by a stochastic pulse stream which takes the value 1 with a probability proportional to the instantaneous value of the signal. The main advantage of SL relies on the fact that the product of two stochastic signals can be implemented by means of a simple AND gate. On the other hand, there is not a direct way to implement the summation in a digital way. In [2], a digital circuit called F was proposed, which is able to make the summation of n stochastic pulses in one clock cycle, at the cost of some hardware complexity. A more efficient implementation can be made if current–mode analog summation is used, as shown bellow.

If one of the two stochastic signals in the input of a AND gate is replaced by a *Pulse Width Modulation* (PWM) signal, the AND gate still carries out the multiplication operation. The product signal can be used to switch on / off a constant current which feeds a capacitor. In this way it is possible to carry out the sysnapses product of a Neural Network. Based on this principles, the basic architecture of one neuron is presented in this paper. In section II the architecture is used to build an efficient implementation of a *Cellular Neural Network* (CNN). A detailed description of the basic modules and the result obtained from simulations are presented in section III, and finally some conclusions are discussed in section IV.

II. The neuron state of a CNN

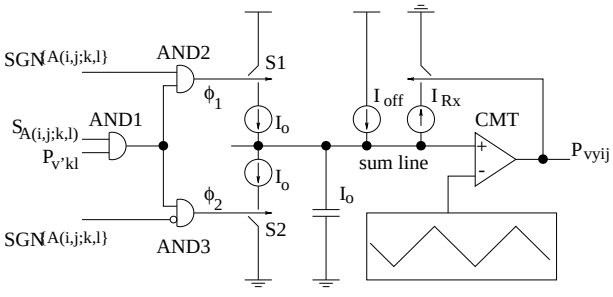


Figure 1: Block diagram of a neuron

A Cellular Neural Network (CNN) is an analog dynamic processor array, where the processing elements interact within a finite local neighborhood. The original CNN paradigm was first proposed by Chua and Yang in [3] and [4]. A concise tutorial on CNNs can be found in [5]. A review of selected communications in the field of CNN theory and applications, and a simulator with many examples, can be found in [6]. Integrated implementation of CNNs can be found in [7]–[15]. An implementation based on Pulse Stream current mode can be found in [16]

The differential equations governing the (i, j)–th neuron in a CNN is given by [3]:

$$C\frac{d\ v_{xij}}{d\ t} = -\frac{1}{R_x}v_{xij} + \sum_{C(k,l)\ \in N_r(i,j)} A(i,j;k,l) * v_{ykl} + \sum_{C(k,l)\ \in N_r(i,j)} B(i,j;k,l) * v_{ukl} + I_{off} \quad (1)$$

$$v_{yij} = f(v_{xij}) \quad (2)$$

where v_{xij}, v_{yij} and v_{uij} are the status, output and input of cell (i,j), respectively. C and R_x are CNN parameters. I_{off} is a bias term. $N_r(i,j)$ is a neighborhood of radius r around the (i,j)–th cell. $A(i,j;k,l)$ and $B(i,j;k,l)$ are the feedback and forward matrices associated to the cell (i,j), respectively [3]. The piece-wise linear function f is a saturated activation function. The time discretized charging equations (1) and (2) of a CNN is implemented electronically by the circuit shown in figure 1. This architecture reproduces the behavioural of one unipolar neuron. (Note that CNNs with bipolar neurons are easily converted to unipolar neurons, with only a change in the bias term I_{off}).

The stochactic signal $S_{A(i,j;k,l)}$ and the PWM signal $P_{v_{ykl}}$ are pulse streams which represent the values of $A(i,j;k,l)$ and v_{ykl}, respectively. So, the gate $AND1$ performs the product $A(i,j;k,l) \times v_{ykl}$. The product pulse drives a bipolar current source and, depending on the sign of $A(i,j;k,l)$, charges o discharges a summation capacitor. In a space–invariant CNN, the templates A and B are shared by all the neurons, so that the stochastic pulses $S_{A(i,j;k,l)}$ and $S_{B(i,j;k,l)}$ are obtained only once in a common module and later propagated to the rest of the chip with a considerable saving in silicon area. Note that a broadcast propagation is possible in our implementation, because these pulse streams are digital signals.

III. Basic modules of the neuron state

PWM signals are associated to neuron outputs. PWM coding has been selected, because it can be implemented with a local ramp generator and a comparator (figures 1 and 2). The comparator and the local ramp generator are used to generate the PWM signal $P_{v_{yij}}$, which is fedback to drive the slave part of a high–swing cascade current mirror (Fig.- 2). In this way it is possible to take into account the effect of the resistor R_x (equation 1) using the current I_{R_x}. A dummy transistor was used to reduce the charge–injection problem; the capacitor can be implemented by means of the gate capacitance of a transistor in a digital CMOS technology.

The circuit in figure 2 has been designed using a $1.0\mu m$

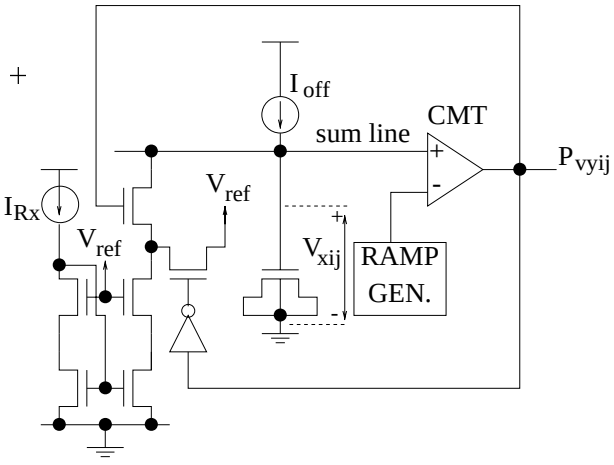
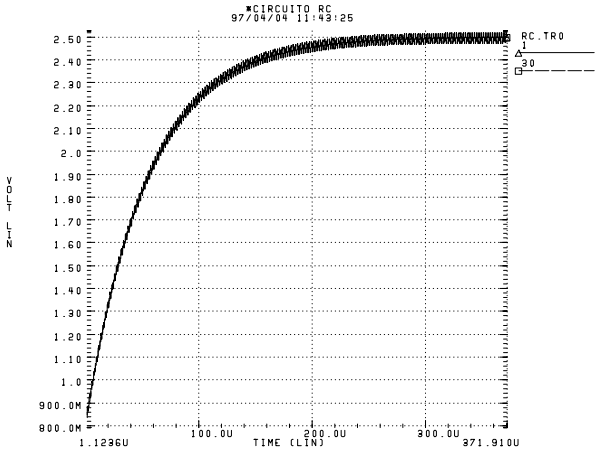


Figure 2: State of a CNN using a PST



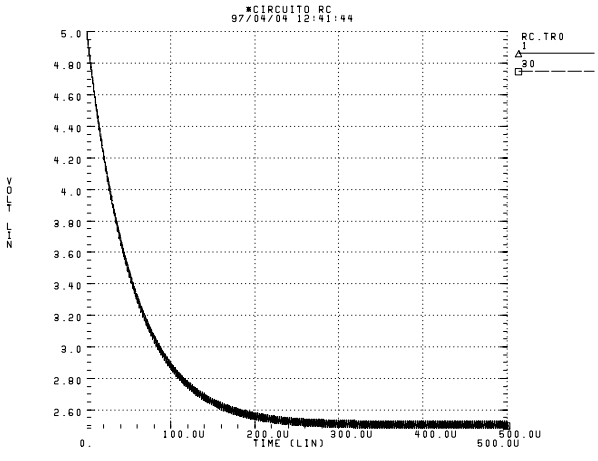


Figure 3: Continuous time dimamic of a circuit RC and the circuit in figure 2: a) Charging and b) Discharging.

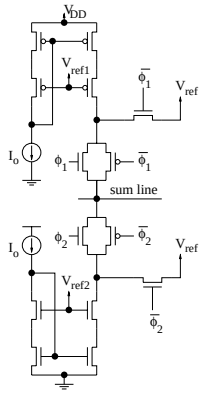


Figure 4: Current mirrors used for driving the cell capacitor.

$5V$, digital CMOS technology. The current I_{off} was adjusted to obtain a steady state value of $2.5V$ in the capacitor. The simulation results (obtained from HSPICE) of the circuit in Fig. 2 using two different initial values are depicted in figures 3.a and 3.b. The time evolution of a RC circuit for the same initial values is superimposed.

The current sources and the switches, which are driven by the products $A(i,j;k,l) \times v_{ykl}$ and $B(i,j;k,l) \times V_{ukl}$, have been built using two high–swing cascode current mirrors as shown in figure 4. The pulse stream signals ϕ_1 and ϕ_2 represent the synaptic products (outputs of the gates AND2 and AND3 in Fig. 1, respectivelly) associated to the cell in figure 4.

High–level language C simulations have shown that the computational strategy proposed in this paper is a good way for implementation Cellular Neural Networks. The simulated networks was able to successfully solve an edge extraction problem of a 21 $\times$ 21 picture.

The comparation of the evolution of the summation capacitance voltage when the results are obtained from language C and HSPICE simulations are shown in figures 5.a and 5.b in two different cases. In both cases, the same stochastic signals associated to the synaptic weight were used in C and HSPICE. In this way we avoid that the simulation results were affected by different behaviour due to the random nature of the stochastic signals. The neighbor pixel values were chosen so that the final values of the cell state voltage was $0V$ in the first case (Fig. 5.a) and $5V$ in the second case (Fig. 5.b). Both figures verify that the pulse stream computational strategy can be implemented using a CMOS technology while preserving high accuracy.

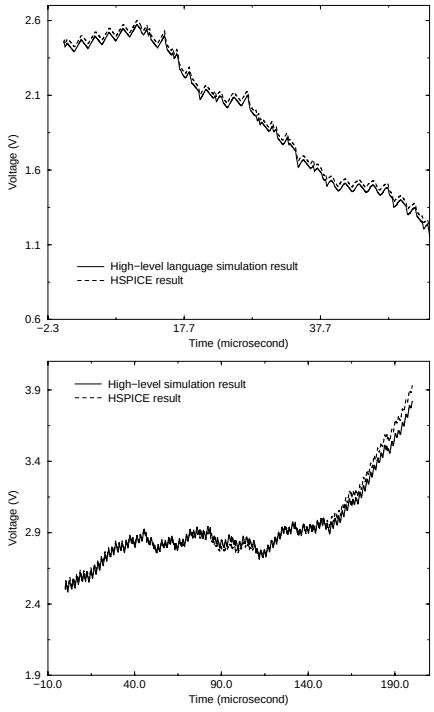


Figure 5: Comparation between the language C and HSPICE results when the final values of the cell state voltage is (a) $0V$ and (b) $5V$

IV. Conclusions

The application of Pulse Stream Techniques (PSTs) to the hardware implementation of Cellullar Neural Networks has been presented in this paper. The main features of the proposed architecture are:

- The information is codified by digital signals which are very robust againts noise and interferences.
- The stochastic nature of some signals allow to implement the multiplication using a simple AND gate. The summation is performed in a current mode approach. Both features allow to perform the computation using a simple circuitry with high accuaracy.
- The feedforward and feedback operators can be stored in digital memory. In case of CNNs with invariant templates, the number of these weights is small and the efficiency in silicon area is preserved.
- Due to the stochastic codification of these weights, they can be propagated through the ASIC without corruption using a reduced number of physical lines.

High–level language simulations have been done in order to check the behaviour of a network when it is implemented using PSTs. Results of transistor–level simulations have been compared with the results of high–level simulations for the case of one neuron and they have been demonstrated negligible non–ideal effect. PSTs seem to be a promising way to build large arrays with programmable synapses. Presently we are designing the neuron layout to build a 21×21 CNN array using a 1μ digital CMOS technology to experimentally demonstrate the advantages of the proposed architecture.

ACKNOWLEDGMENTS

This work has been supported by the Spanish Comisión Interministerial de Ciencia y Tecnología (CICYT), under project TIC96–0860.

REFERENCES

[1] A.Murray, D.del Corso and L.Tarassenlo. "Pulse–Stream VLSI neural networks mixing analog and digital techniques". *IEEE Trans. on Neural Networks*, vol. 2, n. 2, pp. 193–204, Mar. 1991.

[2] A.Torralba, F.Colodro. "Two digital circuits for a Fully Parallel Stochastic Neural Network". IEEE. *Trans. of Neural Network*, vol. 6, no 5, 1264–1268, SEP. 1995.

[3] L.O.Chua and L.Yang. "Cellular Neural Networks: Theory". *IEEE Trans. on CAS*, vol. 35, n. 10, pp. 1257–1272. Oct. 1988.

[4] L.O.Chua and L.Yang. "Cellular Neural Networks: Applications". *IEEE Trans. on CAS*, vol. 35, n. 10, pp. 1273–1290. Oct. 1988.

[5] L.O.Chua and T.Toska. "The CNN Paradigm". *IEEE Trans. on CAS–I*, vol. 40, n. 3, pp. 147–156. March 1993.

[6] *Cellular Neural Networks*. T.Roska and J.Vandewalle eds. John Wiley & Sons: Chichester 1993.

[7] J.M.Cruz, L.O.Chua. "A CNN chip for connected component detection". *IEEE Trans. on Circuits and Systems*, vol. 38, pp. 812–817, July 1991.

[8] H.Harrer, J.A.Nossek and R.Stelzl. "An analog implementation of discrete–time cellular neural networks". *IEEE Trans. on Neural Networks*, vol. 3, pp. 466–476, May 1992.

[9] A.Rodríguez Váquez, S.Espejo, R.Domínguez Castro, J.L.Huertas and E.Sánchez Sinencio. "Current–mode techniques for the implementation of continuous– and discrete–time cellular neural networks", *IEEE Trans. on Circuits and Systems–II*, vol. 40, pp. 132–146, Mar. 1993.

[10] J.E.Varrientos, E.Sánchez Sinencio and J.Ramírez Angulo. "A current–mode cellular neural network implementation". *IEEE Trans. on Circuits and Systems–II*, vol. 40, pp. 147–155, Mar. 1993.

[11] I.A.Baktir, M.A.Tan. "Analog CMOS implementation of cellular neural networks", *IEEE Trans. on Circuits and Systems–II*, vol. 40, pp. 200–206, Mar. 1993.

[12] G.F.D.Betta, S.Graffi, Z.M.Kovacs, G.Massetti. "CMOS implementation of an analogically programmable cellular neural network", *IEEE Trans. on Circuits and Systems–II*, vol. 40, pp. 206–215, Mar. 1993.

[13] M.Anguita, F.J.Pelayo, A.Prieto and J.Ortega. "Analog CMOS implementation of a discrete–time CNN with programmable cloning templates". *IEEE Trans. on Circuits and Systems–II*, vol. 40, pp. 215–218, Mar. 1993.

[14] A.Paasio, A. Dawidziuk and V. Porra "VLSI Implementation of Cellular Neural Network Universal Machine". *IEEE Int. Symp. on Circ. and Syst. ISCAS'97*, vol. 1, pp. 545–548, Jun. 1996.

[15] M.Salerno, F.Sargeni and V.Bonaiuto "6x6DPCNN: a programable mixed analogue–digital chip for Cellular Neural Networks". *Fourth IEEE Int. Work. on CNN and Their Appl.*, pp. 451–456, 1996

[16] Ari Paasio, Adam Dawidziuk and Veikko Porra. "Pulse Stream Current Mode CMOS CNN chip". *Fourth IEEE Int. Work. on CNN and Their Appl.*, pp. 457–460, 1996.