A Two-Stage Charge-Based Analog/Digital Neuron Circuit with Adjustable Weights

Alexandre Schmid, Yusuf Leblebici *, Daniel Mlynek

Swiss Federal Institute of Technology, Integrated Systems Center
CH - 1015 Lausanne, Switzerland

* Worcester Polytechnic Institute, Department of Electrical and Computer Engineering
100 Institute Road, Worcester, MA 01609, USA

alexandre.schmid@epfl.ch, leblebic@ece.wpi.edu, daniel.mlynek@epfl.ch

Abstract

A novel circuit-level neuron architecture based on the principle of analog charge-based computation of neural functions has been developed with the goals of high-speed processing, adjustable weights, and support of perturbation-based learning algorithms. The two-stage architecture which is composed of non-linear synapses, driving a linear capacitive soma, has been implemented using a conventional double-polysilicon CMOS technology. The feed-forward architecture of the proposed neuron model is shown to synthesize a large number of non-linear mappings of the 2D-1D space.

Introduction

Early developments in the field of artificial neural networks (ANNs) have relied on emulation of complex architectures and algorithms on standard serial computer architectures, due to a large demand in hardware resources, in terms of operators, memory, but also precision.

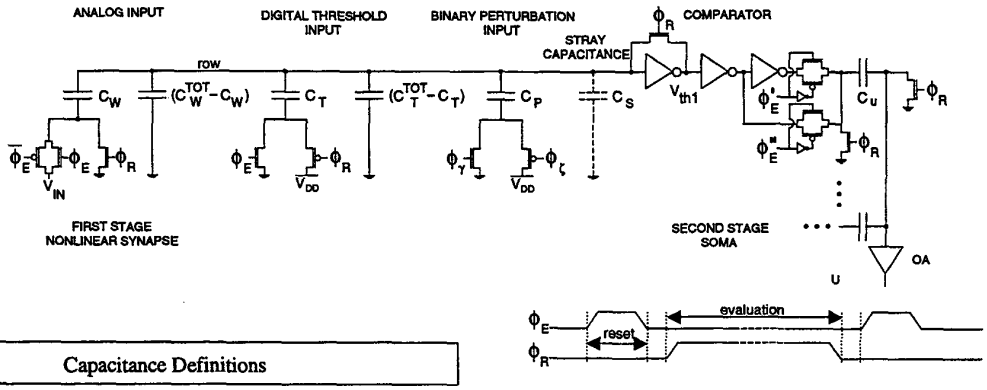
Analog hardware has emerged as a promising solution to the computation of neural operations [1]. Simple operations such as addition and multiplication, and also very complex ones such as the synthesis of non-linear activation function have a very efficient implementation in terms of processing speed, silicon area and power consumption, largely owing to the convenient exploitation of intrinsic electrical properties of integrated elements such as MOSFETs, capacitive or resistive arrays. Moreover, hardware friendly algorithms [2] have been developed so as to fit the algorithmic complexity to the specificities and limitations of hardware integration.

In this paper we demonstrate the use of the charge-based capacitive technique in the synthesis of a two-stage integrated neuron. In the next Sections, we are reviewing the neuron architecture and operation, as well as the study of standard 2D-1D mappings, which are advantageously synthesized with a feed-forward ANN (FFANN) organization of several neurons.

Architecture of the Two-Stage Capacitive Neuron

The charge-based neuron architecture consists of a first hard-limiting synaptic stage, driving a linear capacitive soma as a second stage (see Fig. 1). A fundamental difference with conventional neuron models lies in the non-linearity being placed at the synaptic level, rather than at the output. Synaptic non-linearity has already been proved to have an efficient hardware implementation [3], also supporting supervised learning of an adapted version of the widely used backpropagation algorithm. We show that our model, although different from this latter, also has the ability of synthesizing a wide range of nonlinear mappings, with the advantage of a VLSI-friendly mixed-analog/digital integration.

0-7803-5529-6/99/$10.00



Capacitance Definitions	
C_W	Capacitive programmable weight value
C_W^{TOT}	Upper limit of the capacitive weight range
C_T	Capacitive programmable threshold value
C_T^{TOT}	Upper limit of the capacitive threshold range
C_P	Capacitive perturbation value
C_S	Stray capacitance on the common node
C_U	Second-stage unit capacitance value

Control Signal Definitions	
Φ_R	Reset signal, precharge phase
Φ_E	Evaluation phase signal
Φ_γ	Identical to Φ_R or Φ_E, depending on the desired polarity of the applied perturbation
Φ_ζ	
Φ'_E	Identical to Logic 0 or Φ_E, depending on the desired polarity of the synapse
Φ''_E	

Fig. 1 : Schematic description of the two-stage charge-based neuron architecture, and corresponding driving clock signals.

The synaptic first stage is functionally composed of two programmable capacitances C_T and C_W, which are respectively in charge of implementing the threshold and weight of the neuron model, as well as one fixed capacitance C_P, in charge of inducing row perturbation according to learning algorithms [4]. All these capacitances have one common node called *row*, by analogy to the array of charge-based ANN circuits from which the proposed neuron architecture is derived [5], [6]. Each of the programmable capacitances is composed of a number of n elementary capacitances, each of which can be individually selected to compose the active programmed capacitance value, all unselected capacitances being tied to ground. The selectable weight capacitors are sized in binary increments, i.e., available capacitance values are $C_u \cdot 2^i$ where i=1,2,...,n and C_u is the unit capacitance.

The interface between the first and second stage consists of a chain of conventional full-CMOS inverters. The binary signal resulting from the operation of the first stage is selected at the output either of the second or third inverter, depending on the currently programmed synaptic polarity, which is a parameter resulting from the learning phase.

The second stage is an array of identical capacitances with convenient precharge circuitry, so as to process the sum of the first stage activity. Finally, an output operational amplifier (OA) is used to recover the full dynamic of the signal at the neuron output, to be connected to the next FFANN layer, an IC output pad, or an ADC for digital further processing (learning).

Operation of the Two-Stage Capacitive Neuron

A very simple two phase non-overlapping clock scheme consisting of a precharge (Φ_R) and evaluation phase (Φ_E) schedules the complete operation of the circuit.

All nodes connected to capacitances in the first stage are applied a voltage by a current source during the precharge phase, resulting in charge transfer onto the plates. The row in the first stage has its voltage imposed by the threshold value of the first comparator V_{th1}. The amount of charge transferred to the row is equal to :

$$Q_{reset} = V_{th1}C_W + V_{th1}\left(C_W^{TOT} - C_W\right) + \left(V_{th1} - V_{DD}\right)C_T + V_{th1}\left(C_T^{TOT} - C_T\right) + V_{th1}C_P + V_{th1}C_S \quad (1)$$

The row node is set in a floating state at completion of the precharge phase. The condition of charge conservation on the row node dictates that any change in the voltage applied to its converging capacitances external node during the evaluation phase :

$$Q_{eval} = \left(V_{row} - V_{IN}\right)C_W + V_{row}\left(C_W^{TOT} - C_W\right) + V_{row}C_T + V_{row}\left(C_T^{TOT} - C_T\right) + V_{row}C_P + V_{row}C_S \quad (2)$$

causes a perturbation of the row voltage equal to ΔV_{row} :

$$\Delta V_{row} = \frac{V_{IN}C_W - V_{DD}C_T \pm V_{DD}C_P}{C_W^{TOT} + C_T^{TOT} + C_P + C_S} \quad (3)$$

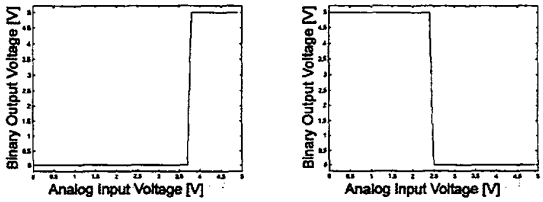
An analog input voltage V_{IN} is applied to the C_W capacitance; the resulting perturbation of the row voltage is proportional to both V_{IN} and C_W, thus processing the weighting of the input. The threshold capacitance C_T has a binary input changing from V_{DD} to GND at evaluation; the voltage shift being constant and negative has the similar role as a threshold in a conventional neuron model. The perturbation capacitance C_P is attributed the task of generating a sufficiently small voltage perturbation of the row, thus allowing the estimation of the error surface with respect to the variable parameters, the weight and threshold, in order to perform analog learning by descent of the gradient towards the minima of the error. The purely binary perturbation input is applied during evaluation without extra precharging. Both polarities of the perturbation are allowed, which is reflected in Equation 3 by the ± sign of $V_{DD}C_P$; in the proposed circuit however, the sign of perturbation has to be determined prior to precharge, thus reducing the circuit complexity, at the cost of lower algorithmic flexibility.

The second stage of the neuron is fully reset to GND during precharge. While in evaluation, each of the binary values resulting from the first stage activity is driving one unit capacitance, causing the OA input node to rise to a voltage value reflecting the weighted sum of the second stage inputs.

Synthesis of Transfer Functions and Surfaces

ANNs are widely used for their ability to synthesize non-linear mappings. The binary synapse architecture with programmable weights proves to be appropriate for the synthesis of a large set of nonlinear transfer functions, the quantification rate being adaptable, within the range dictated by hardware as well as algorithmic limitations. In this Section we focus on the synthesis of some basic transfer functions and surfaces which were obtained by software simulation of ideal synapse, neuron and FFANN models. Throughout this paper we have assumed an 8-bit capacitance resolution and a power supply of 5 V, which are the values chosen in the VLSI integration to be examined later.

The transfer function of a binary synapse is similar to a hard-limiter with programmable parameters. The switching point is adjustable with the C_T/C_W ratio, the polarity being also programmable. As can be seen on Fig. 2, the range of interesting capacitance values is limited to the case : $C_T \leq C_W$.

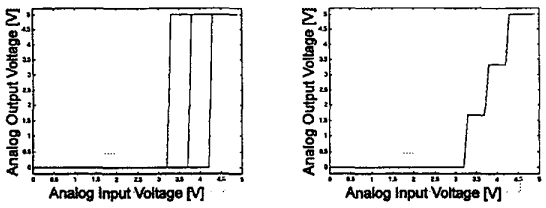


W=200 T=150 SP=1 W=200 T=100 SP=0

Fig. 2 : Simulation of the synaptic hard-limiting transfer function; W is the equivalent decimal weight, T is the equivalent decimal threshold and SP is the synapse polarity.

The basic neuron transfer function is a step-shaped line with a maximal number of quantified steps equal to the number of capacitances in the second stage. Each of the steps in the transfer function correspond to the Logic '1'/Logic '0' switch of the first stage comparator, resulting from the corresponding synaptic threshold being overshoot. Consequently, the one-synapse neuron has the same transfer function as its synapse (see Fig. 2).

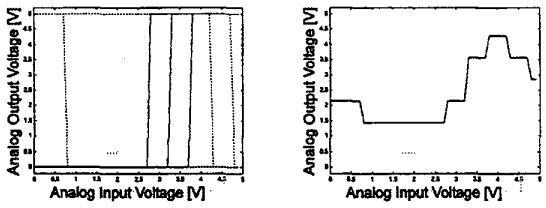
Connecting several synapses to the same input allows synthesizing complex quantified transfer functions. The principle underlying this technique is based on a convenient choice of each synapse's C_T/C_W ratio. A neuron having several synapses of slowly growing C_T/C_W ratios will have its second stage capacitances becoming active each in turn, resulting in a smooth quantified transfer function of adaptable slope and intercept, whereas a neuron with several synapses of identical C_T/C_W ratios will synthesize a very steep slope in the transfer function. Figure 3 illustrates these principles showing the transfer function of a three-synapses neuron, having all of its synapses connected to the same input.



W1=200 T1=130 SP1=1 W2=200 T2=150 SP2=1
W3=200 T3=170 SP3=1

Fig. 3 : Simulation of the neuron transfer function in the case of three synapses connected to the same input.

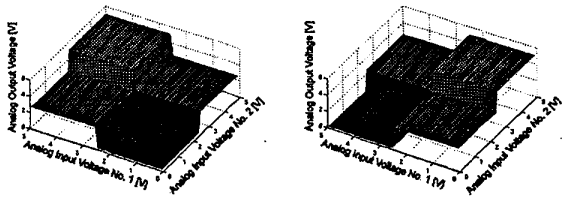
The selection of the inverse polarity for some of the synapses has two main consequences (see Fig. 4). The neuron acquires the ability of synthesizing non-monotonic functions; however, the cost is in terms of loss in the dynamics of the output signal, resulting from the theoretically guaranteed activity of some of the input neurons throughout the whole of the input range.



W1=200 T1=30 SP1=0 W2=200 T2=110 SP2=1
W3=200 T3=130 SP3=1 W4=200 T4=130 SP4=1
W5=200 T5=150 SP5=1 W6=200 T6=170 SP6=0
W7=200 T7=190 SP7=0

Fig. 4 : Right : simulation of a neuron transfer function in the case of single-input neuron of seven synapses; Left : simulation of the seven synapses transfer functions, the synapses with inverted polarity have a dotted line.

Considering the implementation of the proposed neuron model into a realistic FFANN, we have decided to devote a special study to the 2D-1D mapping, as a general case which is the basis of further investigation into more complex space mappings. The basic transfer function surface of a 2-input neuron of two synapses is shown on Fig. 5. The variation of the C_T/C_W ratios allows the transverse modulation of this surface. Also notice that the inversion of the synaptic polarity affects the surface as a symmetry/rotation operation.



W11=200 T11=100 SP11=1
W21=200 T21=100 SP21=1

W11=200 T11=100 SP11=0
W21=200 T21=100 SP21=1

Fig. 5 : Simulation of neuron transfer function; here the neuron has two independent inputs; Right : one synapse has an inverted polarity.

The connection of several synapses to each of the two inputs has a similar effect on the transfer function surface as seen previously (see Fig. 6).

Building a FFANN of the proposed neuron model, together with the careful application of the transfer function surfaces principles outlined here allows the synthesis of complex surfaces with reasonably small networks. Figure 6 (right) illustrates the surface obtained with a two-layer ANN of only three neurons.

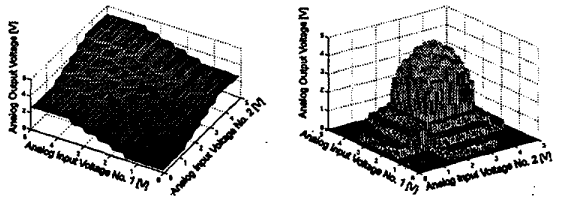


Fig. 6 : Left : Simulation of the neuron transfer function; here the neuron has two input and five synapses per inputs; Right : transfer function of a FFANN with three neurons.

VLSI Integration of the ANN Modules

The integration of the proposed circuit architecture was produced using a 0.8µm double-poly CMOS technology. The 8-bit programmable capacitances are realized as comb-shaped overlaps of the two polysilicon layers, the actual weight and threshold values are stored in the digital domain, allowing robust data recovery.

The microphotograph of the realized test structure is shown in Fig. 7.

The table below lists some of the IC specifications, and observed performances. The IC test strategy includes an HP82000 IC tester, as well as standard analog testing equipment.

IC Specifications and Preliminary Performances	
Technology	AMS-Thesys CXQ 0.8 µm CMOS (double poly)
Power Supply	5 V
Integrated Units	1 8-bit synapse 1 8-bit, 10-synapse neuron
Area : synapse neuron	0.165 mm^2 1.63 mm^2
I/O Pads : total (s. & n.) neuron	21 10 analog in / 1 analog out 8 dig. in / 1 dig. out (test)
Capacitances count/synapse	17
Unit Capacitance (1^{st} & 2^{nd})	50 fF
Transistor count/synapse	397
Cycle time characteristics : min. observed precharge min. observed latency	 18 ns 8 ns

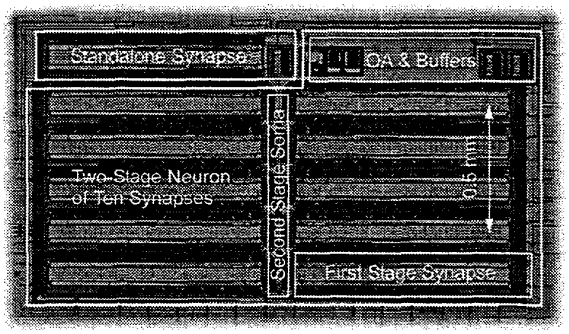
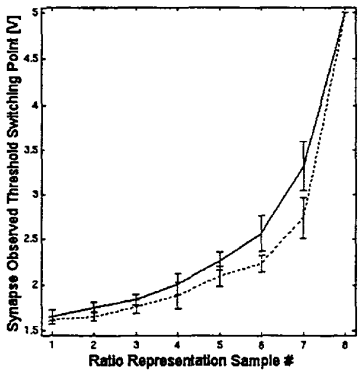


Fig. 7 : Microphotograph of the integrated structures, consisting of one standalone synapse and output driving circuitry, together with one 10-synapses two-stage neuron, and its I/O OA and drivers devices.

The functional tests have shown a correct behavior of both integrated units. The sensitivity tests show a large influence of stray capacitance, which dictates the actual limit to the circuit resolution. The precision of the synaptic threshold switching point, and the hysteresis effect due to the comparator can be seen on Fig. 8, emphasizing the best choice of large capacitance values to represent a given C_T/C_W ratio.

The functional test of the integrated neuron have demonstrated the validity of the principles of transfer function surface synthesis previously exposed. Fig. 9 illustrates the synthesis of a 10-steps pyramidal transfer function, where five synapses have an inverted polarity to create the down-slope.



Trace	Ratio
1	48/160
2	40/132
3	24/80
4	20/66
5	12/40
6	10/33
7	6/20
8	3/10

Fig 8 : Synthesis of a synaptic transfer function with an expected switching point at 1.5 V. The dotted line represents the results obtained with a negative slope ramp applied to the synapse, showing the hysteresis.

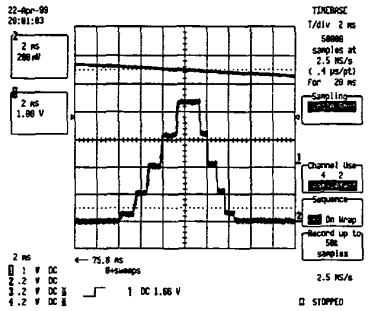


Fig. 9 : Observed synthesis of a pyramidal non-linear transfer function by the ten synapses neuron.

Conclusions

The circuit description and operation of an innovative charge-based neuron architecture with non-linear synapses are demonstrated. This novel analog-digital architecture takes advantage from both domains resulting in a robust neuron building block. The storage of parameters is secured in the digital domain, whereas the computation of neural functions (summation, multiplication, non-linear thresholding) as analog operators allows for fast processing and easy interfacing with the outside world.

A wide array of non-linear mappings with variable quantification steps are possible for high-speed applications. The analog nature of the circuit allows the consideration of high-speed applications in the neuro-fuzzy field.

References

[1] Carver Mead, Mohammed Ismail, *Analog VLSI Implementation of Neural Systems*, Kluwer Academic Publishers, The Netherlands, 1989.

[2] Perry Moerland, Emile Fiesler, Neural Network Adaptations to Hardware Implementations, Chapter E1.2:1-13 in *Handbook of Neural Computation*, Oxford University Publishing, 1997.

[3] Jerzy B. Lont, Analog CMOS Implementation of a Multi-Layer Perceptron with Nonlinear Synapses, Ph. D. Dissertation No. 10244, Swiss Federal Institute of Technology Zürich, 1993.

[4] Gert Cauwenberghs, A Fast Stochastic Error-Descent Algorithm for Supervised Learning and Optimization, Advances in Neural Information Processing Systems (NIPS) 5, Morgan Kaufmann, San Mateo, CA, 1993.

[5] Alexandre Schmid, Yusuf Leblebici and Daniel Mlynek, Hardware Realization of a Hamming Neural Network with On-Chip Learning, Proceedings of the 1998 IEEE International Symposium on Circuits and Systems ISCAS'98, Monterey, CA, 1998.

[6] Ugur Cilingiroglu, A Charge-Based Neural Hamming Classifier, IEEE Journal of Solid-State Circuit, Vol. 28, No. 1, January 1993.