An analog CMOS chip set for neural networks with arbitrary topologies

Lansner, John; Lehmann, Torsten

Published in:
I E E E Transactions on Neural Networks

Link to article, DOI:
10.1109/72.217186

Publication date:
1993

Document Version
Publisher's PDF, also known as Version of record

An Analog CMOS Chip Set for Neural Networks with Arbitrary Topologies

John A. Lansner and Torsten Lehmann

Abstract—**An analog CMOS chip set for implementations of artificial neural networks (ANN's) has been fabricated and tested. The chip set consists of two cascadable chips: a neuron chip and a synapse chip. Neurons on the neuron chips can be interconnected at random via synapses on the synapse chips thus implementing an ANN with arbitrary topology. The neuron test chip contains an array of 4 neurons with well defined hyperbolic tangent activation functions which is implemented by using "parasitic" lateral bipolar transistors. The synapse test chip is a cascadable 4×4 matrix-vector multiplier with variable, 10 bit resolution matrix elements. The propagation delay of the test chips was measured to 2.6 μs per layer.**

I. Introduction

SEVERAL approaches on *artificial neural network* (ANN) implementations in analog VLSI technology have been reported in the literature. Among other things flexible topology [3], [12], [11], differential capacitive weights storage [4], [10], [13], inner product multipliers [1], [2], [10] and hyperbolic tangent activation functions [9], [10] have been considered. In this paper, we have combined and perturbated the existing solutions with our own work to obtain an efficient general purpose ANN in analog VLSI. ANN's are often modeled as

$$\underline{y} = \underline{g}(\underline{\underline{w}}\,\underline{s}), \qquad \underline{s} = \left[\underline{y}^T, \underline{x}^T\right]^T \tag{1}$$

where $\underline{y}$ is the neuron activation vector, $\underline{x}$ is the input vector, $\underline{\underline{w}}$ is the connection strength (synapse) matrix and $\underline{g}$ is a nonlinear function (a squashing function) [8], [7]. Thus a hardware ANN could consist of a matrix-vector multiplier (a *synapse chip*) followed by a squashing function vector (a *neuron chip*); it turns out that this splitting of the synapses and the neurons on separate chips provides easy expandability for fully parallel systems [3], [7], [12]. In this paper, we present such an *analog CMOS chip set.*

II. The Hardware

The signal representation was chosen to ensure the desired cascadability: the neuron chip has current inputs and voltage outputs and the synapse chip has voltage inputs and current outputs. Using this current-voltage scheme, the outputs from several synapse chips can be connected to one neuron input, and the output from one neuron can be distributed to several synapse chips. Thus in principal, any ANN configuration can be made with these chips.

Manuscript received July 13, 1992; revised August 31, 1992.

The authors are with the Computational Neural Network Center, Electronics Institute, Technical University of Denmark, DK 2800 Lyngby, Denmark.

IEEE Log Number 9206625.

A. The Neuron Chip

We have chosen the hyperbolic tangent, **tanh**, as the activation function for two reasons: 1) Due to the exponential nature of bipolar transistors the **tanh** is simple to implement and hence well-defined; 2) it has a convenient gradient function which will make a future implementation of a learning algorithm for the ANN easy (simulations on required accuracy can be found in [7]).

The neuron chip contains an array of neurons. Each neuron has three stages as shown in Fig. 1(a)–(c). Because of the variable number of connected synapses per neuron, the neuron has to have an adjustable gain. The adjusted signal is transferred by a sigmoid function, the hyperbolic tangent.

The input current $i_{s,j}$ (cf. (6)) is converted to a voltage v' by an opamp with feedback. The feedback is a controlled *differential* resistance, R_{gain}, being the gain-term factor. The "Double-MOSFET" method [1], [2], [14] with four NMOS transistors in the non-saturation region is used. We have the converted voltage

$$v' = R_{\text{gain}} i_{s,j}, \qquad R_{\text{gain}} = \frac{1}{K_N \frac{W_g}{L_g} V_{\text{gain}}},$$
$$V_{\text{gain}} = V_{\text{gain1}} - V_{\text{gain2}}. \tag{2}$$

K_N, W_g, and L_g denote the transconductance parameter, the channel width, and the channel length of the four M_g transistors, respectively. V_{gain} controls the gain-term factor. To keep the transistors operating in the non-saturation region we have $V_{\text{gain1}}, V_{\text{gain2}} \in \{1V, 5V\} \Rightarrow V_{\text{gain}} \in \{0V, 4V\}$. The voltage v' is transferred by a hyperbolic tangent function to the voltage v_{out}. The **tanh** function is basically obtained from a differential pair of transistors. Using MOSFET transistors in the subthreshold mode is one possibility [9] but because of the signal levels we have instead chosen to use the "parasitic" *lateral bipolar transistors* inherent in a CMOS process, LPNP [5], operated in the active region. The difference current is given as a function of the voltage v',

$$i_{C1} - i_{C2} = I_{\text{bias}} \alpha \tanh(v'/(2V_t)) \tag{3}$$

where V_t is the thermal voltage and $\alpha = -i_C/i_E$, where i_E and i_C are the emitter- and lateral collector current, respectively, for a single LPNP. Because of the (vertical) substrate collector current we have $\alpha \approx 1/2$. The difference current is converted to a voltage by an opamp with feedback:

$$v_{\text{out}} = V_{\text{ref}} + R_{\text{tanh}} I_{\text{bias}} \alpha \tanh(v'/(2V_t)),$$
$$R_{\text{tanh}} = \frac{1}{K_N \frac{W_t}{L_t} V_{\text{tanh}}}, \qquad V_{\text{tanh}} = V_{\text{tanh1}} - V_{\text{tanh2}}. \tag{4}$$

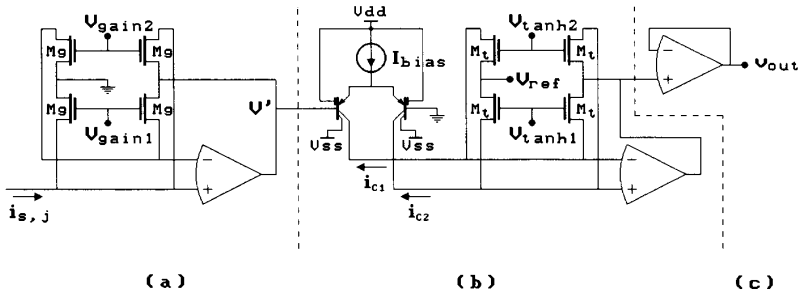


Fig. 1. (a) Input stage of a neuron, the adjustable current/voltage converter. (b) Transfer stage, the hyperbolic tangent function. (c) Output buffer.

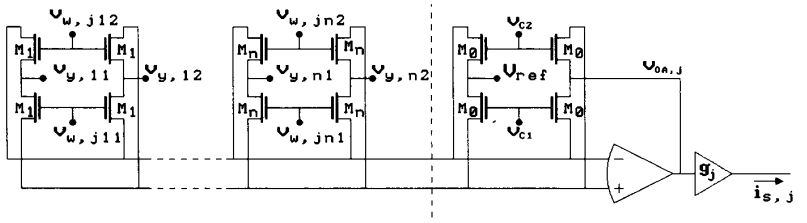


Fig. 2. Inner product vector multiplier. The $v_{w,jik}$'s and $v_{y,ik}$'s are input voltages and $i_{s,j}$ is the output current. In the MVM, the $v_{w,jik}$'s are used as matrix elements and the $v_{y,ik}$'s as elements of the input vector.

W_t and L_t are the channel width and length of the M_t's. $V_{\tanh}$ and I_{bias} control the magnitude of the output range. To keep the transistors working in the non-saturation region we have $V_{\tanh} \in \{0V, 4V\}$. V_{ref} controls the center of the output range.

The transfer function for a neuron is given by (2) and (4),

$$v_{\text{out}} = V_{\text{ref}} + R_{\tanh} I_{\text{bias}} \alpha \tanh(R_{\text{gain}} i_{s,j}/(2V_t)) \quad (5)$$

where R_{gain} and $R_{\tanh}$ are controlled by V_{gain} and $V_{\tanh}$ as stated in (2) and (4).

B. The Synapse Chip

The synapse chip is a parallel, cascadable, analog, CMOS *matrix-vector multiplier* (MVM) which is to be used both in the implementations of the ANN's and in the implementations of learning algorithms in the future. The synaptic weights are stored as differential voltages on capacitors—refreshed by a static RAM via a D/A converter [4], [13].

The $(m \times n)$ MVM consists of *m inner product vector multipliers* (IPM's) as shown in Fig. 2 [1], [2], [10]. (The MOS transistors are working in the nonsaturation region.) It can be shown [1] that the IPM output current ideally is given by

$$i_{s,j} = g_j \cdot (v_{OA,j} - V_{\text{ref}}) = \frac{g_j}{(W/L)_0 (v_{C1} - v_{C2})} \cdot \sum_{i=1}^{n} (W/L)_i (v_{w,ji1} - v_{w,ji2})(v_{y,i1} - v_{y,i2}) \quad (6)$$

where g_j is the transconductance of the output stage. The $(v_{w,ji1} - v_{w,ji2})$'s and $(v_{y,i1} - v_{y,i2})$'s are the voltage represented coordinates of the to input vectors, $v_C \stackrel{\text{def}}{=} (v_{C1} - v_{C2})$ is the control voltage for the "Double-MOSFET" feedback and V_{ref} is a reference voltage. The $(W/L)_i$'s are the width/length ratios of the M_i transistors. Setting $v_{y,i} \stackrel{\text{def}}{=} v_{y,i1} - v_{y,i2} \propto s_i$ for all the IPM's and $v_{w,ji} \stackrel{\text{def}}{=} v_{w,ji1} - v_{w,ji2} \propto \underline{w}_{ji}$ for the jth IPM gives the matrix-vector multiplier (cf. (1)).

To save pins, single-ended signals was selected on the chip (costing 1 bit of resolution); that is $v_{w,ji2} = v_{C2} = 2$ V and $v_{y,i2} = V_{\text{ref}} = -2$ V. To ensure good resolution and high noise rejection (at the cost of linearity), large input voltage levels were selected on the synapse chip: $|v_{w,ji}|_{\max} = |v_{y,i}|_{\max} = 1$ V. The transconductor was implemented with $g_j = 100$ μS.

As the high impedance $v_{w,jik}$ inputs of the IPM's are used as inputs for the matrix elements, these elements can be stored on the chip as charges on capacitors [4]. A *differential sampling scheme* [4] is used to write the matrix elements on the capacitors to reduce the effect of charge injection [6] and leakage currents. This way only four transistors and two capacitors are essentially needed for each matrix element, thus making the potential dimensions $((m \times n)_{\max})$ of the matrix large. The matrix *unit element* (a synapse) is shown in Fig. 3. In addition to the m IPM's, there is a row- and column-decoder on the synapse chip, which are used to address the synapses.

III. Experimental Results

A $k = 4$ input/output *neuron chip* and a $n = 4$ input, $m = 4$ output *synapse chip* has been fabricated to illustrate the principle of operation. A neuron chip with 100 neurons and a synapse chip with $\lesssim 100^2$ synapses should be feasible. The area overhead on the synapse chip caused by opamps, feedbacks, transconductors and address decoders is 224973 μm^2 (or presently $\approx 6 \times$ synapsearea) per row.

TABLE I
MEASURED CHIP CHARACTERISTICS

Property	Value	Bits	Notes
Neuron size	$A_{\text{neu}} = 379309\ \mu\text{m}^2$		
Neuron nonlinearity	$D_g \lesssim 2\%$	6 LSB_8	
Neuron derivative nonlinearity	$D_{dg} \lesssim 10\%$	26 LSB_8	
Neuron input offset	$\lvert I_{s\text{iofs}}\rvert \lesssim 10\ \mu\text{A}$	26 LSB_8	
Neuron output offset	$\lvert V_{g\text{ofs}}\rvert \lesssim 5\ \text{mV}$	1 LSB_8	
Neuron propagation delay[1]	$t_{g\text{pd}} \lesssim 1.8\ \mu\text{s}$	1/2 LSB_8	$C_L \approx 16\ \text{pF}$
	$t_{g\text{pd}} \lesssim 0.8\ \mu\text{s}$	1/2 LSB_1	$\approx 8C_{\text{in}}^{\text{synapsechip}}$
LPNP e/c current gain	$\alpha \approx 0.55$		
Synapse size	$A_{\text{syn}} = 33280\ \mu\text{m}^2$		Reducible by $\gtrsim 50\%$
Matrix offset	$\lvert V_{w\text{ofs}}\rvert \lesssim 16\ \text{mV}$	2 LSB_8	
Matrix resolution	$V_{w\text{res}} \lesssim 2\ \text{mV}$	1/4 LSB_8	
Synapse nonlinearity	$D_{wy} \lesssim 16\%$	21 LSB_8	
	$(D_{wy} \lesssim 3\%)$	(4 LSB_8)	Estimated
Synapse output offset	$\lvert I_{s\text{oofs}}\rvert \lesssim 14\ \mu\text{A}$	14 LSB_8	
Synapse input offset	$\lvert V_{y\text{ofs}}\rvert \lesssim 6\ \text{mV}$	1 LSB_8	
Synapse propagation delay[1]	$t_{s\text{pd}} \lesssim 2.0\ \mu\text{s}$	1/2 LSB_8	$C_L = 16\ \text{pF}, R_L \approx 10\ \text{k}\Omega$
	$t_{s\text{pd}} \lesssim 0.4\ \mu\text{s}$	1/2 LSB_1	$\approx 8C_{\text{in}}^{\text{synapsechip}}$
Matrix write time[2]	$t_{w\text{wr}} \lesssim 150\ \text{ns}$	1/8 LSB_8	
Matrix (weight) drift	$\lvert\delta_w\rvert \lesssim 0.5\ \text{mV/s}$	0.07 LSB_8/s	$C_{xk} = 1\ \text{pF}$
Weight range	$\lvert \underline{w}_{ji} \rvert_{\max} \in [0.4, 40]$		for $y_j = \tanh(s_j)$
Layer propagation delay[1]	$t_{\text{pd}} \lesssim 2.6\ \mu\text{s}$	1/2 LSB_8	$C_L \approx 16\ \text{pF}$
	$t_{\text{pd}} \lesssim 1.1\ \mu\text{s}$	1/2 LSB_1	$C_L \approx 16\ \text{pF}$

[1]Time from input change to output has settled within 1/2 LSB.
[2]Necessary length of write pulse that ensures the output will settle within 1/8 LSB_8

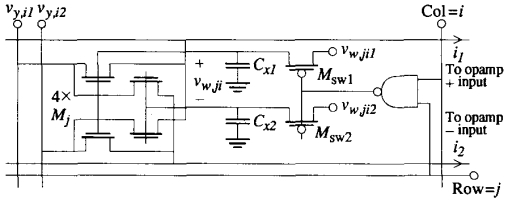


Fig. 3. Matrix unit element (synapse) that calculates $(v_{w,ji1} - v_{w,ji2}) \cdot (v_{y,i1} - v_{y,i2})$ and add this product as a differential current to the lines i_1 and i_2.

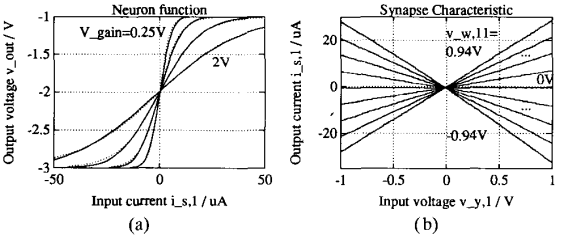


Fig. 4. Measurements. (a) Neuron transfer characteristics. The dotted lines are the desired tanh functions. (b) Synapse transfer characteristics.

A summary of the most *important properties* of the chips is shown in Table I. 1 LSB_X is one *least significant bit* for an X bit resolution of the appropriate signal. The *nonlinearity*, D of a quantity ξ is defined as the maximum deviation from the desired value: $D \stackrel{\text{def}}{=} \max_\xi |f(\xi) - \xi| / |\xi|_{\max}$ where $f(\cdot)$ is a nonlinear function. The offset errors and the nonlinearities cited in the table are caused by device mismatch (e.g., threshold voltage variations) and nonideal components (e.g., the channel mobility is field dependent) [14].

A measurement of the *neuron transfer characteristics* can be seen in Fig. 4(a). The maximum deviation from the desired tanh functions, D_g, is about 2% of the output range. The gain is adjustable with a range of 1:30 ($0.1\ \text{V} < V_{\text{gain}} < 3\ \text{V}$). The *derivative* of v_{out} with respect to $i_{s,j}$ has been compared to $d\tanh s/ds$. The deviation (D_{dg}) is less than 10% of the maximum value of $dv_{\text{out}}/di_{s,j}$.

The *synapse transfer characteristics* is shown in Fig. 4(b). The characteristics showed a good linearity ($D_{wy} \lesssim 3\%$ or 5 bits accuracy)—with the exception of the case with negative $v_{w,ji}$ values and positive $v_{y,i}$ values ($D_{wy} \lesssim 16\%$). This is due to the fact that it was necessary to lower V_{SS} to ensure a reasonable output current swing. The problem can be solved by improving the transconductor and the resulting nonlinearity is estimated to $D_{wy} \lesssim 3\%$. The synapse *matrix resolution* (i.e., the smallest $\Delta v_{w,ji}$ distinguishable at the output) was measured to $V_{w\text{res}} \lesssim 2\ \text{mV}$ or 10 bit at the least for a 2 V range of "matrix voltages" (note that we distinguish between resolution and accuracy). This should be sufficient for a range of ANN applications [7].

The *output offset currents* on the synapse chip and the *input offset currents* on the neuron chip are quite large. The reason could be that the opamps have low gains (< 60 dB), which together with opamp offset voltages of 2 mV would give the measured current offsets. This, however, is not necessarily a major problem (provided that the network is trained and used using the same chips) as the offset currents just displaces the neuron biases [8]. Likewise the matrix offset voltages could be used as small, random, initial weights when the network is trained. It should be noted that the offset errors are (mostly) nonsystematic.

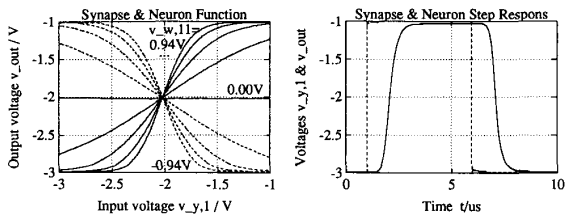


Fig. 5. Measurements. (a) Transfer characteristics for synapse plus neuron (corresponding to a layer in an ANN). $V_{\text{gain}} = 0.25$ V. (b) Step response for synapse plus neuron.

Finally measurements on two interconnected chips were made. In Fig. 5(a) the combined transfer characteristics of a synapse followed by a neuron is shown. The *step response* of the synapse-neuron combination is shown in Fig. 5(b). The delay through one layer of an ANN based on our chips can be measured on this curve: for an 8 bit output accuracy we have $t_{\text{pd}} \lesssim 2.6$ μs. Experimental results on an ANN based on the chip set are not yet available—a PC expansion board is under development and results should be available in the near future.

IV. Conclusions

In this paper we have presented two cascadable, analog CMOS chips: a neuron chip and a synapse chip. The chips have been tested and have shown excellent properties with respect to ANN applications:

The neuron function is well-defined, and the derivative can be calculated directly from the output voltage. LPNP-transistors work well as a differential pair. The adjustable gain ensures that the numbers of connected synapse inputs can be variable within a wide range.

The synapse matrix resolution is about 10 bits and the leakage currents in the capacitors holding the matrix elements are extremely small. The multiplication nonlinearities are probably of magnitudes that can be tolerated in some ANN applications, though it is a problem that must be solved.

The propagation time through the synapse and neuron chips is rather small (2.6 μs), even though the opamps are quite slow. And as the propagation time is essentially independent of the number of devices cascaded, it is possible to get a very high throughput using these chips. The offset errors on the chip set are rather large but it should be possible to reduce them somewhat.

In a *conclusion*, large, fast, accurate, analog *neural networks* with arbitrary topologies can be implemented by using full size neuron chips (with 100 neurons) and synapse chips (with 100^2 synapses).

Acknowledgment

This work was performed as parts of Ph.D. studies under the supervision of Prof. Erik Bruun. It was supported by the Danish Technical Research Council and the Danish Natural Science Council. Thanks are due to Thomas Kaulberg for the design of the amplifiers. The chips were fabricated through the EUROCHIP initiative.

References

[1] S. Bibyk and M. Ismail, "Issues in analog VLSI and MOS techniques for neural computing," in *Analog VLSI Implementation of Neural Systems.* C. Mead and M. Ismail, Eds., Norwell: Kluwer Academic Publishers, 1989, pp. 103–133.

[2] Z. Czarnul, "Novel MOS resistive circuit for synthesis of fully integrated continuous-time filters," *IEEE Trans. Circuits Syst.,* vol. 33, pp. 718–721, 1986.

[3] S. Eberhardt, T. Duong, and A. Thakoor, "Design of parallel hardware neural network systems from custom analog VLSI 'building block' chips," *IEEE Int. Joint Conf. Neural Networks,* pp. II-183–II-190, 1989.

[4] F.J. Kub, K.K. Moon, I.A. Mack, and F.M. Long, "Programmable analog vector-matrix multipliers," *IEEE J. Solid-State Circuits,"* vol. 25, pp. 207–214, 1990.

[5] E.A. Vittoz, "MOS transistors operated in the lateral bipolar mode and their application CMOS Technology," *IEEE J. Solid-State Circuits,* vol. 18, pp. 273–279, 1983.

[6] G. Wegmann, E.A. Vittoz, and F. Rahali, "Charge injection in analog MOS switches," *IEEE J. Solid-State Circuits,* vol. 22, pp. 1091–1097, 1987.

[7] T. Lehmann, "A hardware implementation of the real-time recurrent learning algorithm," in *10th European Conf. Circuit Theory and Design,* vol. 2, 1991, pp. 431–440.

[8] J. Hertz, A. Krogh, and R.G. Palmer, *Introduction to the Theory of Neural Computation.* Redwood City: Addison-Wesley, 1991.

[9] C. Mead, *Analog VLSI and Neural Systems.* New York: Addison-Wesley, 1989.

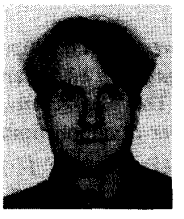
[10] E. Sánchez-Sinencio and C. Lau, *Artificial Neural Networks.* New York: IEEE Press, 1992.

[11] S. Satyanarayana, Y.P. Tsividis, and H.P. Graf, "A reconfigurable VLSI neural network," *IEEE J. Solid-State Circuits,* vol. 27, pp. 67–81, 1992.

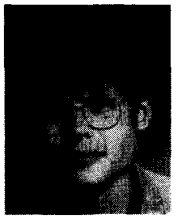
[12] P. Mueller, J. van der Spiegel, D. Blackman, T. Chiu, T. Clare, C. Donham, T.P. Hsieh, and M. Loinaz, "Design and fabrication of VLSI components for a general purpose analog neural computer," in *Analog VLSI Implementation of Neural Systems,* C. Mead and M. Ismail, Eds. Norwell: Kluwer Academic, 1989, pp. 135–169.

[13] Y. Tsividis and S. Satyanarayana, "Analogue circuits for variable-synapse electronic neural networks, *Electron. Lett.,* vol. 23, pp. 1313–1314, 1987.

[14] Y. Tsividis, M. Banu, and J. Khoury, "Continuous-time MOSFET-C filters in VLSI," *IEEE Trans. Circuits Syst.,* vol. 33, pp. 125–140, 1986.



John Arnold Lansner was born in Søllerød, Denmark, in 1966. He received the M.Sc. degree in electrical engineering from the Technical University of Denmark, Lyngby, in 1991. Currently, he is employed by CONNECT, the Computational Neural Network Center, working toward the Ph.D. degree at the Electronics Institute, Technical University of Denmark. His main topic is the implementation of artificial neural networks in analog VLSI technology.



Torsten Lehmann was born in Bagsvaerd, Denmark, in 1967. He received the M.Sc, degree in electrical engineering from the Technical University of Denmark, Lyngby, Denmark, in 1991 and is currently working toward the Ph.D. degree at the Electronics Institute, Technical University of Denmark. The work centers on analog VLSI implementations of learning algorithms for artificial neural networks.

His main research interests are in solid-state circuits and systems (analog and digital) and artificial neural networks.