An analog CMOS chip set for neural networks with arbitrary topologies

Lansner, John; Lehmann, Torsten

Published in:
IEEE 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

Link back to DTU Orbit

Citation (APA):
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 x 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

\[ y = g(wx), \quad z = \left[ y^T x^T \right]^T \]  

where \( y \) is the neuron activation vector, \( x \) is the input vector, \( w \) is the connection strength (synapse) matrix and \( g \) 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.

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 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 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 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 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 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 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 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.
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 $v_{w,j1}$'s and $v_{y,j2}$'s are input voltages and $i_{x,j}$ is the output current. In the MVM, the $v_{w,j1}$'s are used as matrix elements and the $v_{y,j2}$'s as elements of the input vector.

$W_1$ and $L_1$ are the channel width and length of the $M_1$'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

$$u_{out} = V_{ref} + R_{tanh}I_{bias} \tanh\left(\frac{R_{gain}i_{x,j}}{2V_{t}}\right)$$

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

$$i_{x,j} = g_j \cdot (V_{OA,j} - V_{ref}) = \frac{g_j}{(W/L)}(v_{C1} - v_{C2})$$

$$\sum_{j=1}^{n} (W/L)(v_{w,j1} - v_{w,j2})(v_{y,j1} - v_{y,j2})$$

where $g_j$ is the transconductance of the output stage. The $(v_{w,j1} - v_{w,j2})$'s and $(v_{y,j1} - v_{y,j2})$'s are the voltage represented coordinates of the to input vectors, $V_{C}$ is the control voltage for the “Double-MOSFET” feedback and $V_{ref}$ is a reference voltage. The $(W/L)$'s are the width/length ratios of the $M_1$ transistors. Setting $v_{y,j1} \equiv v_{y,j2} \equiv 0$ for all the IPM’s and $v_{w,j1} \equiv v_{w,j2} \equiv 0$, the $j$th 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 $v_{w,j1} = v_{w,j2} = 2V$ and $v_{y,j1} = V_{ref} = 2V$. To ensure good resolution and high noise rejection (at the cost of linearity), large input voltage levels were selected on the synapse chip: $v_{w,j1} = v_{w,j2} = 1V$. The transconductor was implemented with $g_j = 100 \mu S$.

As the high impedance $v_{w,j1}$ 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 $m = 4$ 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 $\leq 100^2$ synapses should be feasible. The area overhead on the synapse chip caused by opamps, feedbacks, transconductors and address decoders is 224973 $\mu m^2$ (or presently $\approx 6 \times$ synapsearea) per row.
TABLE I

MEASURED CHIP CHARACTERISTICS

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
<th>Bits</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neuron size</td>
<td>$A_{neur} = 379309 \text{ \mu m}^2$</td>
<td>6 LSBs</td>
<td></td>
</tr>
<tr>
<td>Neuron nonlinearity</td>
<td>$D_y \approx 2%$</td>
<td>26 LSBs</td>
<td></td>
</tr>
<tr>
<td>Neuron derivative nonlinearity</td>
<td>$D_{dy} \leq 10%$</td>
<td>26 LSBs</td>
<td></td>
</tr>
<tr>
<td>Neuron input offset</td>
<td>$</td>
<td>I_{i,offset}</td>
<td>\leq 10 \mu A$</td>
</tr>
<tr>
<td>Neuron output offset</td>
<td>$</td>
<td>V_{+}</td>
<td>\leq 5 \text{ mV}$</td>
</tr>
<tr>
<td>Neuron propagation delay(^1)</td>
<td>$t_{p,dy} \leq 1.8 \mu s$</td>
<td>1/2 LSBs</td>
<td></td>
</tr>
<tr>
<td>$t_{p,dy} \leq 0.8 \mu s$</td>
<td>1/2 LSBs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LPNPN e/c current gain</td>
<td>$\alpha \approx 0.55$</td>
<td></td>
<td>Reduced by $\geq 50%$</td>
</tr>
<tr>
<td>Synapse size</td>
<td>$A_{syn} = 33280 \text{ \mu m}^2$</td>
<td>2 LSBs</td>
<td></td>
</tr>
<tr>
<td>Matrix offset</td>
<td>$</td>
<td>V_{i,offset}</td>
<td>\leq 16 \text{ mV}$</td>
</tr>
<tr>
<td>Matrix resolution</td>
<td>$V_{wrest} \leq 2 \text{ mV}$</td>
<td>21 LSBs</td>
<td></td>
</tr>
<tr>
<td>Synapse nonlinearity</td>
<td>$D_{wy} \approx 16%$</td>
<td>4 LSBs</td>
<td></td>
</tr>
<tr>
<td>Synapse output offset</td>
<td>$</td>
<td>I_{j,offset}</td>
<td>\leq 14 \mu A$</td>
</tr>
<tr>
<td>Synapse input offset</td>
<td>$</td>
<td>V_{j,offset}</td>
<td>\geq 6 \text{ mV}$</td>
</tr>
<tr>
<td>Synapse propagation delay(^1)</td>
<td>$t_{p,dy} \leq 2.0 \mu s$</td>
<td>1/2 LSBs</td>
<td></td>
</tr>
<tr>
<td>$t_{p,dy} \leq 0.4 \mu s$</td>
<td>1/2 LSBs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Matrix write time(^2)</td>
<td>$t_{wrest} \leq 150 \text{ ns}$</td>
<td>1/4 LSBs</td>
<td></td>
</tr>
<tr>
<td>Matrix (weight) drift</td>
<td>$</td>
<td>V_{j}</td>
<td>\leq 0.5 \text{ mV/s}$</td>
</tr>
<tr>
<td>Weight range</td>
<td>$</td>
<td>V_{i}</td>
<td>_{\text{max}} \in [0.4, 40]$</td>
</tr>
<tr>
<td>Layer propagation delay(^1)</td>
<td>$t_{p,dy} \leq 2.6 \mu s$</td>
<td>1/2 LSBs</td>
<td>$C_{el} \approx 16 \text{ pF}$</td>
</tr>
<tr>
<td>$t_{p,dy} \leq 1.1 \mu s$</td>
<td>1/2 LSBs</td>
<td></td>
<td>$C_{el} \approx 16 \text{ pF}$</td>
</tr>
</tbody>
</table>

\(^1\)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 LSBs.

A summary of the most important properties of the chips is shown in Table I. 1 LSBX is one least significant bit for an X bit resolution of the appropriate signal. The nonlinearity, $D$, of a quantity $\xi$ is defined as the maximum deviation from the desired value: $D \equiv \max_{f(\xi)}|f(\xi) - \xi|/|\xi|_{\text{max}}$, 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, $D_{ty}$, is about 2% of the output range. The gain is adjustable with a range of 1:30 (0.1 V < $V_{\text{gain}} < 3$ V). The derivative of $v_{out}$ with respect to $i_{i,j}$ has been compared to $d \tanh s/ds$. The deviation ($D_{dy}$) is less than 10% of the maximum value of $dv_{out}/ds_{i,j}$. The synapse transfer characteristics is shown in Fig. 4(b). The characteristics showed a good linearity ($D_{wy} \approx 3\%$ or 5 bits accuracy)—with the exception of the case with negative $v_{w,j}$ values and positive $v_{y,i}$ values ($D_{wy} \approx 16\%$). This is due to the fact that it was necessary to lower $V_{SB}$ to ensure a reasonable output current swing. The problem can be solved by improving the transconductor and the resulting nonlinearity is estimated to $D_{wy} \leq 3\%$. The synapse matrix resolution (i.e., the smallest $\Delta v_{w,j}$ distinguishable at the output) was measured to $V_{wrest} \leq 2$ 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.
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 $t_{pd} \leq 2.6 \mu s$. 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 cascaded, 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 $\mu s$), 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


John Arnold Lanzner 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.