An Integrated Mixed-Mode Neural Network Architecture for Megasynapse ANNs

Johannes Schemmel, Felix Schürmann, Steffen Hohmann, Karlheinz Meier
Kirchhoff-Institute for Physics, University of Heidelberg, Germany, e-mail: schemmel@asic.uni-heidelberg.de

Abstract - This paper presents a new VLSI architecture for ANNs based on the combination of digital signalling and analog computing. It achieves a high level of parallelism as well as efficient area and power usage making very large networks possible. An implementation is presented combining 33k synapses and 256 neurons on 9 mm^2 of silicon area.

I. INTRODUCTION

To be useful in real-world applications like image-processing or data communications hardware neural networks should be at least as fast and easy to apply as software implementations. Furthermore their power consumption should not exceed that of conventional microprocessor-based (CPU) solutions. For these reasons most of the hardware realizations reported so far are based on digital signal processing using a medium degree of parallelism and an optimized internal data flow to achieve high performance [1][2]. Since they are essentially based on the same technology as state-of-the-art CPUs their speed versus power consumption ratio is of equal order of magnitude [3]. The incorporation of vector units that perform parallel operations on limited precision data (8 to 32 bits) into nearly all modern CPUs has reduced the architectural advantage of custom digital solutions significantly in the last five years [4].

Analog solutions on the other hand are mostly fully parallel neural network implementations. Their size is limited by the analog nature of their internal signals. The deterioration of the signal quality due to noise and distortion makes it difficult to build larger systems, especially at high data rates. The operational speed of their basic elements is usually much lower compared to digital solutions. Therefore a high level of parallelism is necessary [5]. Finally, it is difficult to interface them from a digital system.

Our group has chosen an approach in between these two opposite poles by combining fully analog network blocks with digital communication between several of these blocks. Each block is based on a feedforward network and can be modelled by the standard Perceptron formula. It consists basically of a two-dimensional array of synapses having input and output neurons attached to its sides. Due to the digital input (I_j) and output signals (O_i) the weight multiplication is reduced to an addition and the activation function $g(x)$ equals the Heaviside function $\Theta(x)$:

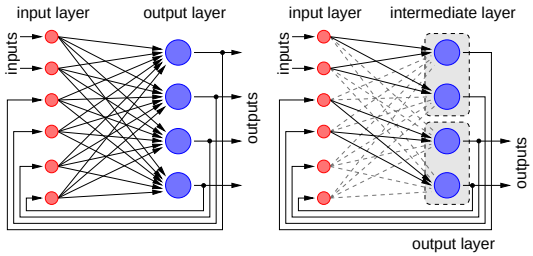

Fig. 1. **Left:** recurrent network, **right:** configured as a two-layer network by setting some synapses to zero (dashed lines).

$$O_i = g(\sum_j \omega_{ij} I_j), \quad g(x) = \Theta(x),\ I, O \in \{0,1\} \qquad (1)$$

The network uses a discrete time update scheme, i.e. Eq. 1 is calculated once for each network cycle. The network can be configured as a recurrent network by feeding some of its outputs back to the input neurons. In a recurrent network the output at time t depends not only on the actual input, but also on the previous network cycle. If Δt denotes the time needed for one network cycle the output function of one network block can be written as:

$$O(t+\Delta t)_i = \Theta(\sum_j \omega_{ij} I(t)_j + \sum_k \omega'_{ik} O(t)_k) \qquad (2)$$

As illustrated in Figure 1, the feedback can be used to configure the network as a multi-layer Perceptron by setting the appropriate weights to zero. In this setup two network cycles are needed to propagate a signal from the input layer to the output layer.

Multiple network blocks can be interconnected. Only digital signals have to be exchanged. Since every network block works at the same frequency $f_{net} = 1/\Delta t$ the blocks can be synchronized to a common network clock. If the digital data has to travel a longer distance, it is possible to insert clocked buffers that are synchronized to the network operation to keep the whole chip running at f_{net}. A buffer increases the feedback delay by

0-7803-7278-6/02/$10.00 ©2002 IEEE

Δt. Each connection of l outputs, originating at block m (O_l^m) and having n clocked buffers in the data path adds one term to Eq. 2:

$$O(t+\Delta t)_i = \Theta(\sum_j \omega_{ij} I(t)_j + \sum_k \omega'_{ik} O(t)_k + \sum_l \omega''_{il} O(t-n\Delta t)_l^m \ldots) \quad (3)$$

For the remaining inter-block connections the according terms must be inserted. Using this technique the mixed-mode organization of the network allows large networks without speed penalties. The analog operation is confined to the individual network blocks. Their size can be selected according to the used circuit technology and the desired speed and analog precision. If the power consumption of the analog blocks is low enough the blocks can be combined to networks as large as the semiconductor manufacturer allows. Our group has developed two different neural network circuits that could be used to build megasynapse chips in the described fashion. The first is based on charge sharing, while this paper focuses on the newer version using current summation.

The inter-block routing can be made configurable to allow a programmable topology. By the same connections the network can communicate at full speed with external circuits providing input or training data. Input data is not confined to binary information since multiple single bit inputs can be used together to feed integer values into the network. The coding can be arbitrary and, since there is no fixed number of bits, also optimized individually for each input signal by the training algorithm. The network can also generate binary coded integer output by using the arbitrary feedback depth provided by the recurrent design. Using n network cycles and n output neurons together, a successive approximation analog to digital converter can be implemented by Eq. 2. Thereby the analog sum over the I_j could be converted into the corresponding n-bit integer value.

II. NETWORK IMPLEMENTATION

A. *Basic Considerations*

An analog neural network implementation based on Eq. 1 is only useful if the resources necessary to build a synapse are low. The reduction to single bit input and output neurons implies that a higher number of them is needed to solve the same problems as with multi-valued neurons. Using Eq. 1 has two advantages: first, the synapse can be made simpler, insomuch the speed and area benefits outweigh the functional disadvantages. The second is linked to the *fixed pattern noise*, a phenomenon caused by the inevitable device variations in the manufacturing process of integrated circuits. If analog circuits are reduced in size, the deviations of the device parameters from their design values increase. The problem becomes worse if high speed is also a design target. Circuits built from large devices have higher capacitances than those built from small ones, hence, higher currents are needed to charge them in the same time. This reduces the maximum number of copies of this circuit that can be put on a chip if the power budget is fixed. A neural network needs both: high speed and a large number of synapses. Therefore the synapse circuit has to be small, fast and limited to a current consumption in the order of magnitude of 10^{-6} Amperes to keep the total power consumption of a megasynapse chip in a reasonable range. The binary synapse does not need a multiplier, depending on its input it just adds its weight value or not. Any error caused by device variations changes only the effective value added. This can be fully compensated by shifting the stored weight by the same amount in the opposite direction. Any chip-in-the-loop training algorithm will do this automatically without a performance degradation. This is not possible for a multiplier-based synapse that has not a single operating point, but a transfer function that may be deformed in a complex way due to device variations.

The weight storage is the second difficult part of an analog neural network. The following boundary conditions should be fulfilled:

- small size
- sufficient precision (> 4 bit)
- low power consumption
- fast update rate

The last item forbids the use of an EEPROM-based storage, which is often utilized in analog ANNs due to its otherwise superior properties [6]. The time needed to reprogram EEPROM cells (usually more than 1 ms) precludes their use if some kind of chip-in-the-loop training is intended. Therefore, the memory must either be based on a digital storage – dynamic or static – combined with some kind of digital to analog conversion or analog storage, i.e. as charge. We have chosen the latter since this allows for the smallest synapse size. Furthermore the precision of a simple charge storage is limited by the kTC-noise:

$$\overline{u_c} = \sqrt{\frac{kT}{C}} \quad (4)$$

For a 60 fF capacitor (as used in the presented chip) at a temperature of 300 K the voltage noise is 0.26 mV. Related to the 1.5 V swing used, this leads to a weight error of 0.17 % which is usually more than sufficient. The disadvantage of a capacitor is the leakage current

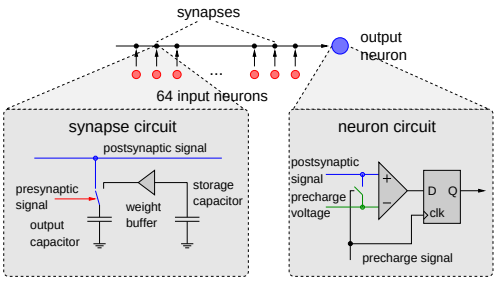

Fig. 2. Operation principle of the first neural network chip based on charge-sharing.

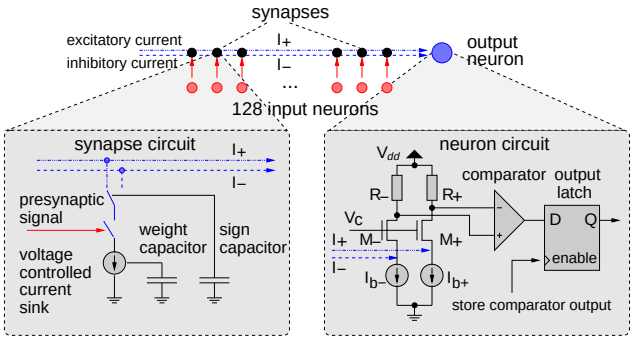

Fig. 3. Operation principle of the new neural network chip presented in this paper.

of the transistor controlling the charge flow. A backup of the weights must be stored elsewhere and periodically transferred on the synapse weight capacitors. Since this refresh must happen every 10 to 100 ms it has no impact on the network operational speed. In the training phase the weights must be updated much more often anyway. Even for a megasynapse chip the external storage of the weight values occupies about 10 to 12 Megabits. Far less than the capacity of the smallest available SDRAM chip from current production.

B. *The Charge-Sharing Neuron*

Figure 2 shows the operational principle of our first neural network prototype [7]. It uses a switched two-phase design with switch frequencies up to 100 MHz. In the precharge phase the synapse weight is copied on the output capacitor. The activated synapses share the charge on their output capacitors in the evaluate phase. The neuron fires if the resulting voltage is higher than the precharge voltage. This chip, which has a total number of 4096 synapses, has proven the feasibility of mixed-mode neural networks, but had some drawbacks that have been removed with the presented new network chip. The differences are listed in Table I.

C. *The New Current-Sum Neuron*

The neuron operation is based on the addition of currents sunk by the active synapses. Figure 3 shows the basic components of the new network block. The input signal of a neuron is generated by 128 synapses. Two lines connect the synapses to the neuron. One is conducting the excitatory and the other the inhibitory current. Each synapse connects either to the exciting (I_+) or the inhibiting (I_-) line, depending on the voltage at the sign capacitor. A voltage controlled current sink determines the weight of the synaptic connection. Its control voltage is stored on the weight capacitor.

If the input neuron i fires ($N_i = 1$), the current sink is connected to either the I_+ or the I_- line. If the sign switch is at the left side ($S_{+i} = 1$), the current from the I_+ line flows through the synapse i. The currents sunk by the synapses produce voltage drops $\Delta V_\pm$ across the resistors R_+ and R_- in the neuron:

$$\Delta V_\pm = R_\pm \sum_{i=1}^{128} S_{\pm i} N_i I_i \qquad (5)$$

A comparator amplifies the voltage difference $\Delta V_+ - \Delta V_-$ to a logic zero or one which is stored in the output latch after the comparison has finished. A logic one in the output latch means that the neuron has fired. The two transistors $M_\pm$ in the current paths act as cascodes to isolate the comparator inputs from the large capacitance of the I_+ and I_- lines, therefore speeding up the network operation. The two current sinks $I_{b\pm}$ force a small current through $M_\pm$ even if no synapse is active, thereby keeping them in saturation.

D. *Architecture of the Presented Network Chip*

Figure 4 shows a micro photograph of the neural network chip. It is partitioned in four network blocks containing 128 synapses and 64 neurons each. The 32768 synapses take up 3.4 mm^2 of silicon area, the whole chip measures about 12.3 mm^2 (3 mm×4.1 mm). The second largest portion of the core area is used by the digital to analog converters (DAC) and the analog weight storage circuits. Two opposing network blocks share one of these DAC units. Together they constitute one half of the network.

The chip communicates with external hardware by a fast digital bus to load the synaptic weight values and the input data into the network and to read back the neuron output data. It uses bidirectional low-voltage differential signalling (LVDS [8]) to achieve data rates of 600 Megabit/s on each differential pair without disturbing the analog network operation. The usage of LVDS to-

TABLE I
ENHANCEMENTS MADE IN THE PRESENTED NETWORK CHIP

	first chip	presented chip	advantage
method of weight summation	charge sharing	current sum	increased precision
sign of weight	voltage level	different signal lines	no zero reference needed
weight loading input	analog	digital (integrated DACs)	digital IO only, faster
synapse power consumption	fixed per synapse	proportional to weight	lower total power consumption

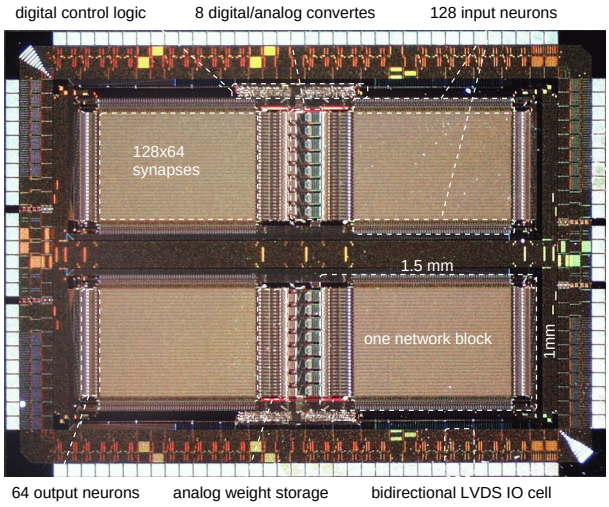

Fig. 4. Micro photograph of the presented neural network chip.

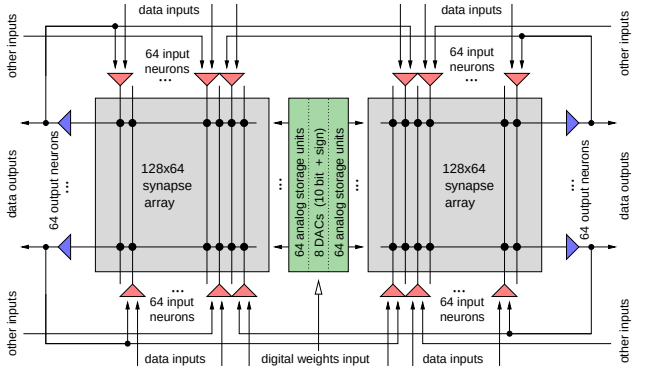

Fig. 5. Interconnections between two opposing network blocks sharing a DAC unit.

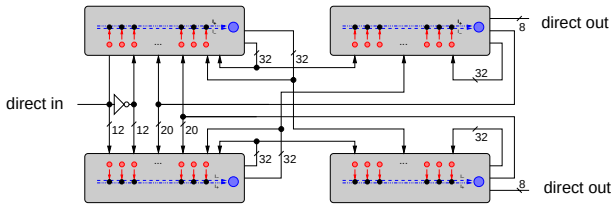

Fig. 6. Data flow between the four network blocks.

gether with a double-data rate source-synchronous clocking scheme allows to connect multiple network chips together on a common bus. The bus is organized in 16 data and 3 address lines. Its total transfer rate adds up to 11.4 Gigabit/s.

Figure 5 shows the interconnection between two network blocks that share a DAC unit. The input neurons at the top and the bottom feed the signals into the network block. Each input neuron has two inputs. A configuration bit stored in each input neuron selects one of them. The inputs named *data inputs* can be directly set by the external data bus of the chip, the feedback inputs are connected to the output neurons. With each network cycle the selected input signal is amplified and sent to the column of synapses connected to the input neuron. There is always a direct feedback connection between the output neurons and the input neurons of one block. But there are also connections from the opposite block as well as from the other half of the chip (named *other inputs* in Figure 5). Each neuron data output can also be read back by the external data bus.

Figure 6 shows the feedback connections between the four network blocks. The routing of the neuron output signals is biased towards two layered network structures with data flowing from left to right. This is also reflected by the *direct in* and *direct out* signals that connect directly to standard CMOS IO pads of the chip. They allow the direct connection of external data sources and sinks to the network. If the chip is used in data communication applications for example, an analog front/back end could be connected there. The *direct input* data bits are connected to the input neurons twice: inverted and non-inverted. Thereby binary coded signals could be fed into the network in a way that for each code value an equal number of synapses is activated.

The DAC unit consists of 8 DACs with 10 bit resolution. The conversion time of each DAC is 40 ns. Between the DACs and the synapses there are 64 analog weight storage units that store the analog output currents of the DACs. Since each DAC has two banks of digital input latches capable of the full interface speed, the time to load the digital data can be totally hidden

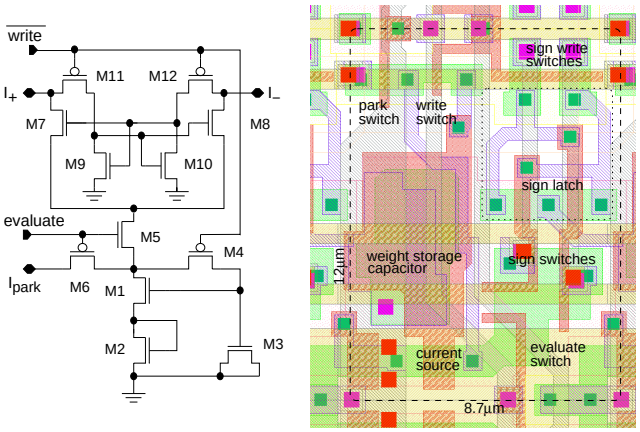

Fig. 7. **Left:** Circuit diagram of the synapse, **right:** layout drawing.

in the conversion time. After 8 conversion cycles all the 64 analog weight storage units have their weight values stored. While they are transferred into one selected row of synapses the DAC unit is used to load the weight storage units of the opposite network block. The digital sign bits are transferred together with the weight values into the storage units. They get programmed into the synapses in parallel with the weight currents. The weight data is transmitted to the DACs in the top and bottom half of the network chip simultaneously.

By using the scheme described above a total weight storage rate of 400 Megaweights/s is achieved. All the weights of the chip can therefore be refreshed in 82 μs. By using a 10 ms refresh rate, the network speed is reduced by less than 1 %.

E. *Synapse Operation*

The circuit diagram of the synapse is depicted in Figure 7. The main parts are the current sink (transistors M1 to M6) and the sign storage (M7 to M12). Transistors M1 to M4 form the current memory cell. M1 acts as a current sink, controlled by the voltage on M3, which is connected as a capacitor. Since its capacitance decreases if its gate voltage falls below its threshold voltage V_t, the source voltage of M1 is raised by the voltage drop over the diode-connected NMOS transistor M2, which has the same W/L ratio as M1. M2 keeps the gate voltage of M3 always above V_t and also increases the output resistance of the current sink.

The dependency between the gate voltage of M1 and its channel current is not linear and also differs between each synapse due to the transistor parameter variations. Therefore the synapse weight is programmed by a current instead of a voltage. The current is stored by activating the $\overline{write}$ signal and keeping the *evaluate* signal low. Thereby M4 and M6 are conducting while M5 isolates the current memory from M7 and M8. M1 will sink any current forced into it through the I_{park} input by the analog storage unit. After deactivating the $\overline{write}$ signal the gate voltage necessary to sink the desired weight current by M1 and M2 is kept on M3. The voltage error caused by the charge injection of M4 can be compensated by a slight shift in the weight value.

The sign of the synapse is determined by the state of M9 and M10. While the $\overline{write}$ signal is active, either the I_+ or I_- line is set to V_{dd} by the analog storage unit. This causes charge flowing on the gates of either M9 or M10, letting it conduct and thereby discharge the gate of its crosscoupled counterpart. The charge injection caused by M11 and M12 in the moment $\overline{write}$ becomes inactive (switches to V_{dd}) increases the voltage on the activated side even further while the other side is held low by the conducting transistor. This ensures that the selected output transistor, either M7 or M8, is switched on. It is also possible to disable a synapse completely by setting both I_+ and I_- to ground.

During normal network operation the input neuron activates the synapse with the *evaluate* signal. This connects the current sink to the I_+ or I_- line, depending on the state of M9 and M10, i.e. the programmed sign. To keep the current sink in its correct operational region and avoid voltage drifts at the drain terminal of M1 the current sink is connected to the I_{park} line once it is deactivated by the input neuron. I_{park} is connected to V_{dd} via a transistor acting as a cascode set to the same bias voltage V_c as I_+ and I_- in the neuron (see Figure 3).

The right half of Figure 7 shows the layout drawing of a synapse embedded in the synapse array. Every second row is mirrored. This allows adjacent rows to share their n- respectively p-well with each other. The mismatch introduced by this mirroring can be calibrated by the weight value. One synapse occupies only 104.4 μm^2. Table II summarizes the features of the network chip.

TABLE II
CHIP PERFORMANCE SUMMARY.

process features	0.35 μm, 1 poly, 3 metal
die/core size	4.1 × 3 mm^2/3.6 × 2.5 mm^2
blocks/neurons/synapses	4/256/32768
supply voltage	3.3 V
network frequency f_{net}	50 Mhz typ.
connections/s	1.64 Teracps
weight update rate	400 Megaweigths/s max.
weight resolution	10 bit (nominal) + sign
bus data transfer rate	11.4 Gigabit/s max.

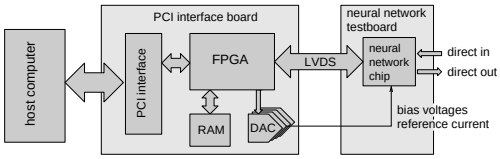

Fig. 8. Testbench used for the evaluation of the chip.

III. EXPERIMENTAL SETUP AND FIRST RESULTS

The network chip is connected to a digital control unit that generates the synaptic weights and initializes the internal configuration registers. If the data transmission capacity of the direct IOs is too low or their connection to the network block not adequate for the application, the digital control unit can also route any input and output data across the external bus to the network blocks. Chip-in-the-loop training algorithms can use the external bus for their training and result data transfers.

Figure 8 shows the setup used for the experiments with the neural network chip. Since the training algorithms are implemented in C++, the network is connected via a PCI-card to a standard PC. The PCI-card carries a Virtex-E FPGA[1] that acts as the digital controller for the network chip. It is also capable of directly interfacing to the LVDS-signals. To avoid unnecessary data transfers across the PCI-bus the FPGA has its own memory. The DACs generate the bias voltages and the reference currents needed by the network chip. If the training algorithm was implemented completely in the FPGA, no software interaction would be necessary.

As the training system is in an early phase we have only been able to run a few tests to verify that the chip is working properly. A genetic algorithm was successfully used to train the network for the 4 and 5 bit parity problem. Two network cycles were performed by one network block. A similar configuration has been reported as a software solution for the 4 bit parity problem in [9]. A more detailed description of the algorithm used can be found in [7]. Figure 9 shows the results.

IV. CONCLUSION AND OUTLOOK

We have introduced a mixed-mode neural network architecture that allows ANNs to be realized in VLSI microchips that combine high performance with low area requirements. With a synapse size of about 100 μm^2 more than a million synapses could be integrated with today's process technologies. The small synapse size is made possible by using digital input and output neurons

[1]The Virtex-E FPGA is manufactured by Xilinx Inc., San Jose, CA, USA

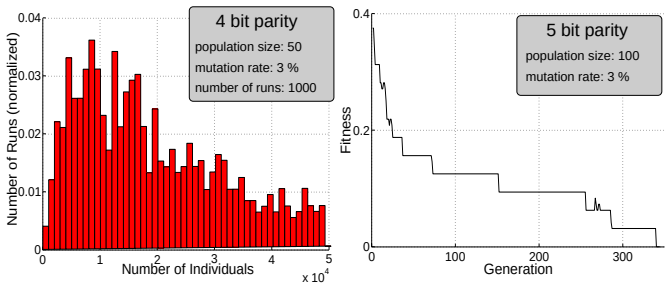

Fig. 9. **Left:** Histogram of the number of individuals needed to solve the 4 bit parity problem. **Right:** Example fitness curve of a 5 bit parity experiment.

combined with an analog weight summation. We have presented an implementation of this architecture having 33k synapses. The neuron operation is based on the comparison of excitatory and inhibitory current sums. Due to the integrated DACs for the weight input any host communication uses only digital signals. First results of this chip show that it is possible to train the 4 and 5 bit parity problem with a genetic algorithm using a chip-in-the-loop configuration. In the future we plan to train the chip for time-domain problems, like data communication applications, filtering, etc. We also intend to develop hierarchical training methods to make the best use out of the large possible number of synapses.

References

[1] Moerland, P., Fiesler, E.: "Neural Network Adaptions to Hardware Implementations," Handbook of Neural Computation, E1.2:1-13, Institute of Physics Publishing and Oxford University Publishing, New York, 1997

[2] Lindsey, C., Lindblad, T.: "Survey of Neural Network Hardware," SPIE Vol. 2492, pp. 1194-1205, 1995

[3] Lazzizzera, I., and Lee, P., and Sartori, A. and Tecchiolli G. and Zorat A.: "Advances in the design of the TOTEM neurochip," *Nuclear instruments & methods*, vol: 389, 1997

[4] Lee, C., DeVries, D.: "Initial Results on the Performance and Cost of Vector Microprocessors," International Symposium on Microarchitecture, Proceedings, pp. 171-182, 1997

[5] Kramer, A.: "Array-Based Analog Computation," *IEEE Micro*, pp. 20–29, October 1996

[6] Shibata, T., Kosaka, H., Ishii, H. , Ohmi, T.: "A Neuron-MOS Neural Network Using Self-Learning-Compatible Synapses Circuits," *IEEE Journal of Solid-State Circuits*, Vol. 30, No. 8, pp. 913-922, August 1995

[7] Schemmel, J., Meier, K., Schürmann, F.: "A VLSI Implementation of an Analog Neural Network suited for Genetic Algorithms," ICES 2001, Proceedings, Springer, ISBN 3-540-42671-X, pp. 50-61, 2001

[8] ANSI/TIA/EIA-644 1995 Telecommunications Industry Association, "Electrical Characteristics of Low Voltage Differential Signaling (LVDS)," March 1996

[9] Pujol, J., Poli, R.: "Evolving Neural Networks Using a Dual Representation with a Combined Crossover Operator, " Proceedings of the IEEE International Conference on Evolutionary Computation (ICEC), pp. 416-421, 1998