HIGH PERFORMANCE SINGLE NEURO-CHIP WITH ON-CHIP-LEARNABILITY

Frank Stüpmann, Silicann Technologies GmbH, Rostock, Joachim-Jungius-Straße 9, 18059 Rostock, Germany
Ansgar Wego, University of Rostock, Institute GS, Dep. Electrical Engineering, Einsteinstr. 2, 18059 Rostock, Germany

ABSTRACT

It will be shown the newest results of a hardware realization of a neural net for fast decision making functions in real time. There is a digital micro core with several functions – proceeding of the learning and testing of the net, supervising of training process and computation of some calculations in pre- and post-processing. The patterns are automatically presented to the network. The heart of the classifier is a trainable integrated analog neural network structure. Because of its speed the hardware realization is able to solve real time image recognition problems The number of neurons integrated in the whole chip is 100 in the input layer, 60 in the hidden layer and 10 in the output layer. The back propagation algorithm is implemented in an analog circuit.

INTRODUCTION

The chip is meant to be used for making decision functions in real time. [1], [7] deal with the examination of existing hardware realization of neural nets. There are some analog neural net chips [2], [3], [4], [5], [6]. In [1] it was stated that previous solutions contain some disadvantages. Thus the number of the integrated neurons is small and often on-chip learning is not possible. The low complexity, not sufficient for many problems, only permits a restricted number of applications. Therefore the aim of the work was deduced to contribute to the development of an fast, complex neural integrated circuit capable of learning.
The classifier consists of the units switch, classification and control. The switch unit carries out the switching between learning vectors and input vectors requested from the unit classification in correspondence to the learning process or the working process respectively.

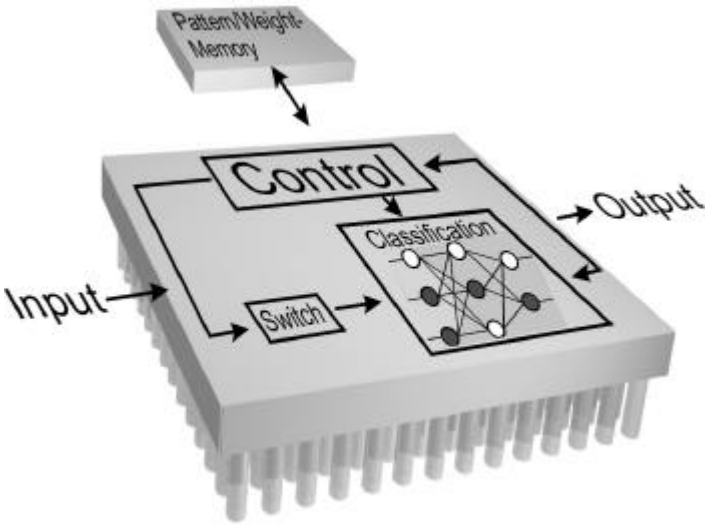

Fig.1: Structure of the whole chip

In Fig.1 the whole neural classifier is shown with its units control, switch and classification. The classifier is realized as a single chip.
It is not necessary to use a second chip for controlling the neural net itself because the fast analog realization of the net and the control function as a digital part are implemented on one chip. The switch unit is realized by an analog switch. The switch unit carries out the switching between learning vectors and input vectors that are requested by the classification unit in correspondence with the learning process or the working process respectively. It is possible to program the chip like a microcontroller but the internal processing speed is determined by a fast analog structure.

The net's topology integrated in the function classification in analog circuitry is the multi-layer perceptron. All operations for the learning and the reproduction phase including the learning in hidden neurons are implemented in analog circuitry. The learning algorithm used is the back propagation algorithm (BPA). The chip uses a SIMD-architecture. The time it takes for the data to propagate from the input to the output in the working process is 2µs. Fortunately, neural systems are more tolerant of low-accuracy components than conventional computation systems. The internal resolution is 6 bit and the resolution at the signal inputs and outputs is 10 bit. The chip has analog input/output buffers. The technology used is the 0,6 µm CMOS-technology CUP from austriamicrosystems.

Multilayer perceptron with Learning Automation

The net's topology integrated in the chip is the multi-layer perceptron. The learning algorithm used is the backpropagation algorithm (BPA), this algorithm is well known with all its advantages and disadvantages. A lot of users will apply this chip and more than fifty percent of current applications of neural networks use the BPA. In the final version it will have 100 input neurons, 60 hidden neurons and 10 output neurons. The activation of the neurons lies in the range of [0,1]. A sigmoid function where the final value is reached asymptotically very fast is used.

$$f(x) = \frac{1}{1+e^{-bx}}$$

with $\beta > 0$.
In the backpropagation algorithm the changing of the weights W of the multi-layer perceptron is realized after propagation of the input pattern $i^{(p)}$ ($p \in L$) by:

$$?_p W(u,v) = ? d_v^{(p)} a_u^{(p)} \text{ with } u \in U_{i-1}, v \in U_i$$

($2 \le i \le n$) and $\eta > 0$, whereas

$$d_v^{(p)} = \begin{cases} f`(net_v^{(p)})(t_v^{(p)} - a_v^{(p)}) \text{ if } v \in U_n \\ f`(net_v^{(p)}) \sum_{\tilde{v}} d_{\tilde{v}}^{(p)} \cdot W(v,\tilde{v}) \text{ if } v \in U_j \; ; \; 2 \le j \le n-1 \end{cases}$$

is.
There is au(p) the activation of the unit u after propagation of the input patterns i(p) and tv(p), v? Un is the default output prescribed from the output pattern t(p) of an output unit un.
The realized net has no bias values, like the implementation of the function classification.

CONTROL UNIT

The *control* unit controls the chip. This unit is subdivided in control functions which have the following tasks during the separate oparating phases:

control functions	Function	phase, in which the unit has a meaning
pattern-control	presentation of the input- and output-patterns	learning phase
weight-control	supervision and control of initialization, update and refresh	learning, test and operating phase
error-control	supervision of the error in the learning- and test phase	learning and test phase
random-unit	random numbers for initialization and pattern presentation	learning and test phase

TABLE I
Tasks of control function

PATTERN CONTROL

The pattern control function controls the reading in of learn patterns into the pattern memory. Afterwards this function reads the patterns from the pattern memory and presents them to input and output of the function classification. On this occasion every pattern is presented separately for a short time. If all patterns from the pattern memory are presented and if the learning was successful, the test patterns are presented to the classificator. The output of the pattern elements and also of the variable learnrate is realized as 10 bit value.

WEIGHT CONTROL

The internal representation of knowledge has to be adapted to the function which has to be solved. The algorithms necessary for this purpose are realized in the function classification. The system stays in the state of learning until the error in all patterns becomes small enough and the adaptation of the classificator is successful.
Weight control controls the initialization of the weights, the state of learning, the learn rate and the outputs for the learning patterns.
Before the learning weight-control forwards the initialization values of the weights to the synapses of the function classification. Furthermore the central refresh of weights and the saving to the pattern/weight memory are realized by weight-control.

ERROR CONTROL

During the learning process the square error is calculated from the digitized value handed over by the output neurons and the expected outputs. If the calculated error is greater than the maximum permissible net error the instruction of the net has to be continued. If the error is less than the desired minimal error the learning is successful and the net has converged. If the converging fails the net has to be reinitialized and to be reinstructed with a smaller learn rate. To control the learning only the learn rate and the number of learning cycles are available. After the learning the test patterns are presented to the classificator. The observed error is summed to a summation error. This function is also solved by the error control. If greater errors appear, the net has also to be reinstructed. It is switched over to learning mode again.

RANDOM UNIT

The random function is one of the most important functions of the controller, because it is both necessary for initialization and for pattern presentation. At the beginning of learning the weights are initialized with low random values so that the values of weights are approximately zero. The derivation of the logistic activation function has a maximum in that region. It has to be guaranteed, that the values of weight are different from zero. During the learning control takes care of presenting each pattern to the net only once and in every cycle in another order. This characteristic of pattern presentation is a basic condition for the success of learning.
The controller generates the random numbers with method of additive congruence. The numbers are serially selected after that [8]. Beside the simple implementation of this algorithm the advantages are the occurrence of any possible combination in each cycle and the fast process because of the absence of complicated and time consuming multiplication and division instructions.

CONCEPT OF ANALOG STORAGE

The analog storage of the synaptic weights is a difficult task, since not all desirable parameters like precision, long time stability, fast adaption ability can be found in one circuit variant. Earlier analog implementations of neural networks used floating gates (e.g. Intel ETANN). Transistors with floating gates can change their threshold voltage and therefore represent the synaptic weight in the zero stages of multipliers. This is a very high circuit area economizing implementation. Weights remain stored up to ten years. The disadvantages are, that floating gates can not be produced in standard technology, programming takes several milliseconds and this is the reason why a weight update of a whole neural net can take up to several minutes, because programming works only sequentially. Also a high programming voltage is necessary. The reproducibility of programming is limited to $10^{4 \cdots} 10^{6}$ programming cycles.

Since high speed on chip learning should be an essential feature of the development of this chip the emphasis is put on capacitive storage in development. Capacities can relatively fast and arbitrarily often be reprogrammed (ns ... µs range). Furthermore they can be easily produced in standard technology.

However leak currents are a large disadvantage which must be overcome by suitable circuit technology. Integrated capacities are already unloaded after some milliseconds by leak currents. Not only the leak current of the capacity but also the reverse current of the source bulk junction of the pass transistor must be considered as well as the drain source leak current. The latter occurs at high drain source voltages and causes the touch of the space charge zones (punchthrough effect). Therefore a refresh must be implemented. This, however, means that the weight signal must become discrete and the precision of the weight is limited.

The punchthrough effect can be avoided by the reduction of the source drain voltage of the pass transistor. Figure 2 shows a circuit [9] for the reduction of the leak currents, called cunit. The stored voltage will loop back through an operational amplifier and a transistor M3 to the input of the pass transistor M2 so that the voltage drop over it is 0V and the leak currents are fundamentally reduced.

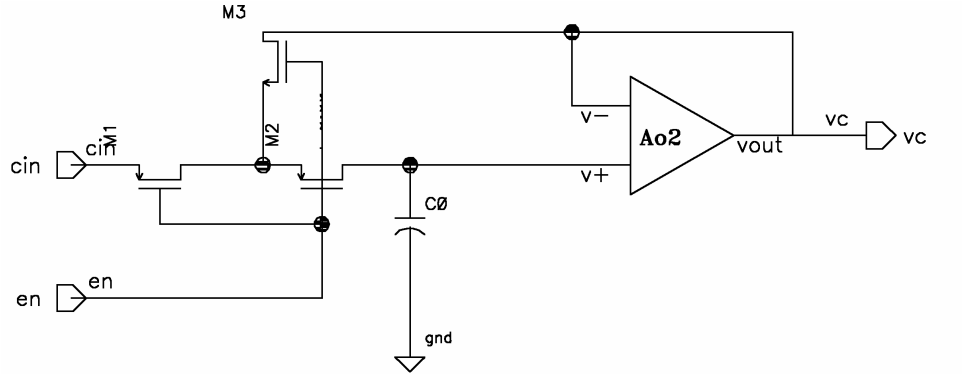

Fig. 2: storage times from a few seconds up to one minute are obtainable with the cunit using a capacity of only 1pF

The feedback transistor M3 switches inversely to the pass transistor M2 so that during loading there is no feedback. Another transistor (M1) is at the input for decoupling the input cin from output vc. An advantage of this circuit is the load independence of the output.

LEARNING WITH CAPACITIVE STORAGE CELLS

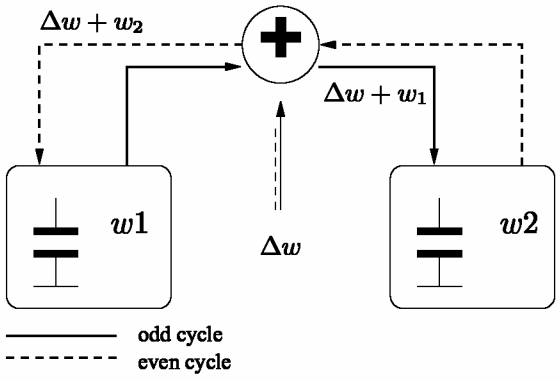

Fig. 3: Principle of the update of the weights in the weight processing unit (WPU)

How does learning with capacitive storage cells work? The BPA works in a way that at the beginning of learning all weights are initialized with coincidental values near zero. The weights are then changed so that the output vectors approach the desired training vectors. The necessary weight modification ?w are calculated by the BPA . This value must be added to the current weight:

$$w_{new} = w_{old} + \Delta w$$

Since w_{old} and ? w are in the form of voltages, w can not easily be added to in a capacity. Therefore the principle from figure 3 is used.
Two capacities are needed of which only one is active at one moment. Assuming w_1 is initialized, the first learning cycle starts by adding calculated weight modification ? w to w_1 (= w_{old}) and being saved as w_2 (= w_{new}). In the next learning cycle $w_2 = w_{old}$ *and* $w_1 = w_{new}$ etc. The storage units cunit are in the weight processing unit (WPU). Every single synapse contains a WPU. The update process is executed for all synapses at the same time. By this massively parallel mode of operation learning becomes very fast since the learning process does not have to be executed by a processor working sequentially.

RESULTS

For the realization of the neural structures a test chip was designed which contains single components as well as complex circuit blocks. WPUs, subtractors, operational amplifiers, gilbert multiplier, cunits (see figure 2) single synapses and neurons as well as a small neural network belong to this. The results of the cunit shall be represented here.

The circuit cunit was created and simulated with Cadence. Already in the simulations has been recognized that at a resolution of 6 bit (6 bit corresponds to a quantization level of 31.25mV) storage times in a range of seconds to minutes can be achieved. Offset of the operational amplifiers as well as geometrical dimensions of the pass transistor M2 are parameters influencing this characteristic quantity. Figure 4 shows the drift of the stored voltage over the period of time of more than one hour starting at different start points. These are already real measurement results and no simulations. For the worst case the drift is approx. 0.5 mV/s. The drift in the measurement result is positive, however, what does not mandatorily always have to be that way.

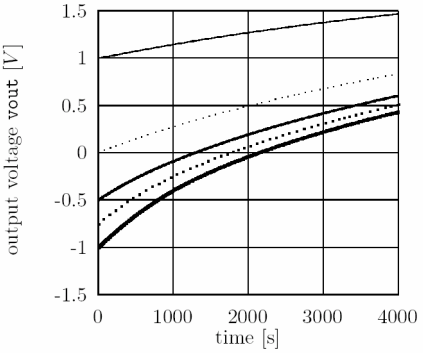

Fig. 4: measuring of the drift of one cunit (capacity 1pF)
over a periodof time of more than one hour

However not only the drift but also the voltage level difference when closing the cunit is decisive for the storage duration. This excursion results from the voltage edge (5V) at the *en*-pin and the quotient of capacities $C_{Gate\ M2}/C_0$. This voltage level excursion is desired and is +8mV on average. By this excursion the stored voltage is raised and centered more or less between two quantization levels to gain time for a refresh also at a negative drift. Figure 5 shows a zoom section for the drift of avoltage started at 1V. The switching point of *en* lies at approx. 300ms. It takes approx. 50s on average for the reaching of the next quantization level (0.96875mV). Unfortunately, a large scattering of the offset of the operational amplifiers occurred by the scattering of the transistor parameters. Normally all curves should start with -1V and then at the LH edge of *en* jump for +8mV. However the margin up to the quantization levels is strongly limited by the statistical distribution of the offset. In spite of this it is positiv that the offset did not affect the drift as strongly as assumed.

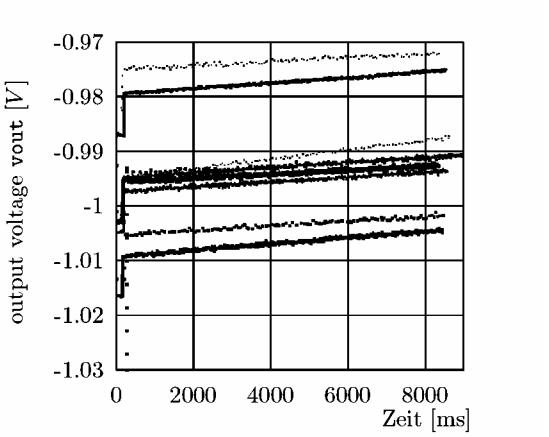

Fig.5: measuring of the drift of ten cunits (capacity 1pF) over the time of 8 seconds

CONCLUSION

The evaluation of the storage behavior of the cunit has shown that the capacitive storage method even with small capacities of only 1pF is suitable as an analog storage for the synaptic weights of a neural network. Due to this storage duration, that is for a resolution of 6 bit in a range of seconds, 10.000 synapses can be easily refreshed by a central refresh unit even if the refresh of a single synaptic weight takes up to 500µs.

The problem of the offset in the operational amplifier of the cunit occurring till now does not only impair the storage duration but also affects harmfully the BPA. The algorithm cannot converge for very small weight modifications since an offset is added to weight modification ? *w* at every learning cycle. This can be avoided by using more narrowly offset tolerated operational amplifiers. On the one hand this is to achieve by larger circuit area and on the other hand by better layout methods such as folding of zero stages.

Since the drift of the storage voltage despite high offsets is very little the prospects of future developments are promising.

[1] Lindsey, C.S.: Neural networks in hardware: architectures, products and applications, *www.practicle.kth.se/lindsey*, 1998. Lecture at Royal Institute of Techn. Stockholm, Sweden.

[2] INTEL: 80170NW electrically trainable analog neural network. *INTEL Information Sheet E358*, INTEL Corporation, 2200 Mission College Boulevard, Santa Clara, USA, 1990

[3] INTEL: 80170NX Neural network technology and applications. *Technical report INTEL Corporation*, 2200 Mission College Boulevard, Santa Clara, USA, 1992

[4] Ramacher, U. and Rückert, U.: *VLSI design of neural networks*; Kluwer, Boston, USA, 1991

[5] Masa, P. and Hoen, K. and Wallinga, H.: A high-speed analog neural processor; *IEEE Micro-Journal*; vol. 14, pages 40-50; 1994

[6] Hammerstrom, D.: A VLSI architecture for high performance, low-cost, on-chip-learning; *IEEE-Journal*; vol. II, pages 537-544; 1990;

[7] Graf, H.P. and Sackinger, E. and Jackel, L.D.: Recent developments of electronic neural nets in north America; *Journal of VLSI Signal Processing*; vol. 5; pages 19-31; 1993

[8] Pfenniger, E.: Erzeugen von Zufallszahlen mittels Schieberegister, *www.ing.pfenniger.ch/zufall.html*, 1998.

[9] M. Kruse: Entwurf einer Synapse für eine selbstlernendes neuronales Netz als analoge VLSI Schaltung, Universität Rostock, Diplomarbeit, 1997