VLSI Implementation of an Artificial Neural Matrix with Analog Nonlinear Synapses

MOMCHIL MIHAYLOV MILEV
Faculty of Electronic Engineering and Technology[1]
Technical University of Sofia
8 St. Kliment Ohridsky Blvd, Sofia
BULGARIA

Abstract: A simple five-transistor analog-signal synapse circuit is designed capable of high processing speed, low power consumption and VLSI implementation on a standard CMOS process. A neural matrix, "NEURO-MATRIX-1", of more than 2048 synapses is implemented on Taiwan Semiconductor Corporation's (TSMC) 0.35µm, mixed-signal, 3.3V, standard CMOS fabrication process to demonstrate feasibility of using the proposed nonlinear synapse circuit in a practical application. The neural-matrix is the essential part of a system-on-a-chip (SoC) to be used for feature-extraction in higher-level, hybrid (hardware and software) finger-print image recognition system. Synapse circuit diagrams and simulation results are shown. First silicon test-results are discussed. System-level and data-flow diagrams of the SoC are presented. Design performance considerations, prototype solutions, and performance results are discussed in subsequent sections. Design implementation notes conclude our paper.

Key-words: Artificial neural networks (ANNs), Analog implementations, Very Large Scale of Integration (VLSI), Nonlinear synapse, Synapses with nonlinearity.

1 Introduction

Simple analog synapse model using a single MOSFET to compute synapse multiplication (internal activity) function was proposed [1][2]. In subsequent chapters we develop and design an artificial neuron circuit based on that model. To study the feasibility for real-life applications, we design and build a Neural Matrix implemented in a VLSI circuit to be used in a system for finger-print feature extraction.

In order to be successfully used in a real-life VLSI circuit, accounting for various parasitic effects and signal noise requirements, we devise and pursue the following goals for the design of the artificial neural synapse and synapse matrix itself:

(1) Design a small-silicon-area analog synapse and minimize the total chip area yet allow for a large number of synaptic connections.

(2) Use analog-current signal representation for data propagation from one neuron to another and thus minimize parasitic capacitive effects due to highly-interconnected neuron's input/output nodes

(3) Use dynamic analog RAM-like memory on capacitors to store temporary weight data to avoid multiple SRAM data read out operations

(4) Provide easy-expansion capability

We believe we achieved these goals in a small, five transistor synapse circuit which interconnects with one or more of the same to form an artificial neuron.

2 Synapse design

2.1 Synapse circuit design

In order to use the drain current of a MOSFET as a post-synaptic activity signal an independent variation of

[1] Momchil Milev is currently employed by Texas Instruments Inc, Tucson, USA. The information contained in this paper is not subject to intellectual property, trademark or nondisclosure agreements, should not be affiliated with Texas Instruments Inc nor it does describe a product by TI or its subsidiaries.

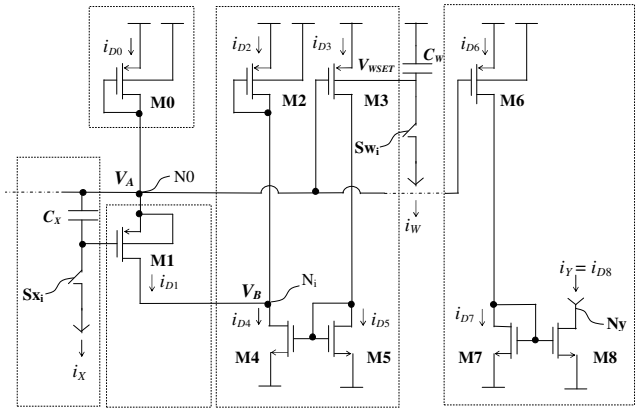

Figure 1 Analog current synapse; synapse current input; weight-control and neuron output circuit schematic

gate-source and drain-source voltages is required. We devise a circuit, which utilizes a current-feedback technique to allow drain-source voltage to be set and maintained independently from the input gate-source voltage. The circuit can have one or more current inputs and a single current output. The current through each input is converted into voltage difference on an input capacitor C_X and then applied on gate-source MOSFET terminals.

The neuron circuit is comprised of: a current summing node N0, common for all synapses; a synapse MOSFET M1; active load M0, common for the neuron; current sensor M3; summing-node voltage drop compensation circuit built on M2, M4 and M5; weight control switch S_{Wi} and capacitor C_W; input current switch S_{Xi} and input current conversion capacitor C_X; and a neuron output N_Y current sink on M6, M7 and M8. In practice, the input-current switch S_{Xi}, for all but the first layer of neurons, is embedded in the output stage of the previous-layer of neurons.

First, the synapse weight is set by applying a constant reference current pulse on C_W through S_{Wi}, for a variable time period T_W, which is generated by synchronous down-counters pre-loaded with a 9-bit weight value. Then, the pre-synaptic input current signal is sampled and integrated over a fixed[2] period of time, T_X on the poly-silicon, thin-oxide, n-well capacitor C_X. The input current remains constant for T_X. After the switch S_{Xi} is opened, summation of all synapse currents at node N0 begins, and the output current, i_Y, now represents the sum of all synaptic currents through the active load M0. Devices M0, M6, M2 and M3 are all-parameters-matched devices. Ratio of M4 and M5 is chosen[3] such that:

$$i_{D2} \approx i_{D0} \tag{1}$$

Any difference in i_{D2} from i_{D0} creates voltage difference between V_A and V_B thus determining the synapse transistor weight voltage:

$$\upsilon_W = \upsilon_{DS1} = V_A - V_B \tag{2}$$

$$V_A - V_B \cong V_{DD} - \sqrt{\frac{2i_{D0}}{\beta_0}} - V_{T0} - V_{DD} - \sqrt{\frac{2i_{D2}}{\beta_2}} + V_{T2} \tag{3}$$

Since M2 and M0 are all-parameter matched pair:

$$\upsilon_W \cong \sqrt{\frac{2}{\beta_0}}\left(\sqrt{i_{D2}} - \sqrt{i_{D0}}\right) \tag{4}$$

The difference between the drain current of M2 and M0 is established by offsetting the threshold voltage of the current sensor M3 through a small weight-setting voltage stored on C_W This voltage ranges from 0 – 400mV and allows for a 9-bit weight value representation. V_W is much smaller and varies from 0 to 100mV maximum.

Current synapse M1 operates in non-saturated mode (V_X ranges from 1000mV to 2024mV) at all times, while all other devices are normally in saturation.

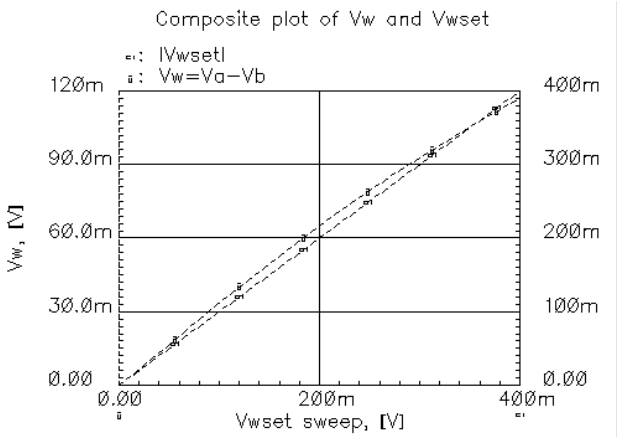

Figure 2 Composite plot of weight voltage V_W and weight-setting voltage V_{WSET}

[2] Input current conversion time, T_X , is programmable for each NN layer in the range of 1 to 32xT_{CLK} (5-160ns) to provide input signal scaling capabilities

2.1.1 Relationship between weight-setting voltage on C_W and weight-voltage

The derivation of the differential equation that yields the relationship between the weight-setting voltage on C_W and the actual weight-voltage across the terminals of the synaptic transistor can be found in [5]. Although the resultant relationship is nonlinear it can be approximated with linear. We believe it will be best described by a graphic plot, shown in Figure 2.

2.1.2 Current summing node voltage drop compensation

The overall synaptic current collected in node N0 creates a voltage drop on the active load M0. To compensate for this drop and to maintain the weight voltage across synaptic transistor channel independent of this change we use a current feedback on M3, M5 and M4. We do this by mirroring the overall synaptic current with the current through the matched active load M2. Each such current mirror of every synapse has a ratio of k. This ratio is determined by:

$$k = \frac{1}{N} + 1 \qquad (5)$$

where N is the number of synapses attached to node N0. This ratio is 2.0 for N = 1 and approaches 1.0 with increasing number of synapses. For N = 17 (our case), this ratio is 1.059. In this way, for each synapse, equality (1) can be maintained. Any change in any individual synaptic current, now, would create the same amount of change in the overall synaptic current, and then through the current feedback all synapses compensate the voltage drop of node N0 by lowering their respective V_B node voltages.

2.1.3 Output current hard-limitation

There are two factors, beside many others[4], which contribute to a natural hard-limiting function to be inherently implemented by the circuit. These are – first, power and ground limitation, and, second – MOSFET devices leaving their normal mode of operation (from saturation to non-saturation). Upper current limit: due to the current-feedback compensation circuit, any increase of the total synaptic current causes voltages V_B to drop with the same amount as the common node voltage V_A drops. At the same time gate voltage of M5 increases

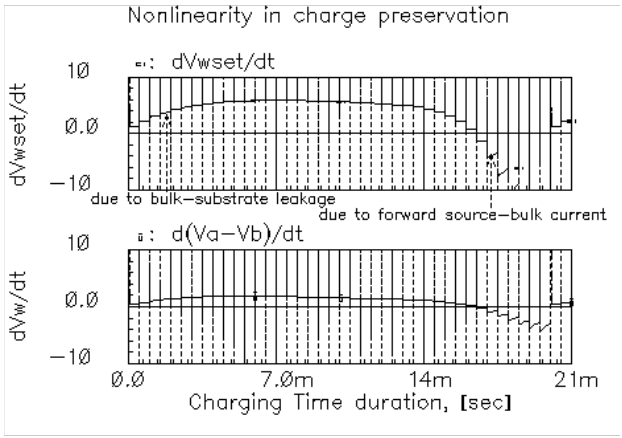

Figure 3 Curves of weight and weight-capacitor voltage derivatives with respect to charge preservation duration (no 'compensation')

due to increase of its drain current. At certain point, M4 leaves non-saturated mode due to its drain voltage dropping below V_{DSAT}. Any further increase of its drain current will be linear and not exponential. Any further increase of the overall synaptic current will not be compensated which leads to gradual decrease in the weight voltage across the synaptic transistor. This decrease in the weight voltage, along with V_B voltage dropping to almost ground, creates a hard-limited upper boundary for the increase of the overall synaptic current. Similarly, for very small synaptic currents, the drain current of M5 would be insufficient to generate drain voltage large enough to open M4 ($V_{D4} < V_{T5}$). This means that V_B and V_A nodes are going to be at close to V_{DD} potential and the neuron load M0 is going to be in under-threshold region, which will create almost zero lower boundary for the overall synaptic current.

Also, the weight-setting voltage on C_W, hence V_W, is limited, naturally, by the forward-bias source-bulk current of M3. Synapse input voltage, V_X , is also ultimately bound by the neuron's power supply ($3V$) and current sink transistor (input switch S_{Xi}) shut-off voltage due to drain-source voltage approaching zero.

2.2 Weight range and resolution

Synaptic weights are first read-in and stored as 9-bit numbers in the 19K SRAM array. 9-bit weight words pre-load 9-bit down-counters for each synaptic weight, which determine the duration (0 to $512T_{CLK}$ i.e. 0 to 2.56μs) of the fixed-amplitude[5] current pulses that charge each synapse's weight. The latter pulses charge

[3] see section 2.1.2

[4] Weight-setting circuitry includes static registers and down-counters limited to 9-bit by design. Similarly, input DAC limits input vectors to 9-bit (in digital-input mode).

[5] The weight-setting current is fixed at about $80\,nA$ by a chip-level reference source. This reference current can be set/adjusted externally to the chip by outside analog current source reference if needed.

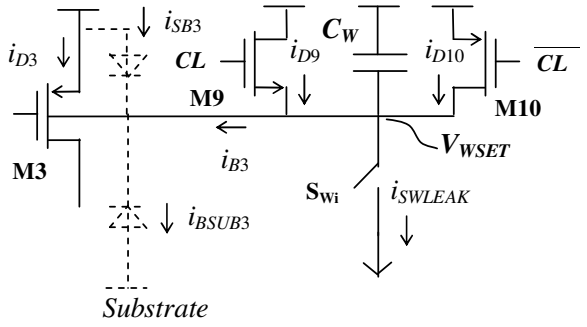

Figure 5 Weight leakage current compensation – drain currents of (51) and (52) balance off open-switch and bulk leakage of (43) which would otherwise sink current through C_W

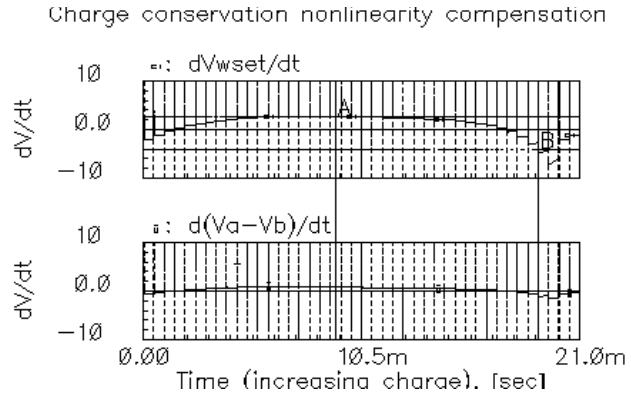

Figure 4 Curves of weight and weight-capacitor voltage derivatives with respect to charge preservation duration - with leakage current error "compensation".

each synaptic weight capacitor to a voltage, V_{WSET}. Weight-setting voltage V_{WSET}, stored on each weight capacitor, offsets the threshold voltage of the current-sensing FET of each synapse, and subsequently offsets the currents through M2-M0 pair of matching active load transistors. This offset creates weight voltage $V_W=V_A-V_B$. Curves of weight capacitor voltage, V_{WSET}, and weight voltage, V_W, derivatives with respect to charge or preservation duration time, are shown on Figure 3. Ideally, these quantities have to be zero during charge retention (no change in stored charge) and should be constant with respect to the amount of charge stored (independent of charge or conservation duration). In practice, these quantities are not zero during charge conservation, and not constant for different weight-setting voltages. In charge conservation mode, the curves show worst nonlinearity for very small and very large amounts of charge stored. The first is, most likely determined by increased leakage through the reverse-biased bulk-substrate junction of M3 (i_{BSUB3}) including capacitor oxide and open-switch leakage (i_{CWLEAK}), while the second – is most likely determined by the increased sub-threshold forward-bias current through source-bulk junction of the same device (i_{SB3} on Figure 5). In the middle of the weight-setting voltage range the curves are almost flat (constant). From these curves, the maximum dynamic range is determined such that to allow less than $0.5V_{LSB}$ error– 0 to 400mV for V_{WSET} and 0 to 120mV for V_W, respectively. The weight resolution achieved is 9-bit on a poly-silicon-gate n-well capacitor of 500 fF. One Least-Significant-Bit voltage (V_{LSB}) is 0.78mV.

Capacitor recharge current is estimated to be close to 5.5pA (worst case – due to forward-bias source-bulk current). This corresponds to worst-case change in weight-setting voltage of about:

$$\frac{\partial |V_{Wset}|}{\partial t} = \frac{I_{Cleak}}{C_W} = \frac{5.5pA}{0.5pF} = 11.0 \qquad (6)$$

With this rate of change, in charge conservation mode, we will be able to keep the charge from degradation of less than half of least-significant-bit ($0.5V_{LSB}$=390μV) for not more than:

$$\Delta T_{\max} = \frac{0.5V_{LSB}}{11.0} = \frac{390.0\mu V}{11.0} \cong 35\mu s \qquad (7)$$

Since one full scan-region is processed in only 8μs, we believe that 9-bit resolution is completely feasible, provided the actual leakage current does not exceed the estimated with more than 500%.

In practice, to avoid much larger leakage currents due to impurities and other fabrication non-idealities, a nonlinearity charge preservation minimization scheme is used (Figure 5).

Nonlinearity charge preservation minimization is implemented through the weight-capacitor zeroing complementary MOSFET switch[6]. In charge-preservation mode, injection of very small (leakage) current into the bulk of the current-sensing MOSFET M3, approximately equal to half of the worst-case capacitor leakage current, shifts the derivative curves down, thus providing for a smaller absolute value of weight-capacitor charge degradation across the whole range of voltages stored on the capacitor.

Since the leakage current is different for different weight-capacitor voltages, complete compensation is not possible, but rather minimization of the absolute error due to this nonlinearity in the leakage current is in place. The plot of the rate of change of weight-setting

[6] Not shown on Figure 1, detailed on Figure 5.

and weight voltages, with charge preservation error minimization is shown in Figure 4.

Mismatch in M9 and M10 between different synapses could lead to issues with this compensation. In the current design, these devices are effectively in 'off' state and provide only their channel-leakage current as injection current which seems to be not as much sensitive to mismatch.

With leakage current "compensation", weight capacitor charge degradation of one LSB or less is observed only after about 260µs, which allows for a refresh cycle to be run less often than every 8µs[7]. Since only a total of 159µs are spent in refreshing weight capacitors during the processing of one complete CIF frame, processing speed of up to 30 frames/s is feasible.

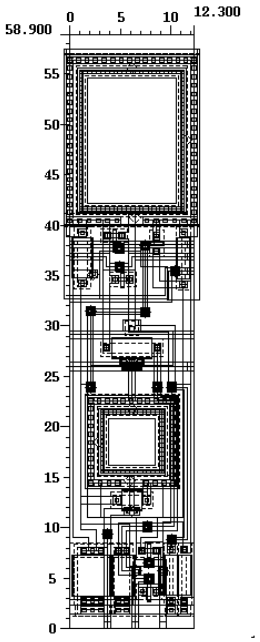

Figure 6 Physical layout of the synapse circuit. Size is in microns.

2.3 Synapse Physical Layout Topology

The primary focus in building synapse's layout topology is on minimization of silicon area, through techniques for sharing well and diffusion regions; and avoiding current switching noise by layout symmetry and shielding.

Matching techniques are used to layout and route matched-pair devices. The synapse circuit topology is shown in Figure 6. The silicon area for the synapse, weight-capacitor, weight-current switch, input capacitor and weight-control circuit is under 700µm^2. This translates into a theoretical density of more than 1,400 synapses per square millimeter. In practice, however, this is only feasible for processes with four or more layers of metal interconnect (TSMC 0.35µm). For fabrication processes with three metal layers, or less, this number will be decreased due to interconnect and shielding of sensitive signal-lines. 2,176 synapses occupy area of approximately 3 mm^2, which is less than

[7] after processing of each 4x4 scan-region

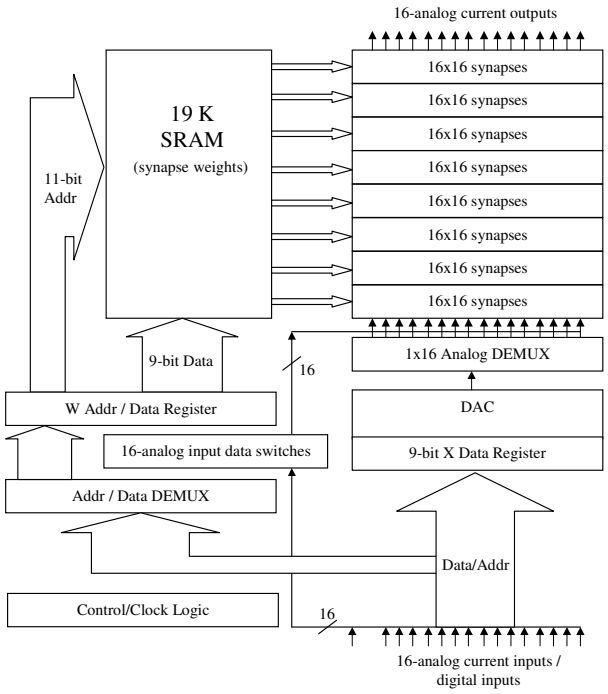

Figure 7 System-level diagram of NEURO-MATRIX-1

six percent of the total chip area (7.3x7.3µm^2). The remaining area is occupied by (in decreasing order of area) synapse-weight counters and SRAM, input DAC and digital control logic. The Neural Matrix, digital weight storage, weight control and input sample-and-hold switches, not counting control logic and the input DAC consists of more than 1,260,000 MOSFET gates.

2.4 System-Level

Neuro-Matrix-1 is a SOC comprised of: a matrix of 16x8 neurons of sixteen inputs each interconnected with sixteen system inputs by the "all-to-all" method. The system's sixteen analog-current inputs and outputs are provided for system expandability and cascaded connection between same products. The analog-current inputs are multiplexed with sixteen digital-inputs for 12-bit weight address/9-bit weight programming and 9-bit-digital input data loading. After POR (power-on-reset) the system reads-in a series of address-value pairs, which either pre-load weight values into the 19K on-chip SRAM, or set internal control registers (input-current-conversion interval programmable timer etc). After the programming is complete (EN high at next positive clock edge), the matrix is put in one of two feed-forward modes – digital-input or analog-current-input mode. Then, input processing is started. In digital-input mode, the system receives a series of sixteen 9-bit

input-data words, which are converted into input pre-synaptic current by the on-chip 9-bit DAC. Next, the input feed-forward processing is performed, simultaneously, for all inputs. In the case of analog-current input mode, the signal is processed by the neural matrix immediately after the input-current-to-voltage conversion time interval is over. This mode benefits the most from the speed of the analog signal processing technologies used and makes this ANN design a feasible choice in building analog front-end applications. A system-level diagram is shown in Figure 7.

3 Implementation Notes

3.1 Chip Training

At the present stage of the design, the training of the Neural Matrix is done externally by a general purpose digital computer (PC). The SoC, however, is included in the forward computational data path and although each update to the weights is controlled externally, the actual weight value is generated internally to the matrix. Thus final weight values are not simply computed, at the end of the training, by the external PC and then "uploaded" into the matrix, but rather obtained inside matrix circuit itself. What is externally available, at the end of training, are the digital weight control numbers which produce internally the weight quantities. These weight numbers could be stored externally for subsequent upload or programming of other chips. These numbers, however, do not necessarily have straightforward interpretation in terms of charge or voltage quantities since they do include SoC implementation non-idealities and nonlinearity built-into them. Since the actual internal charges on the weight capacitors as well as produced weight voltage across synaptic transistor terminals could only be estimated/simulated but not really extracted or measured without significant signal degradation and information loss, we have not carried any analysis or comparison for the differences that can occur when the weights are externally computed by a digital computer and simply loaded into the network as opposed to weights obtained in our training procedure. In this way, our choice for training procedure eliminates first the difficulty in trying to match the performance of the chip in real silicon with a specific computer model of the same, and second, the problem of the weights, possibly, being "different" after upload into the chip due to non-idealities. Instead, we do guarantee that the externally generated weight control numbers, when uploaded, will produce the same result as the result obtained during training.

3.2 Weight-charge refresh

Extra refresh circuits at each synapse weight are not needed since the weights are refreshed by first zeroing weight capacitor charges and then recharging them according to the weight pulse duration values stored for each synapse in the on-chip SRAM. Currently, weight refresh is performed every 8μs. For the refresh procedure two cases were considered. First, to use a "wave" refresh pattern, refreshing the weights layer by layer. And second, to refresh all weights of the matrix simultaneously. Due to issues with processing synchronization and noise encountered in the original design of Neuro-Matrix, in the current design, the second method was chosen. In this way processing synchronization is needed only for the input signal path which simplifies signal flow and increases significantly the speed of the overall input-output processing. This method, however, has the disadvantages of slightly increased silicon area and peaking of current consumption during the refresh of all weights.

Once weight control values are uploaded into SRAM, it is feasible to refresh all weights of the matrix in only 2.56μs. This is possible due to the completely parallel and distributed construction of the SRAM – each synapse has its own weight pulse counter which has a static input register keeping the weight control number for that synapse. All counters/registers are topologically located in four separate SRAM blocks away from the analog circuitry of the Neural-Matrix. Although each such register is addressed individually and could be loaded independently as a typical RAM would allow, the construction of the SRAM does not involve cell arrays and/or shared resources as address or data buss lines. This makes it possible for all weight counters to be triggered at once, and each of them simultaneously to control the charge pulse of its own synapse.

3.3 Input error accumulation due to leakage

Small errors due to charge degradation in the input sample-and-hold capacitors could accumulate during the forward computation and cause a noticeable error in the output. Therefore, the input signal is processed through the network in a "wave" pattern. Sampling and holding each previous layer input (output) current signal into input current to voltage conversion capacitors of each synapse eliminates this issue almost completely since the signal has to be stored for only a relatively short

period of time equal to synapse propagation delay plus input conversion time of the next layer.

4 Summary and Conclusion

We show that it is feasible to implement an analog synapse, a neuron and a complete ANN using only basic properties of MOSFETs in a standard CMOS fabrication process. Further, we demonstrate that the inherent quadratic non-linearity with respect to synapse weight is not detrimental to the ability of the synapse to function in feed-forward and LMS training modes of operation [1]. We do so by offering results from both theoretical and experimental research we have conducted [2][3][4][5]. We show that this simple synapse circuit proves useful in VLSI systems-on-a-chip and we demonstrate its feasibility for on-chip integration with other CMOS products. We describe a specific VLSI system implemented on TSMC 0.35µm using 2176 such synapses. We report successful results as well as shortcomings in the design and implementation of the system in support of the feasibility claims we make. Therefore, we believe that the described implementation is, in fact, valuable, especially where ANN integration with standard CMOS product is desired.

5 Acknowledgments

Most of all, I would like to thank Cynthia Roedig, Texas Instruments Inc, for proofreading the first draft and for her help in improving phrasal expressions. I very much thank Ivaylo Milev for his contribution in improving clarity of the original manuscript and overall presentation structure. I thank my wife, Diana for her support and encouragement. Also, I thank Vadim Ivanov, Texas Instruments, for his suggestion to present this paper at WSEAS-2004 conference.

References:

[1] M. Milev, "A Simple MOSFET Model of an Artificial Synapse", *Third WSEAS International Conference in Applications Of Electrical Engineering*, AEE'04, May 12-15, 2004 (pending acceptance).

[2] M. Milev, "Hardware model of a neural synapse with nonlinearity", *Tenth National Application-Scientific Conference on Electronic Technology with International Participation ET-2001*, Sozopol, Bulgaria, 2001.

[3] M. Milev, "LMS training of a PE with non-linearity with respect to weights as a linear classifier", Ninth National Conference on Electronic Technology with International Participation- ET2000, Sozopol, Bulgaria, 2000.

[4] Milev, Momtchil Mihaylov , "Method and apparatus for modeling a neural synapse function by utilizing a single conventional MOSFET", United States Patent Application No.09/968,263. Pub. No. US2002/0057606 A1 May 16, 2002, http://www.uspto.gov.

[5] M. Milev, M. Hristov, "Analog Implementation of ANN with Nonlinearity in Synapses", *NNS,IEEE Transactions on Neural Networks, Special issue on hardware implementations* pp.1187-1200 vol.14, September 2003.

NOTE TO REVIEWERS:

Due to the imposed limitations as to maximum size of the manuscripts, this paper refers and hereby includes by reference material presented in another, separate, paper[1] submitted to WSEAS-2004 AEE'04 conference board for consideration at the same time as this one. To facilitate the review of the material I kindly include reference [1] bellow.

Thank you,
For your consideration and undivided attention!

Best regards,
Momchil Milev

NOTE TO PUBLISHER:

The material to follow is not to be included as part of the paper named "VLSI Implementation of an Artificial Neural Matrix with Analog Nonlinear Synapses". It is included here for reference purpose only!

A Simple MOSFET Model of an Artificial Synapse

MOMCHIL MIHAYLOV MILEV
Faculty of Electronic Engineering and Technology[8]
Technical University of Sofia
8 St. Kliment Ohridsky Blvd, Sofia
BULGARIA
momchil@ecad.tu-sofia.bg http://ecad.tu-sofia.bg

Abstract: A simple analog-signal synapse model is developed and afterward implemented on a standard 0.35µm CMOS process to provide for large scale of integration, high processing speed and manufacturability of a multi-layer artificial neural network. Synapse non-linearity with respect to synapse weight is studied. Demonstrated is the capability of the circuit to operate in both feed-forward and learning (training) mode. The effect of the synapse's inherent quadratic nonlinearity on learning convergence and on the optimization of weight vector update direction is analyzed and found to be beneficial. The suitability of the proposed implementation for very large-scale artificial neural networks is confirmed.

Key-Words: Artificial neural networks (ANNs), analog implementation, Very Large Scale of Integration (VLSI), nonlinear synapse, synapse with nonlinearity.

1 Introduction

The signal processing speed, scale of integration, low power consumption and manufacturability of nowadays ANNs determine their feasibility and usage in real-life applications. Due to conflicting requirements in lowering the supply voltages and increasing clock speeds of the digital circuits, many researchers consider analog implementations of neural networks as a way to carry over signal processing functions with fewer numbers of active semi-conductor devices. The integration of large numbers of neurons in a single chip is beneficial since it increases the VC-dimension[1][2]. It requires the minimization of the synapse area and a more efficient way of data exchange between neurons to be devised. In this respect, analog implementations offer certain benefits making them good contenders for real-time applications. First, they offer high processing speed since the analog signal processing is carried out through summation and multiplication of continuous current or voltage signals with virtually no delay. Second, analog implementations, typically, can have larger scale of integration since they avoid data-path organization which often requires data multiplexing, bus sharing, and data-flow control logic, further limiting the effective rate at which digital neural circuits can process input signals. The main disadvantages of the analog-based designs of ANNs are considered to be their lower accuracy and the difficulties with linearity in the computations. These two factors are challenged in this article. First, it is demonstrated that the term "absolute accuracy" is often of lower significance with respect to the ability of a neural network to function in many practical applications. Second, it is demonstrated that ideal linearity in the multiplication computations is not necessarily desirable or even required In most cases, nonlinearity in the synapse transfer function is, in fact, beneficial[11][12][13]. This article is limited to the discussion of the quadratic nonlinearity in the synapse multiplication function of a specific analog implementation.

[8] Momchil Milev is currently employed by Texas Instruments Inc, Tucson, USA. The information contained in this paper is not subject to intellectual property, trademark or nondisclosure agreements, should not be affiliated with Texas Instruments Inc nor it does describe a product by TI or its subsidiaries.

The paper is structured as follows. Section 2 describes the proposed analog, nonlinear, one-transistor synapse model, explains the motivation behind avoiding use of floating-gate devices. Section 3 examines the inherent nonlinearity of the synapse with respect to its weight. Synapse model functional verification results follow brief extracts from our analytical research on the effects of the quadratic nonlinearity on the feed-forward and LMS training. Results of our circuit simulations and system-level MatLab™ verification of an artificial neuron acting as linear classifier are presented next. Summary and conclusions wrap up the paper.

2 Model overview

By using the physics of analog devices, analog implementations of ANNs offer the advantage to carry out synaptic function with only a small number of transistors. In order to benefit fully from the simple current summing law and avoid parasitic capacitive load delays, we chose pre- and post-synaptic activity signal to be represented by analog current. To simplify synapse design and minimize synapse silicon layout area, as well as to allow for ANN on-chip integration with other standard CMOS products, it is decided to implement synapse multiplication function by a simple single semiconductor device—a MOSFET.

From the first-order DC, large-signal low frequency approximation[9] model (1) of a MOSFET's drain current in non-saturated region of operation $v_{DS} \le (v_{GS} - V_T)$, after simplification[10][3][4],

$$i_D = \beta\left[(v_{GS} - V_T)v_{DS} - \tfrac{1}{2}v_{DS}^2\right](1 + \lambda v_{DS}) \qquad (1)$$

we use the product of the gate-source and drain-source voltage to produce one of the components of the synaptic activity value defined by[5]:

$$\nu = \sum_{k=1}^{N} w_k x_k \qquad (2)$$

Summing the currents of those "partial products", we get the complete "sum of the weighted products". To express this, we consider a single synapse, *k*, and define:

$$\nu_k \triangleq \tfrac{i_D}{\beta}; x_k \triangleq (v_{GS} - V_T); w_k \triangleq v_{DS};$$

Next, from (1), we derive a generic form of the quadratic nonlinearity of the synapse's internal activity field with respect to its weight:

$$\nu_k = x_k w_k - \zeta w_k^2 \qquad (3)$$

where ζ is a constant ($\zeta = 0.5$). We chose the above definitions due to practical considerations- to provide for signal values that are of the same or close order of magnitude. Nevertheless, the results in this text are more generic and can be applied to other, similar to expression (3), non-linear relationships, provided that the relationship can be approximated linearly within a certain operational range.

For N -number of synapses, the overall synaptic activity is:

$$v = \sum_{k=1}^{N} \nu_k = \sum_{k=1}^{N} x_k w_k - \zeta \sum_{k=1}^{N} w_k^2 \qquad (4)$$

Based on these considerations, a single MOSFET device offers a simple way of constructing a "linear combiner" in hardware. Its main advantage over single-transistor synapses, implemented in analog-floating-gate capable technologies, is that it does not require any special fabrication technology, and thus it is easily integrated with other standard CMOS applications to build a complete system-on-a-chip (SoC). Floating-gate technology is available in most "standard" CMOS processes; however, it is most often used for binary information storage. In order to reach a 9-bit or better analog storage resolution more specialized and expensive floating-gate fabrication technology is required. Additionally, analog floating-gate control circuits are complicated and small weight updates are difficult [14].

The proposed synapse model is inherently nonlinear but simple enough in its implementation to occupy a

[9] For $V_{GS} \ge 1.0$ V, $V_{DS} \le 100$mV and $V_{SB} = 0$V, second-order effects, including channel-length modulation, short-channel and temperature effects are estimated to contribute an average error of -5.1%. This error, however, is considered included in the overall nonlinearity of i_D and does not change the applicability of the considerations given.

[10] For typical operating drain-source voltage ($v_{DS} < 100mV$) in non-saturated mode of operation, channel-length modulation contributes error of no more than 0.02% which is ignored in further consideration.

very small silicon area, making it very useful in VLSI systems. Further, we show that this nonlinearity is not detrimental to the qualities of the proposed synapse but, in fact, could be beneficial. We also include circuit simulation and system-level behavioral simulation results that support the feasibility of using such nonlinear synapses as building blocks of ANNs.

3 Effects of the nonlinearity

3.1 Effect of synapse quadratic nonlinearity in feed-forward mode

To show the effect of the quadratic nonlinearity with respect to synapse weight, due to the described implementation, we evaluate the error defined by:

$$\varepsilon_{lin} = \frac{(\nu - \nu_{ideal})}{\nu_{ideal}} 100 \quad ,[\%] \tag{5}$$

Expressed in terms of synapse transistor quantities:

$$\varepsilon_{lin} = -\zeta\left(\frac{w}{x}\right) = -\frac{1}{2}\frac{v_{DS}}{v_{GS} - V_T} 100 \quad ,[\%] \tag{6}$$

From (6) we note that the linearity error does not depend on transistor transconductance parameters i.e., on process or geometrical parameters. For a typical signal range ($v_{DS} = 100mV$, $v_{GS} = 1.0V$, $V_T = 0.65V$), we estimate this nonlinearity "error" to be less than 15% (14.29% worst-case). We could apply an input bias

$$x_0 = -\theta\left(\zeta\sum_{k=1}^{N} w_k^2\right)^{-1} \tag{7}$$

to an extra synapse (theta-synapse) to eliminate this "offset" error[11] in feed-forward mode if needed. In feed-forward mode this bias term is a known constant, thus we could eliminate this term after network training is complete and weights are known. Such correction, however, is not applied in the experiments shown since it is our belief that this inherent offset term is accounted for by the Back-Propagation algorithm during training and, thus, it can be treated by the adaptive process as "constant input noise".

[11] in several applications, this nonlinearity in feed-forward mode proved not relevant to the success of the network for correct classification due to flexibility in the output space definition and, therefore, correction was not necessary

3.2 Effect of synapse quadratic nonlinearity in least-mean-square (LMS) training

To study the effects of the "offset" term in (4), we use the instantaneous estimate of the gradient and the method of steepest descent in LMS training:

$$\varepsilon(\mathbf{W_N}) \triangleq \tfrac{1}{2} e_N^2 \tag{8}$$

$$\mathbf{W_{N+1}} = \mathbf{W_N} - \eta.\nabla\varepsilon_{\mathbf{W}}(\mathbf{X_N}, \mathbf{W_N}) \tag{9}$$

Using equation (4) for the method of steepest descent we obtain weight update rule in vector format:

$$\nabla\varepsilon_{\mathbf{W}}(\mathbf{X_N}, \mathbf{W_N}) = (-\mathbf{X_N} + 2\zeta\mathbf{W_N})e_N \tag{10}$$

$$\mathbf{W_{N+1}} = \mathbf{W_N} + \eta\mathbf{X_N}e_N - 2\eta\zeta\mathbf{W_N}e_N \tag{11}$$

A corresponding weight-update vector diagram is shown in Figure 8. We define the difference between the update vector in the case of an ideally linear synapse output and the case of a nonlinear synapse with quadratic weight-nonlinearity as a 'residual weight gradient vector':

$$\mathbf{W}_R \triangleq 2\zeta\mathbf{W_N}e_N \tag{12}$$

We then define a 'modified'[12] instantaneous error gradient vector estimate:

$$\widetilde{\nabla}\varepsilon_W\big|_N = -\mathbf{X_N}e_N + W_R \tag{13}$$

and then re-write the weight update rule(11):

$$\mathbf{W_{N+1}} = \mathbf{W_N} + \eta(\mathbf{X_N}e_N - \mathbf{W_R}) \tag{14}$$

We have analyzed the effect of the modified gradient

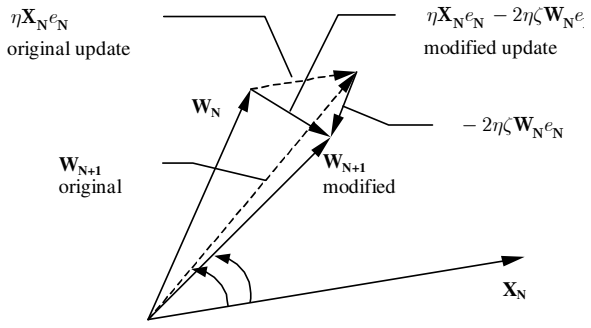

Figure 8 Modified weight vector update diagram due to 'residual weight vector' term

[12] Compared to original LMS steepest descent method

vector in two ways: effect on the direction of the weight-vector update, and effect on the magnitude (norm) of the update. We have concluded that:

1. The modified weight update vector due to (12) is always rotated in the direction of the input vector regardless of the input or weight vectors relative position or size. Thus, in both angle and Euclidean-distance sense the new weight-update vector is closer to the input vector when compared to traditional LMS with gradient descent.

a) An angle exists between the input and weight vectors

$$\alpha_{critical} = arc(\zeta \frac{\|W\|}{\|X\|}); \alpha_{critical} \in [0;\pi] \qquad (15)$$

for which:

- magnitude of the modified update is larger than the norm of the update in the original LMS steepest descent method[13] if $\measuredangle(X_N, W_N) > \alpha_{critical}$
- magnitude of the modified update vector is smaller compared to the norm of the original method if $\measuredangle(X_N, W_N) < \alpha_{critical}$

Therefore, the effect of the 'residual weight gradient vector' on the adaptation is considered beneficial – increasing the amount of update, hence, speeding up the convergence of the weight vector when it is 'far' from the steepest descent direction and decreasing the amount of update for weight vectors close to the direction of steepest descent [6][10]. The latter is considered helpful in avoiding weight-vector oscillations around the optimum solution for increased learning-rates, thus, again providing faster convergence conditions.

Additionally, by expanding the error cost function in a Taylor series around the weight vector at any given time, it has been proven that the error is minimized with every step of the iterative descent regardless of the modification due to the residual weight gradient vector i.e. synapse quadratic nonlinearity with respect to its weight. A comparative analysis was also conducted between the modified update(11) and the generalized 'delta rule' including the 'momentum term' as it is known by Rumelhart et al [7]. It was concluded that, while the use of the momentum term can decrease the stable range of the learning rate parameter and lead to instability [8][9], the effect of the residual weight vector, in contrast, does not decrease the learning rate range and is stabilizing inside the $\alpha_{critical}$-determined spatial cone. The details of this research, however, are outside of the scope of the present article and are not included here. More information on training ANNs with non-linear synapses can be found in [15][16].

3.3 Experimental data

To verify and support the theoretical findings, a number of circuit-level and system-level simulations were carried out. Circuit level simulation results and plots for nonlinear synapse operation, weight-charging, input-signal conversion and others are exhaustive and available from the author upon request[17][19]. System-level simulations were conducted using MatLab™ software to train and test a neuron using synapses with quadratic nonlinear synapses as modeled by (3) to perform a linear classifier function. Sets of 2D linearly separable clusters of random vectors were generated and then LMS steepest descent training was performed over the same data twice – once for a neuron having ideally linear synapses and again for the described model of a neuron with nonlinear synapses. More than 200 simulation runs over clusters of 100 vectors with varying cluster size and dispersion were evaluated. The results showed [10] that the classification success of the neuron using nonlinear synapses modeled by (4) was, generally, not lower than the success rate of the correct classification of the neuron with linear synapses, and in many instances was better. Additionally, in most cases, convergence during the training of the neuron using nonlinear synapses was reached in fewer epochs than for the case of the neuron with ideally linear synapses. The results for the original neuron with ideally linear synapse are depicted by an 'o'-symbol and the results of the neuron with quadratic nonlinearity in the synapses are shown with an 'x'-symbol. Selected plots showing the final MSE, the number of epochs in which convergence was reached, final learning rate parameters and rate of successful classification for the training and test runs in the two cases are shown in Figure 9 through Figure 13.

[13] For same learning rate and instantaneous error amount

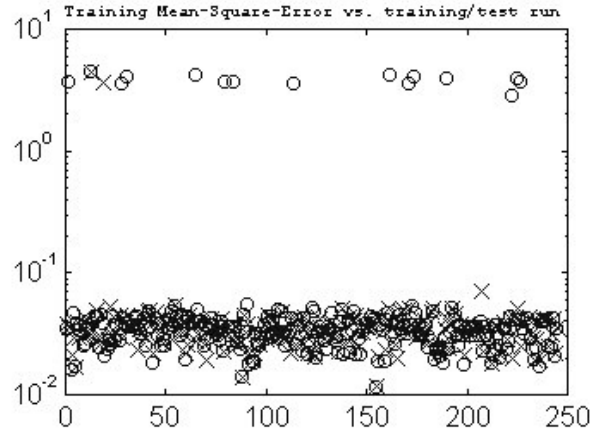

Figure 9 Final MSE vs. training/test run number

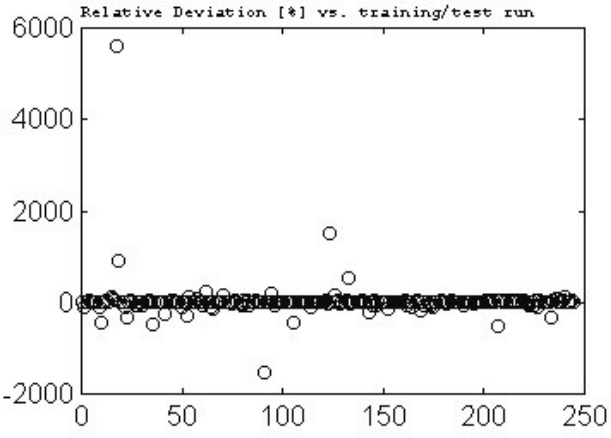

Figure 10 Relative deviation in percent of the output response of a neuron using nonlinear synapses with quadratic nonlinearity vs the response of a neuron with linear synapses

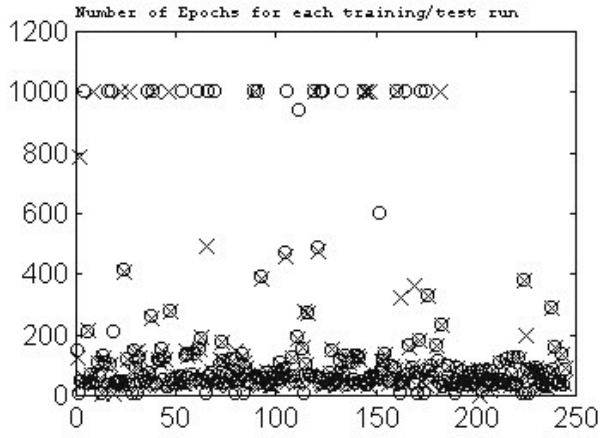

Figure 11 Number of epochs in which convergence was reached in the case of the original linear synapse neuron model and neuron using nonlinear synapses

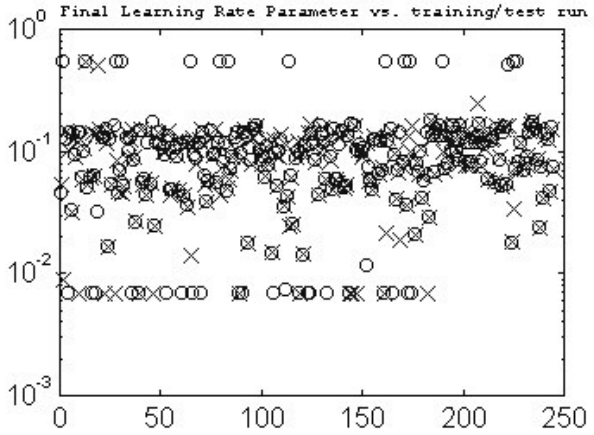

Figure 12 Final learning rate parameter for each training run

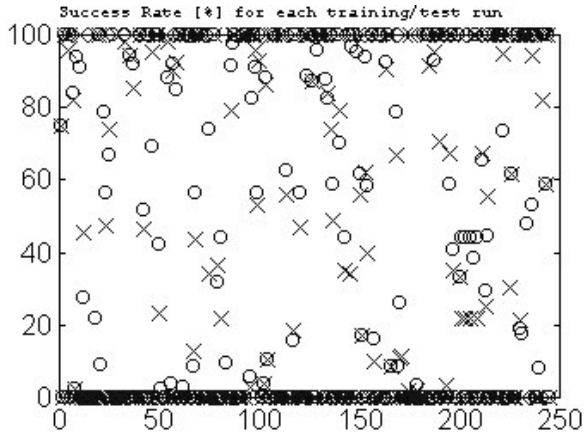

Figure 13 Rate of successful classification of the neuron with linear synapses -'o' and with non-linear synapses -'x'

4 Summary and Conclusion

We show that it is feasible to implement an analog synapse using only basic properties of MOSFETs in a standard CMOS fabrication process. We describe the model and investigate its operation in feed-forward and learning modes of operation. Due to limited size of this presentation, we show the complete design of the synapse circuit, a neuron and ANN based on this model in [5] and [19]. We demonstrate that not only is the inherent quadratic non-linearity with respect to synapse weight not detrimental to the ability of the synapse to function in LMS training mode, but also that the latter can offer distinct advantages in learning convergence. We do so by offering results from both theoretical and experimental research we have conducted. We suggest that a simple synapse circuit, based on this synapse model can prove useful in VLSI systems-on-a-chip and we further exploit this topic in

[19] to demonstrate its feasibility for on-chip integration with other CMOS products.

5 Acknowledgments

Most of all, I would like to thank Cynthia Roedig, Texas Instruments Inc, for proofreading the first draft and for her help in improving phrasal expressions. I very much thank Ivaylo Milev for his contribution in improving clarity of the original manuscript and overall presentation structure. I thank my wife, Diana for her support and encouragement. Also, I thank Vadim Ivanov, Texas Instruments, for his suggestion to present this paper at WSEAS-2004 conference.

References:

[1] A.A. Blumer, Ehrenfeucht, D. Haussler and M.K. Warmuth, "Learnability and the Vapnik-Chervonenkis Dimension", *Journal of the Association for Computing Machinery*, vol.36, pp.929-965, 1989.

[2] R.W. Newcomb, N. El-Leithy, "Perspectives on realizations of neural networks", *CAS,IEEE International Symposium on Neural Networks,* pp.818-819 vol.2, 1989.

[3] Phillip E. Allen, Douglas R. Holberg, *CMOS Analog Circuit Design*, Oxford University Press, 1987; pp.98-101, pp.198-207.

[4] Roubik Gregorian, Gabor C. Temes, *Analog MOS Integrated Circuits For Signal Processing*, A Wiley-Interscience Publication, John Wiley & Sons, New York, 1986; pp.462-483.

[5] Simon Haykin, "*Neural Networks – A Comprehensive Foundation*", second edition, 1999 Prentice Hall, New Jersey; pp. 3-18, pp.51-53, pp.121-122, pp.128-132.

[6] M. Milev, "Hardware model of a neural synapse with nonlinearity", *Tenth National Application-Scientific Conference on Electronic Technology with International Participation ET-2001*, Sozopol, Bulgaria, 2001.

[7] D.E. Rumelhart, G.E. Hinton, R.J. Williams, "Learning representations of back-propagation errors", *Nature (London)*, vol.323, pp.533-536.

[8] M. Hagiwara, "Theoretical derivation of momentum term in back-propagation", *International Joint Conference on Neural Networks*, vol. I, pp.682-686, Baltimore 1992.

[9] Simon Haykin, "*Neural Networks – A Comprehensive Foundation*", second edition, 1999 Prentice Hall, New Jersey; pp.170-171, p.249 note 2.

[10] M. Milev, "LMS training of a PE with non-linearity with respect to weights as a linear classifier", *Ninth National Conference on Electronic Technology with International Participation- ET2000*, Sozopol, Bulgaria, 2000.

[11] Liang, P.; Jamali, N. , "Artificial neural networks with nonlinear synapses and nonlinear synaptic contacts", *Systems, Man and Cybernetics, 1992., IEEE International Conference on , 1992* pp.1043 - 1048 vol.2

[12] Lont, J.B.; Guggenbuhl, W, "Analog CMOS implementation of a multilayer perceptron with nonlinear synapses", *IEEE Transactions on Neural Networks ,* vol.3 issue 3 , May 1992 pp. 457 -465

[13] Lont, J.B., "A large scale neural network with nonlinear synapses", *1994 IEEE World Congress on Computational Intelligence., 1994 IEEE International Conference on Neural Networks*, Volume: 1 , 1994 pp. 350 -353 vol.1

[14] Horio, Y.; Ymamamoto, M.; Nakamura, S, "Active analog memories for neuro-computing", *IEEE International Symposium on Circuits and Systems, 1990*, 1-3 May 1990 pp. 2986 -2989 vol.4.

[15] Nakayama, K.; Hirano, A.; Fusakawa, M., "A selective learning algorithm for nonlinear synapses in multilayer neural networks", *Proceedings. IJCNN '01. International Joint Conference on Neural Networks, 2001*, Volume: 3, 2001, pp. 1704 -1709 vol.3

[16] Myung-Ryul Choi; Jin-Sung Park , "Implementation of MEBP learning circuitry with simple nonlinear synapse circuits", *Fuzzy Systems Conference Proceedings, 1999. FUZZ-IEEE '99. 1999 IEEE International* , Volume: 1 , 1999 pp. 315 -320 vol.1

[17] Milev, Momtchil Mihaylov , "Method and apparatus for modeling a neural synapse function by utilizing a single conventional MOSFET", United States Patent Application No.09/968,263. Pub. No. US2002/0057606 A1 May 16, 2002, http://www.uspto.gov.

[18] M. Milev, M. Hristov, "Analog Implementation of ANN with Nonlinearity in Synapses", *NNS,IEEE Transactions on Neural Networks, Special issue on hardware implementations* pp.1187-1200 vol.14, 2003.

[19] M. Milev, "VLSI Implementation of an Artificial Neural Matrix with Analog Nonlinear Synapses", *Third WSEAS International Conference in Applications of Electrical Engineering*, AEE'04, May 12-15, 2004.