A Simple Hardware Model of a Synaptic Function of a Neuron for CMOS VLSI implementation.

Eng. Momchil Mihaylov Milev, Technical University - Sofia
Assoc. Prof. Ph.D. Marin Hristov Hristov, Technical University – Sofia
milev_momtchil@ti.com mhristov@ecad4sun.vmei.acad.bg

Synopsis: The report gives a brief overview of the most commonly used artificial neuron computational model and then leads to the construction of a hardware model using MOSFET devices as the essential elements for modeling the synaptic connections of the neuron. It also gives an outline of the construction of the complete hardware model of an artificial neuron with any number of synaptic connections. The objective is not to present the complete model in detail which is a subject of another publication, but to focus on the benefits of the specific hardware implementation of such a model using a single MOSFET device in any conventional CMOS technology.

Biological roots. An overview of artificial neuron modeling.

The computational artificial neuron model that is a primarily processing unit in nowadays simulations of artificial neural networks, has definitely some roots leading back to it's biological counterpart – the cerebral cortex neuron or the so called "delta cell" in the brain of many species with highly developed nervous system. We can make a definite parallel between the various computational functions used in the artificial neural network models and their biological counterparts. For instance, the synaptic connections of the neuron's synapses are modeled by weight-coefficients applied to the "input signals" along the dendrite trees – then the "output" of the neuron can be found to be modeled by a non-linear function (typically of sigmoidal form) that processes the neuron's internal activity such as to model properly both the state of saturation and very low activity and its response produced along the axon of a neuron.

Nevertheless, the computational model is far from "mimicking" the original neuron and the neural network science and its applications are already far from the idea of simulating the actual functioning of the human brain. However, this does not in any sense limit the vast domain of research, applications and opportunities to use an "artificial neural networks" to find solutions in areas where any other engineering or mathematical algorithmic approach has failed. That is why most often we will reference the artificial neuron simply as a Processing Element (PE). Typically, a given PE simulates an artificial neuron by linearly combining the weighted input signals (from its synaptic connections) and then applying a non-linear saturation function over the above

sum to produce the neuron's response to its internal activity level. Such a PE input weights can be adjusted (trained) by a number of various optimization algorithms iteratively minimizing an output error estimate function, so that the output response matches with the desired pattern of responses to the input stimuli.

The input-output relation of the "linear Processing Element" (so called "linear combiner") is given by:

$$v = \Sigma\, w_k\, x_k \qquad \{1\}$$

where x_k *is* the k-th vector component (input) and v is the output of the linear combiner or so called "internal activity" value. Further, typically, such a PE will have a sigmoidal type of non-linear function applied at his output as mentioned above.

$$y = \text{sigm}(\, v\,) = \text{sigm}(\, \Sigma\, w_k\, x_k + \theta\,) \qquad \{2\}$$

Since this (sigmoidal) function is most "sensitive" around the origin of the x-axis, additional free parameter θ is used to "adjust" the neuron's activity to be in that vicinity at least initially. This parameter is adjusted ("trained") usually along with all the "weights" - free-parameters w_k which are adjusted according to the "error signal" computed at the output of the PE. As we will see further, just the existence of the mentioned free parameter θ will play a crucial role in the considered hardware model.

Now, let's take a look at what are the possible ways of building a hardware model of the PE described in such simplicity by the above mathematical relationship.

Use of a single MOSFET to model the "synaptic connection of a neuron".

It is not hard to see that there are numerous ways in which the above relationship can be implemented in hardware. One can find various implementations in the literature and technical conference proceedings starting with the use of a numerical computer or at least it's ALU, using digital multipliers and adders or going through frequency multipliers and voltage or current controlled oscillators (VCO's) reaching really innovative solutions involving "optical transistors" or polarized laser beams modulated by liquid-crystal lattices. Here, the author presents the use of MOSFET devices to model the above relationship, thus modeling the synaptic function of a neuron.

Let's look into the first-order approximated model of the MOSFET drain current (assuming linear region of operation ($U_{ds} \le U_{gs} - U_{th}$):

$$I_{ds} = \beta\,[\,(U_{gs} - U_{th})\,U_{ds} - \tfrac{1}{2}U_{ds}^{\,2}\,] \qquad \{3\}$$

As we can see, we can use the gate voltage to control the drain current which depends additionally on the drain-source voltage. Thus, we can use the product of the gate and drain-source voltages to produce one of the components of the above relation

{1}, then summing the currents of those "partial products" to produce the complete "sum of the products". Let's define:

$$v \equiv I_{ds}, \quad x \equiv U_{gs} - U_{th}, \quad w = \beta U_{ds}, \quad \text{then we have:}$$

$$v = x.w - c.w^2, \quad \text{where } c = 1/2\beta \qquad \{4\}$$

By that merit a single MOSFET device offers great potential and yields a rather simple and unique way of constructing a "linear combiner" part of an "artificial neuron" in hardware. It is true that the above relationship is "far" from linear in general. In that respect, the suitability of the above might be under question. But in practice, it can be almost always provided that the drain-source voltage is much less the "effective gate" voltage such that the above relationship is "very close" to linear.

Let's examine the linearity error:

$$\varepsilon = (v - v_{ideal})/v_{ideal}.100, \qquad [\%]$$

or, $\varepsilon = -c\,(w/x) = -0.5\,U_{ds}/(U_{gs}-U_{th}).\,100\,, \qquad [\%]$

This leads to the following observations for the linearity error defined as above:

- the error is independent of β (independent from MOSFET geometry and technology process constants)
- the smaller the drain-source voltage and the greater the "effective" gate-source voltage is, the "more linear" is the relationship.

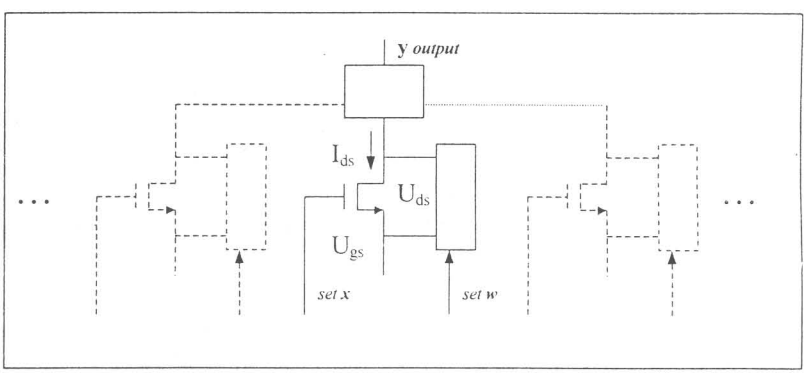
Let's inspect that error, for typical values of the above quantities: for U_{ds} = 200 mv and U_{gs}–U_{th} = 1v, we have ε = 10 %. Is this "that" bad ? No, not at all. This error is completely acceptable, due to the following two main reasons:

- the output of the linear combiner is usually processed by a non-linear function anyway, a large variety of these functions have been considered in the literature which all show very similar performance results, from which we can infer that absolutely accurate performance of the linear combiner is not absolutely "required" since the final output varies for the various functions..
- the training of the weight coefficients of the above PE is adaptive, it can and will compensate for "some" non-linearity in the linear combiner.(see [3])

General overview of the hardware model with any number of "synapses".

This article will only briefly show the hardware of the above model using MOSFET devices in any, even low performance CMOS technology. For the complete hardware solution see [4].

The essence of the given hardware solution is a single MOSFET device to model the synaptic activity of an artificial "neuron". The voltage across the source and drain regions determines the weighting coefficient(s) in {4} and the gate-source voltage represents the input stimulus applied to that synaptic connection either by the input signal of the network at that node or by the previous-layer PE's. The drain current representing the "internal activity" of that particular synaptic connection is then summed up along with all of the "activity" signals induced at the rest of the "neuron" synapses. Finally the overall "activity" signal is processed through a non-linear (sigmoidal) function to generate the "neuron's output" signal. The described model is shown on Fig.1.



Advantages of the given hardware model

The advantages of the above presented model are primarily stemming from the fact that it is hardware and as such it will not require any computational operations of a computer, thus in feed-forward mode allowing for real-time applications for signal processing, pattern recognition etc. So, the first unbeatable advantage over software models is *speed*. The second is *simplicity*, which in comparison to the hardware models using arithmetic units, adders and multipliers is in order of magnitudes less in terms of number of gates. Furthermore, it allows for significantly large scale of integration which can easily accommodate 10.10^{+3} to 100.10^{+3} number of PE's in a single chip. As noted in [5], the "capacity" of a neural network to "memorize" is proportional to the order of the number of processing elements N given by:

$$C \sim O(N^2)$$

Which means that the VLSI implementation with as many PE's as possible is desired could be desirable and beneficial. Thus *feasibility to VLSI* implementation is also essential advantage of the above described approach.

Summary

In summary, one can see that simulating an artificial neural network in hardware can be very advantageous in many aspects. Many of those advantages are being utilized in the given model. The distinct features of the presented hardware model are it's completely analog circuit design, simplicity and VLSI feasibility. More specifically, the use of a single MOSFET device to model the synaptic connection strength (weight) not only offers much more straightforward implementation then any of the digital implementations observed in the literature but also among the analog processing solutions found is probably one of the simplest to model the multiplication function. Here it should be noted that the given solution does not impose no special requirements neither to the lateral topology of the MOSFET nor to the vertical geometry and technology parameters. This is major advantage since it allows the model to be implemented even on low-performance CMOS technologies and represents generally technology independent design.

References:

[1] "Neural Networks – a comprehensive foundation", Simon Haykin, 1991 ·

[2] "CMOS Analog Circuit Design", P.Allen, D.Holberg, 1987

[3] "LMS training of a PE with non-linearity with respect to weights as a linear classifier", M.Milev, IEEE publication pending, 2000.

[4] "Hardware model of an Artificial Neural Network for VLSI CMOS implementation", M.Milev, (IEEE publication and patent pending).

[5] "Neural Networks – anniversary overview", IEEE transactions, CAS, 1992.