Comparison of Three Different Architectures for MOS-Compatible Quadratic Synapses

S. Mehdi Fakhraie, K. C. Smith, J. M. Xu*

Department of Electrical and Computer Engineering
University of Toronto
Toronto, Ontario M5S 1A4, Canada

* On Leave at the
Department of Electrical and Electronics Engineering
Hong Kong University of Science and Technology
Hong Kong

Abstract: Inspired by the harmony between the basic functional elements of biological neural networks and their natural operating media, we have been seeking for ways to implement Artificial Neural Networks (ANNs) using the intrinsic functionality of the most commonly available devices in an electronics technology, in contrast to the method of hardware-compilation of software-simulation modules. In the case of MOS technology, we employ a quadratic functional equation similar to that found in standard MOS transistors to implement synapses in ANNs. A structure has been proposed in [3] to implement a MOS device with externally-controllable threshold voltage to be employed as a synapse. In the present work, we develop and compare practical architectures within which these synapses can be utilized optimally. A simulator and proper training algorithms have been developed to simulate different hardware-based architectures.

1. Introduction

Artificial Neural Networks (ANNs) are adaptable architectures composed of simple processing elements interconnected by a set of links. Parallelism in the processing of the input data is one of the major advantages of this computational paradigm. Among possible implementations, analog-hardware implementation can possibly take best advantage of full parallelism [1].

Silicon compilation of software-based ANNs is not the most efficient way for hardware-implementation, because it does not attempt to make optimum use of the hardware resources. As a possible direction for efficient solution, we have constructed artificial networks from the simplest and smallest building blocks available in a MOS technology having similar functionality to that of software-based networks. Starting with a single transistor as a simple building block, we have investigated many possible variations by which to use it in an adaptable fashion while preserving stable and predictable characteristics.

Our theoretical investigations and simulations have proven that a network composed of neurons with sigmoid activation functions interconnected with single-transistor synaptic blocks can be used successfully in the implementation of feedforward-multilayer perceptron networks [2]. As will become clear below, the intrinsic MOS characteristics of $I = k\cdot(V - V_{th})^2$ provides the basis for these new architectures.

2. MOS-Compatible Quadratic Synapses

A quadratic synapse is implemented by the equation: $i_{out} = k\cdot(v_{in} - W)^2$, where "W" is the adaptable parameter, or memorized synaptic information. This resembles the characteristic equation of a MOS transistor with an adjustable threshold voltage. Such a MOS device can be realized in several ways: One of them is to use charge-injection floating-gate transistors. Another solution is to apply a control voltage to the substrate of the transistor. However to do this, each device must be fabricated in a separate well. Another solution is to use a double-gate transistor, where one gate is used to control the threshold voltage as seen from the other one [3]. The latter is more readily adaptable and suffers less from cycling and durability in subsequent read and write operations. However, it requires means by which to store analog control voltages on the chip. Moreover, if a dynamic capacitive storage scheme is used, the analog memory has to be refreshed continuously. In [3], it has been shown that the equation $i_{out} = k\cdot(v_{in} - W)^2$, where v_{in} is applied to the input terminal and k is a constant, characterizes our proposed synaptic device. Because of the capability of this transistor to perform as a basic quadratic synapse, we call it a Synapse-MOS or a SyMOS device. We should note that this device is superficially similar to the one introduced in [4], however, our design approach is different than the one followed there.

0-7803-1865-X/94/$3.00 © 1994 IEEE

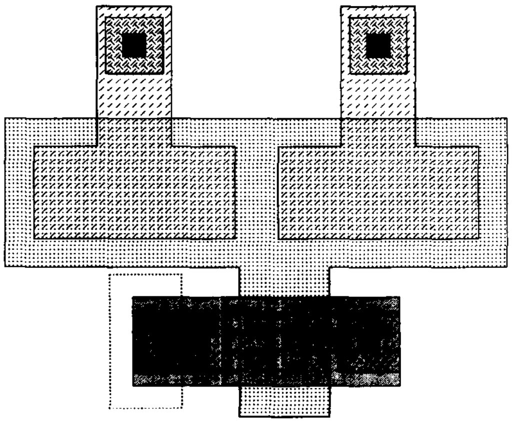

Fig. 1. Layout of a SyMOS device fabricated in CMOS4S[1] technology.

Employment of the smallest possible subblock in the technology, is achieved by the construction of networks based directly on the functional requirements and the theoretical guidelines, a process which excludes the overhead paid to translate a software-implementation of the theory into hardware modules. We have proved that these networks are trainable through the solution of a constrained optimization problem [2], and an algorithm similar to the error back-propagation algorithm [5] has been developed.

In this design, extra constraints are required to preserve the operating conditions of the synaptic devices: The synaptic transistors are employed in their saturation region of operation. This provides higher speed, higher stability, and achieves some application-oriented advantages such as more efficient detection of quadratic features in the input space.

We have solved several test problems using simulations of our MOS-compatible quadratic ANNs. Comparison of their discriminating performance with perceptrons using linear synapses has demonstrated that these networks are far superior to those with linear synapses in the detection of quadratic features. Figure 2 illustrates the discriminating surfaces formed by each type.

According to these geometrical interpretations and their governing equations, our MOS-compatible networks can be categorized in the class of ANNs with nonlinear discriminating surfaces. Among them are Radial Bases Functions (RBFs) [6], software-based quadratic neural networks[7], and some unsupervised ANNs [8].

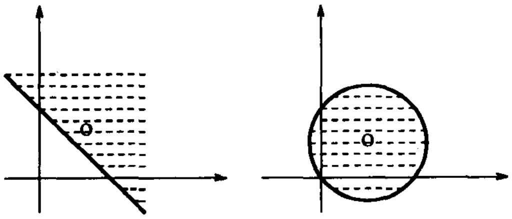

Fig. 2. Quadratic synapses form quadratic discriminating surfaces, while linear synapses produce hyperplanes.

3. Three Architectures by Which to Employ SyMOS Transistors

A SyMOS device is the lowest-level hardware block. It has certain operating constraints and limitations. For example its output current is always positive. As well, the input voltage always should be greater than the threshold voltage. Further, in order to operate in the saturation region, its drain voltage should be greater than gate minus threshold voltage. All of these restrictions will limit our freedom in using it. In order to maintain the usefulness of this device in real applications, we should find a good architecture in which to obtain the maximum device utilization. In this paper, we reflect our attempts to find an optimized architecture to employ SyMOS devices.

3.1. Current-Source Inhibited Architecture (CSIA)

In the equation $i_{out} = k \cdot (v_{in} - W)^2$, i_{out} is always positive. This, together with the operating condition ($v_{in} \geq W$) will limit the range of application of this synapse. For example, a single neuron using two of these synapses can discriminate only points inside and outside a quarter-circle rather than implementing the complete theoretic circle-detection function. In neural-network terms, a synapse of this type, provides only excitatory stimulation, without any possibility of implementing inhibition.

In order to provide, selectively, either inhibition or excitation, we have adopted the synaptic relation $i_{out} = (v_{in} - W)^2 - 1$, where "W" can take on either negative or positive values. Depending on its magnitude in comparison to the input value, the output i_{out} becomes positive (excitatory) or negative (inhibitory). In consideration of its accompanying hardware implementation, we call this implementation the Current-Source-Inhibited (CSI) quadratic synapse.

3.2. Switchable-Sign-Synapse Architecture (SSSA)

Returning to the geometric interpretation of quadratic synapses, a neuron with positive quadratic synapses and with a negative bias term represents a hyper-sphere [$(i_1 - w_1)^2 + \cdots + (i_N - w_N)^2 - R^2 = 0$] in the N dimensional input space.

[1] CMOS4S is Northern Telecom Electronics 1.2μm CMOS technology provided to us through the auspices of Canadian Microelectronics Corporation (CMC).

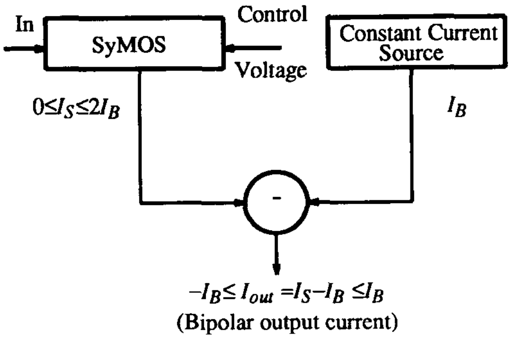

Fig. 3. Block diagram of the Current-Source Inhibited Architecture (CSIA).

If some of the terms have a negative sign, then it will represent other quadratic shapes, for example a part of a hyperboloid. Therefore if we can switch the sign of the output of a synapse, the neuron can be extended to discriminate points inside and outside of various quadratic hyper-surfaces. This is an advantage which will compensate the effects of the limited operating range of an individual synapse. Correspondingly, in neural terms, each synapse can show either an excitatory or inhibitory behavior. In this architecture, we add a sign-of-synapse register, which is externally programmable, and, in turn, determines the way in which a pair of switches connect the output of the synapse to one of the positive or negative input lines of the neuron.

3.3. Digital-Analog Switchable-Sign-Synapse Architecture (DASA)

Noting that in sub-micron MOS technologies, fast minimum-size digital devices are readily available, we have designed DASA in an attempt to reduce the effects of the constraints imposed on the operation of our SyMOS devices.

In the DASA architecture, in each processing cell (a synapse) we process sign and amplitude variables separately, and then combine them. Each synaptic cell has a weight factor embedded in the threshold voltage of its SyMOS device. As well, we have a sign-bit register storing the introduced sign-of-synapse variable (S_1). Both inputs and thresholds vary between -1 and 1. The absolute value of the input is used as the input of the SyMOS device. Its sign (S_2) is multiplied by the sign-of-synapse variable and the result of this operation determines whether the output of the synaptic operation is inhibitory or excitatory. The following equation shows the final output of the synapse:

$$i_{out} = (S_1 \cdot S_2) \cdot (\mid v_{in} \mid - W)^2$$

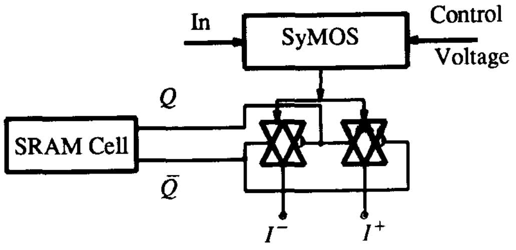

Fig. 4. Block diagram of the Switchable-Sign-Synapse Architecture (SSSA)

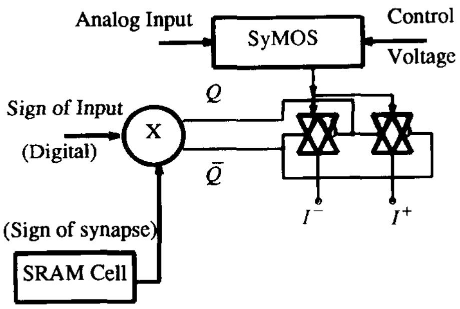

Fig. 5. Block diagram of the combined digital and analog Switchable-Sign-Synapse Architecture (DASA).

In this equation, none of the multiplications is analog: S_1 and S_2 are digital values. The result of their sign multiplication can be obtained by the use of a minimum-size digital XNOR gate. Multiplication of this result by the output of the SyMOS is achieved by proper adjustment of the switches connecting the SyMOS to one of the inhibitory or excitatory inputs of the neuron body.

In all of these architectures, each neuron is composed of a number of synapses connected to summing input lines. In some cases, the neuron has two inhibitory and excitatory input lines, whose values are finally combined with proper sign. The output stage has a "$\frac{1}{1+e^{-x}}$" functional relation for binary input-output levels and a "$\frac{2}{1+e^{-x}} - 1$" relation for cases with bipolar outputs.

4. Simulations

We have developed a software simulator to imitate the operating conditions and constraints of a hardware implementation. A training algorithm, employing a modified version of error back-propagation, has been developed [2], and, for each case, proper training equations have been derived. Using this simulator, we have examined many different network compositions and have compared their performances based on criteria such as convergence of the training algorithm, stability, running time, number of iterations to achieve an acceptable solution, hardware complexity, number of interconnects, and achievable low and high output levels with the same number of hidden units.

The test problems considered were logical functions of two input variables with emphasis on XOR and XNOR, and a simple character-recognition problem based on data provided through a 5-by-3-input retina. The networks tested are composed of input, hidden and output layers. In each application, the number of units in the input and output layers is fixed, while the number of units in the hidden layer is gradually increased. For each case, the results obtained from different architectures are compared.

5. Summary of Results

CSIA: CSIA has the simplest architecture. All of its decisions are being made locally. Its interconnects are minimum in comparison to the other two schemes. It suffers most from the limited operating range of SyMOS devices. For example, in the case of an XOR problem with two neurons in the hidden layer, maximum low and minimum high are at the 36% and 57% levels, leaving only a 19% noise margin. The noise margin will improve by using a higher number of hidden units. It can be used in either binary or bipolar configurations. The performance is better in the bipolar case, because of wider operating margins. As well, it may suffer from a greater static power consumption, however, we can add a switch to turn the current source off when the synapse is not active.

DASA: DASA shows the best performance. It can be designed to operate using local operations and decisions. However, in this case, each cell gets more complex and requires more area in terms of necessary analog and digital hardware. Some of the operations and decisions can be transferred outside and be handled remotely; however, more area is then consumed by the interconnect lines. In the case of the XOR problem with two neurons in the hidden layer, a margin of 90% between worst-case high and low levels is obtained.

SSSA: SSSA is relatively simple in comparison to DASA. It requires less hardware in each cell. In each synapse, the sign-of-synapse register is randomly set by the training algorithm: it connects the output current of the synapse to one of inhibitory or excitatory inputs of the neuron accordingly. Correspondingly, we need a module in the training algorithm to detect the necessity for a change in the sign of a synapse and to apply that change. This also can be handled locally by monitoring the situation in which a synaptic weight approaches its extreme low or high value. In our current approach, in order to maintain faster after-training hardware, we handle this job globally. When the necessity of a change is detected, we modify the sign-of-synapse register. SSSA has a more direct physical and geometrical interpretation, which makes its application to problems easier. Also its global-sign-adjustment version requires less analog area in comparison to the other two architectures. For the XOR problem, a margin of 60% between low and high output levels is achieved using only two hidden-layer neurons.

All three of the architectures are able to solve the test character-recognition problem, however with different training time and complexity. In order to achieve similar performance, CSIA requires the highest and DASA the lowest number of units in the hidden layer.

we have fabricated our basic cells in the CMOS4S $1.2\mu m$ technology. For solution of simple pattern-recognition problems, we are working on the implementation of larger networks with SSSA.

6. References

1. C. A. Mead, *Analog VLSI and Neural Systems.* Reading, MA: Addison-Wesley, 1989.
2. S. M. Fakhraie, and K. C. Smith, "Generalized Artificial Neural Networks," in *Proc. Canadian Conf. Elec. & Comp. Eng.* (Vancouver, BC, Canada), pp. 469-472, Sep. 14-17, 1993.
3. S. M. Fakhraie, and K. C. Smith, "Synapse-MOS (SyMOS) transistors: Intelligent MOS transistors with learning thresholds," in *Proc. Canadian Conf. VLSI* (Banff, Alberta, Canada), pp. 7.28-7.33, Nov. 14-16, 1993.
4. T. Shibata, and T. Ohmi, " A functional MOS transistor featuring gate-level weighted sum and threshold operations, " *IEEE Trans. on Electron Devices*, 39(6), pp. 1444-1455, June 1992.
5. D. Rumelhart, G. Hinton, and R. Williams, "Learning internal representation by error propagation, " in *Parallel Distributed Processing.* D. Rumelhart, J. McClelland, and the PDP Research Group, Cambridge, MA: The MIT Press, 1986.
6. J. Moody, and C. Darken, "Fast learning in neural networks of locally-tuned processing units, " *Neural Computation*, No. 1, pp. 281-294, 1989.
7. J. Volper, and S. Hampson, "Quadratic function nodes: use, structure, and training, " *Neural Networks*, No. 3, pp. 93-107, 1990.
8. J. Hertz, A. Krogh, R. Palmer, *Introduction to the Theory of Neural Computation.* Redwood City, CA: Addison-Wesley, 1991.