# A Hierarchical Approach to the Analog Behavioral Modeling of Neural Networks using HDL-A[1]

Mona M.Ahmed, and Hani F.Ragaie
ECE Dept., Faculty of Engineering, Ain Shams University, Cairo, Egypt

Hisham Haddara
ANACAD Egypt, Cairo, Egypt

E-mail: ICL@shams.eun.eg

## Abstract

*A hierarchical methodology for analog behavioral modeling of the basic building blocks of neural networks is presented using HDL-A[1]. This hierarchy is formed of three levels in order to satisfy the different requirements of the CAD tools which may incorporate the models. The presented models include all the nonidealities present in the actual circuit in addition to being flexible and consuming shorter simulation time. This improvement in simulation time is verified through examples at both the circuit and system levels.*

## 1.Introduction

The concept of behavioral modeling has been efficiently utilized in the digital circuits domain leading to a considerable reduction in the design cycle [1]. When being applied to the analog domain, several difficulties were encountered mainly due to the fact that high level analog functions are simple and highly technology dependent which contradicts with the abstractness and technology independence of the modeling concept. Nevertheless behavioral models incorporated in libraries for systems, utilizing the analog VLSI technology as a major technology such as artificial neural networks (ANNs), can enhance the activities in the field of analog design automation.

Analog VLSI is one of three technologies used for hardware implementation of ANNs. Its suitability arises from the small computational circuits and the high speed which it is capable of providing, thus leading to fast and area efficient ANN systems with low power dissipation[2].

ANNs have a structure composed of a limited set of electronic circuits mainly programmable synapses (multipliers), variable gain neurons (nonlinear elements), summers, storage elements, and winner-take-all circuits [2-5,7]. There are other circuits depending on the application as the cochlea, analog delays [6], and the Guassian circuit that can be employed in speech recognition systems. Several designs for the above mentioned circuits have been proposed [6-13] starting from the pioneering work of Mead [14]. Most of these circuits are simple and programmable, thus any proposed model should be as simple as the circuit and consuming a shorter simulation time.

In this paper, we propose a hierarchical methodology for the modeling of analog primitive cells and apply it to the case of ANNs. The objective of this work is to develop a library for the analog cells used in neural networks. In the next section, the applicability of the modeling concept to the neural networks case is discussed. Section 3 presents the proposed modeling approach and section 4 emphasizes it through examples at the circuit and the system levels together with the simulation results. Finally conclusions are drawn in section 5.

## 2.Behavioral Modeling of Neural Networks

Digital CAD tools use behavioral modeling at the design, simulation, and validation phases. Analog circuits have not yet mature CAD tools as those of their digital counterpart[15].

The field of ANNs is one of those fields employing analog VLSI technology in addition to its nature being formed of a limited set of modules. Applying the concept of behavioral modeling to neural networks may have some restrictions.

- ANNs are still in a development phase lacking well approved circuit architectures.
- The model should be independent of the nature of the CAD tools utilizing it.
- The model should be closer to the circuit level in order to be distinguished from the algorithms used for software simulation of ANNs.

[1] HDL-A is a VHDL based analog language developed by ANACAD Electrical Engineering Software.

EURO-DAC '96 with EURO-VHDL '96
0-89791-848-7/96 $4.00 

- Analog libraries in general are difficult to implement due to the fact that the primitive analog cells have more than one function with a number of nonidealities depending on the used technology.

The above points can be counteracted by:

- The trend in neural networks is towards modular and reconfigurable architectures which are well suited for the modeling concept.
- The number of CAD tools depending on the behavioral description of any system as its entry are those used at the behavioral simulation phase, the validation phase, and possible synthesis phase, whose requirements can be gathered in a single model.
- The level of accuracy of the model and the degree of closeness to the circuit level is mainly user defined.
- In order to capture all the circuit functions, the model could be formed of a single entity(describing the interface of the circuit) and a number of architectures(describing its function).

## 3. A Proposed Modeling Approach

In order to satisfy the four points previously mentioned, we propose a hierarchical methodology for the modeling of ANNs. The same point has been the subject of previous publications [16,17,18]. In this work, the model, written in analog hardware description language (HDL-A) supported by the electrical simulator ELDO[2], is composed of three hierarchical levels. This decomposition is important for providing flexibility in the choice of the desired degree of accuracy depending on the application. The first level is a block diagram representation of the circuit serving as a first order functionality check, in fact this level is approximately the same as the mathematical algorithm of the ANNs except for having an interface related to the physical circuit. The second level incorporates all the nonidealities of the circuit. This level is the most important one due to its adherence to the circuit technique utilized together with the fact that it is parameterized thus providing both fast and accurate simulation.

Improvement in the design cycle can be achieved using the second level, as the designer can first simulate the system using the model in order to optimize the different circuit parameters, and finally realize these parameters using the actual circuit [16].

The last level involves statistical modeling as it includes statistical information about the circuit in addition to errors arising from the variation in its characterizing parameters resulting from technological process. These errors could be found by applying Monte carlo analysis while varying the technological parameters and visualizing its effect on the circuit [17]. The last level should be provided for each used technology. Contradiction with the technology independence of the modeling concept is encountered in the last level but in the analog case, this effect is extremely important especially if the model will be further processed by any synthesis tool that selects the circuit satisfying the system requirements with the smallest acceptable error.

[2] ELDO is a trademark of ANACAD Electrical Engineering Software.

## 4. Results and discussion

In this section, we will focus on the first and the second hierarchical levels of the model, while the last level will be treated in future phases of work. Two primitive cells, the wide range Gilbert multiplier and the Guassian circuit utilized in the radial basis function neural network, are presented and their circuits and models are compared regarding simulation time and accuracy. A macromodel composed of the basic primitives (Transconductance element, I-V converter, and Hard-Limiter) and describing the operation of the 4 bit Hopfield A/D converter is given to emphasize the power of the modeling concept at the subsystem level.

### 4.1. Wide Range Gilbert Multiplier

The Gilbert multiplier circuit has been implemented in different ways and technologies [14,19]. The one published by Mead [14] (figure 1a) has the advantage of operating in the subthreshold regime thus consuming low power. The ideal output current of this circuit is given as

$$I_{out} = I_b \tanh\left(\frac{n(V_1 - V_2)}{2KT/q}\right) \tanh\left(\frac{n(V_3 - V_4)}{2KT/q}\right) \quad (1a)$$

Where $I_b$ is the bias current provided by the transistor Mb, and n is the subthreshold slope. In order to model this circuit using the second level, all the nonidealities should be added to the ideal behavior represented by equation(1a). The model parameters must be the external variables which the user senses independent of the internal circuit operation. Therefore instead of including the subthreshold slope, we include the dynamic range represented by the parameter ($V_{sat}$) which is directly related to the subthreshold slope.

The main nonidealities of this circuit are the offset voltages, the common mode rejection ratio (CMRR) of the

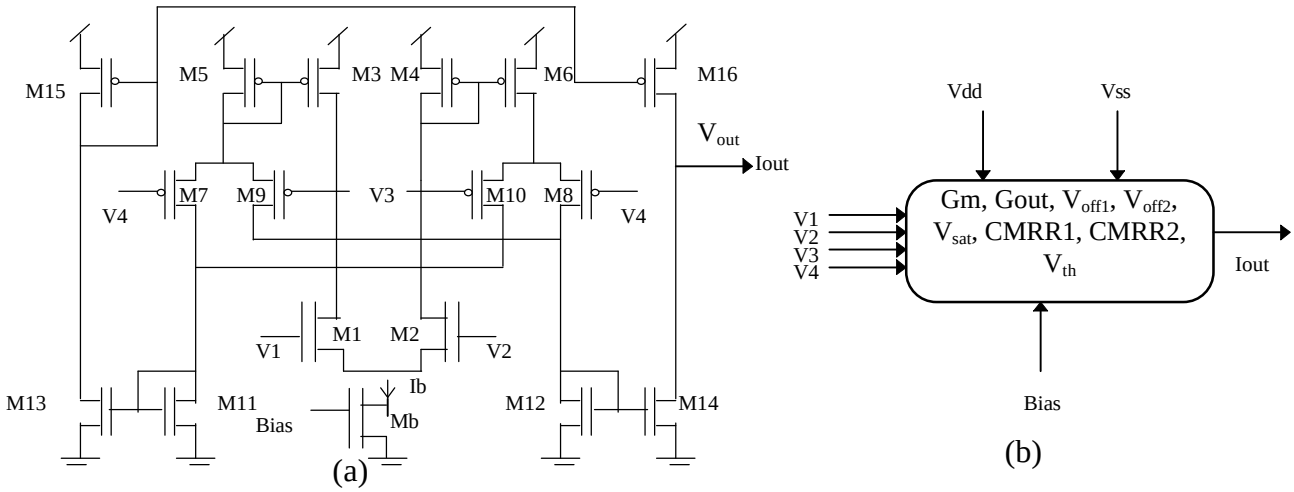
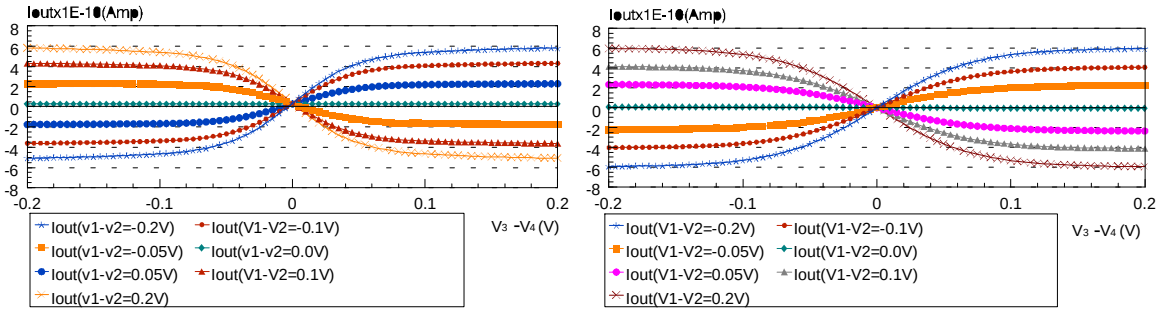



**Figure 1:Wide range Gilbert multiplier (a) Circuit diagram. (b) Model parameters and interface.**




(a) (b)

**Figure 2: Simulation results for the output current of the Gilbert multiplier. (a) Circuit. (b) model**

first and second differential pairs, and finally the output conductance (figure 1b). We can model all these effects by rewriting equation(1a) such that the output current in this case is given as

$$I_{out} = I_b \tanh\left(\frac{V_1 - V_2 + V_{off1}}{V_{sat}}\right)\tanh\left(\frac{V_3 - V_4 + V_{off2}}{V_{sat}}\right) + (Gm/CMRR1)\tanh\left(\frac{V_1 + V_2}{V_{sat}}\right) + (Gm/CMRR2)\tanh\left(\frac{V_3 + V_4}{V_{sat}}\right) - G_{out}V_{out} \quad (1b)$$

Since this circuit operates in the subthreshold region, the use of the parameter ($V_{th}$ ) was unavoidable to act as a restriction on the value of the bias voltage in order to guarantee proper circuit operation. This parameter is related to the threshold voltage for a given technology.

The Gilbert multiplier circuit is known to have a restriction on the values of the output voltage which may

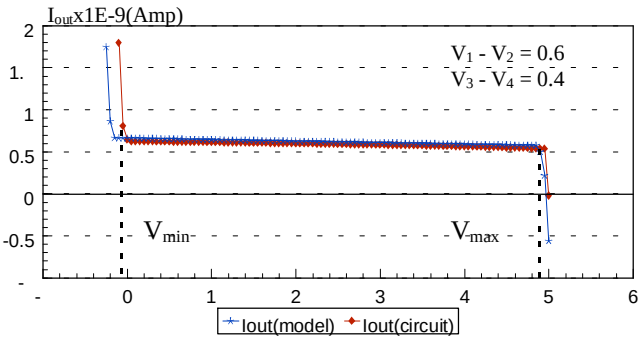



**Figure 3: nonidealities in Wide range Gilbert multiplier.**

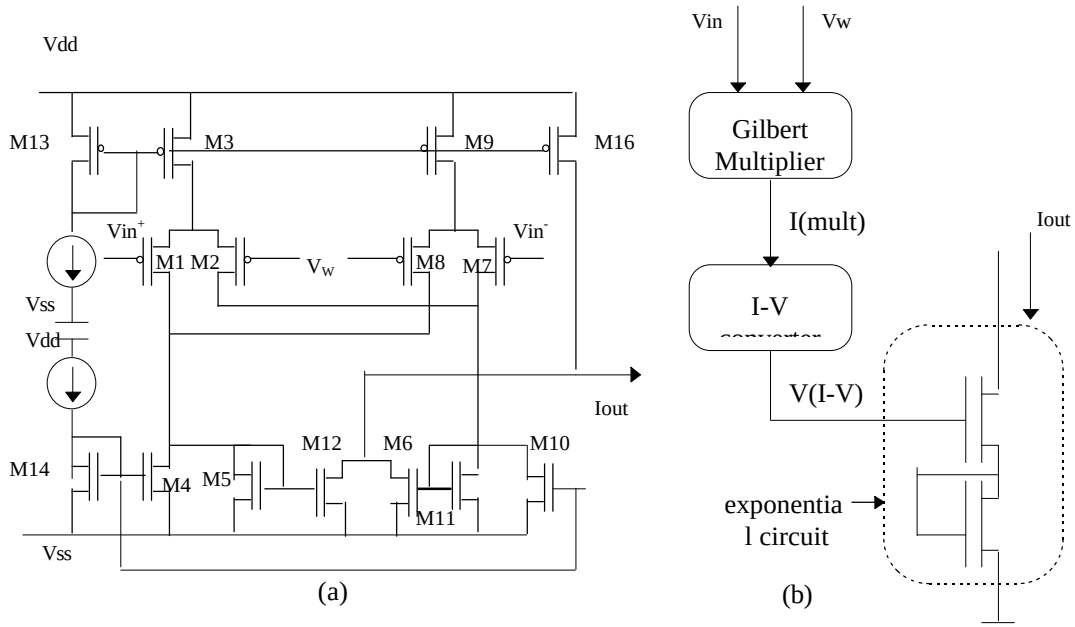



**Figure 4: The circuit implementation of the Guassian function. (a)Implementation_1[19] (b)Implemenation_2[20]**

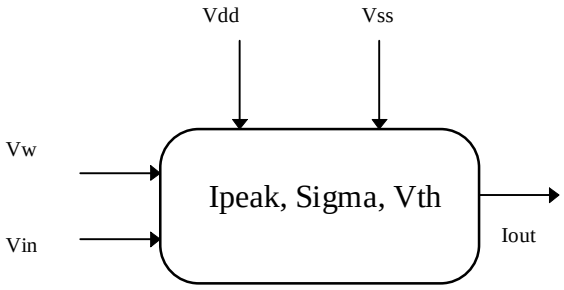



**Figure 5: The model parameters and the interface of the Guassian circuit**

be imposed on it when being employed in any system. For small values of output voltages below ($V_{min}$), the output current suffers a large increase while the opposite occurs for large values of voltage approaching that of the supply($V_{max}$)[14]. In order to model this limitation empirically , equation(1b) is multiplied by a function exponentially increasing below Vmin and exponentially decreasing above $V_{max}$

The simulation results (figure 2a,2b) for the model and the circuit represent the output current of the multiplier as a function of the second differential voltage ($V_3$-$V_4$) for different values of the first differential voltage ($V_1$-$V_2$). The results are identical except at zero differential voltage. This is mainly due to the way the model treats the transistor mismatch problem as it accounts for its effect just by including the two parameters $V_{off1}$ and $V_{off2}$ in the tanh functions of equation(1b) while the circuit behavior is much more complex.

Nonideal effects as the finite output conductance and the variation of currents at small and large output voltages are shown in figure 3. The equations describing the current for output voltages below ($V_{min}$) and above ($V_{max}$) are empirical because high accuracy is not required when operating outside the useful range of operation.

If we define the improvement in simulation time as the ratio between the difference in simulation time of the circuit and the model to the simulation time of the circuit under the same simulation conditions regarding the type of analysis and the number of points, then 58% improvement is achieved when performing DC analysis.

## 4.2. The Guassian Circuit

This circuit is used in pattern classifiers such as the radial basis function Neuural Networks [19]. The main equation describing the behavior of this circuit is

$$I_{out} = I_{peak} e^{\frac{-(V_{in}-V_w)^2}{2\sigma^2}} \quad (2)$$

Where $I_{peak}$ is the peak value of the current, $V_{in}$ is the input voltage, $V_w$ is the voltage describing the weight of the synapse, and $\sigma$ is the standard deviation.

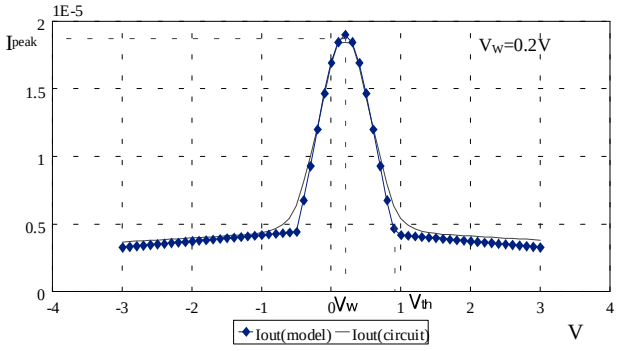
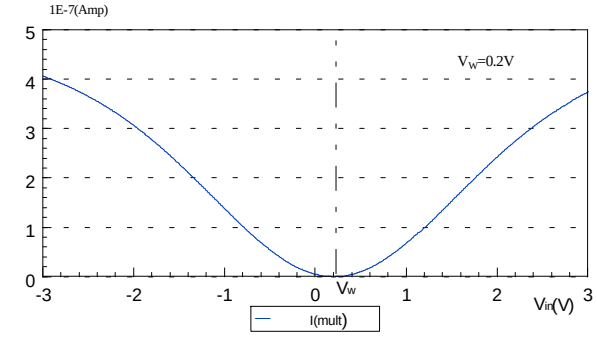
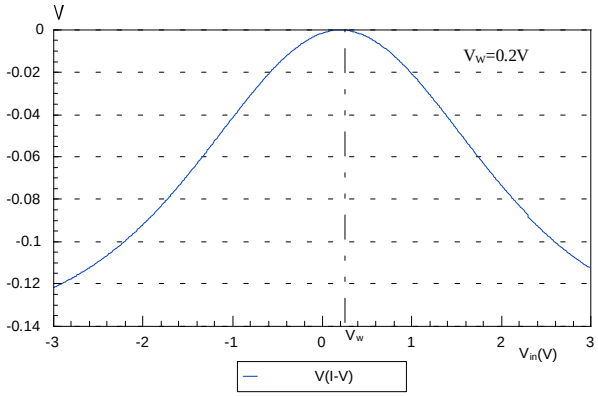


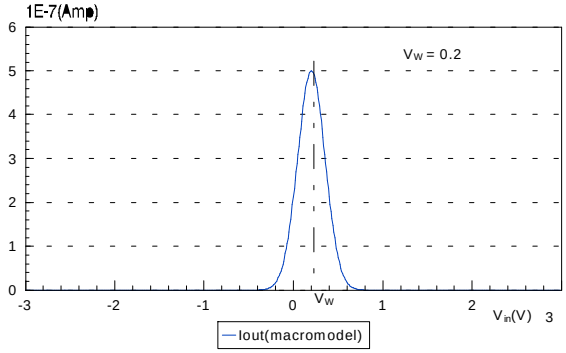

**Figure 6: Comparison between the output current resulting from the simulation using the circuit (Implementation_1) and the model of the Guassian function ( $V_w$ = 0.2V)**




(a)




(b)




(c)

**Figure 7: Simulation results for the macromodel describing the Guassian function (a) The output current of the multiplier circuit (b) the output voltage of the I-V converter (c) the output current of the exponential function**

Two possible circuit architectures have been published satisfying the Guassian function [19,20]. The circuit of figure 4a contains 15 transistors and it is modeled simply by expressing the current as in equation(2) but multiplying the right hand side of the equation with a certain function ***m(v)***, where this function is constant inside the operating guassian range and thus the output current is a pure Guassian function of the input voltage and is characterized by $I_{peak}$ and $\sigma$. Outside this range, the function ***m(v)*** tends to make the current changes its slope to acquire a linear behavior and it is characterized by the parameter $V_{th}$ which is the input voltage after which the current changes its slope(Refer to appendix I).

The second implementation is a macromodel formed of three primitives (Gilbert multiplier, I-V converter, and an exponential circuit) [14] and it has an identical performance to the first circuit in its first region thus it can be described by the same model or it can be considered as a macromodel formed of three simple models , where in this case it will be related to the structure of the circuit. The main parameters of these models are the transconductance of Gilbert multiplier Gm, the conversion gain of the I-V converter A, a constant current $I_o$ and a factor K, related to the parameter $\sigma$ in the first model, for the exponential function. The curves of figure 6 showing the output current from the Guassian circuit ( figure 4a) and our model (figure 5) are well matched, indicating the high accuracy of the model in addition to 95% improvement in the simulation time (DC analysis). The macromodel (figure 4b) shows larger simulation time as compared to that of the first model. Its simulation results are given in figure 7 showing the output from each stage.

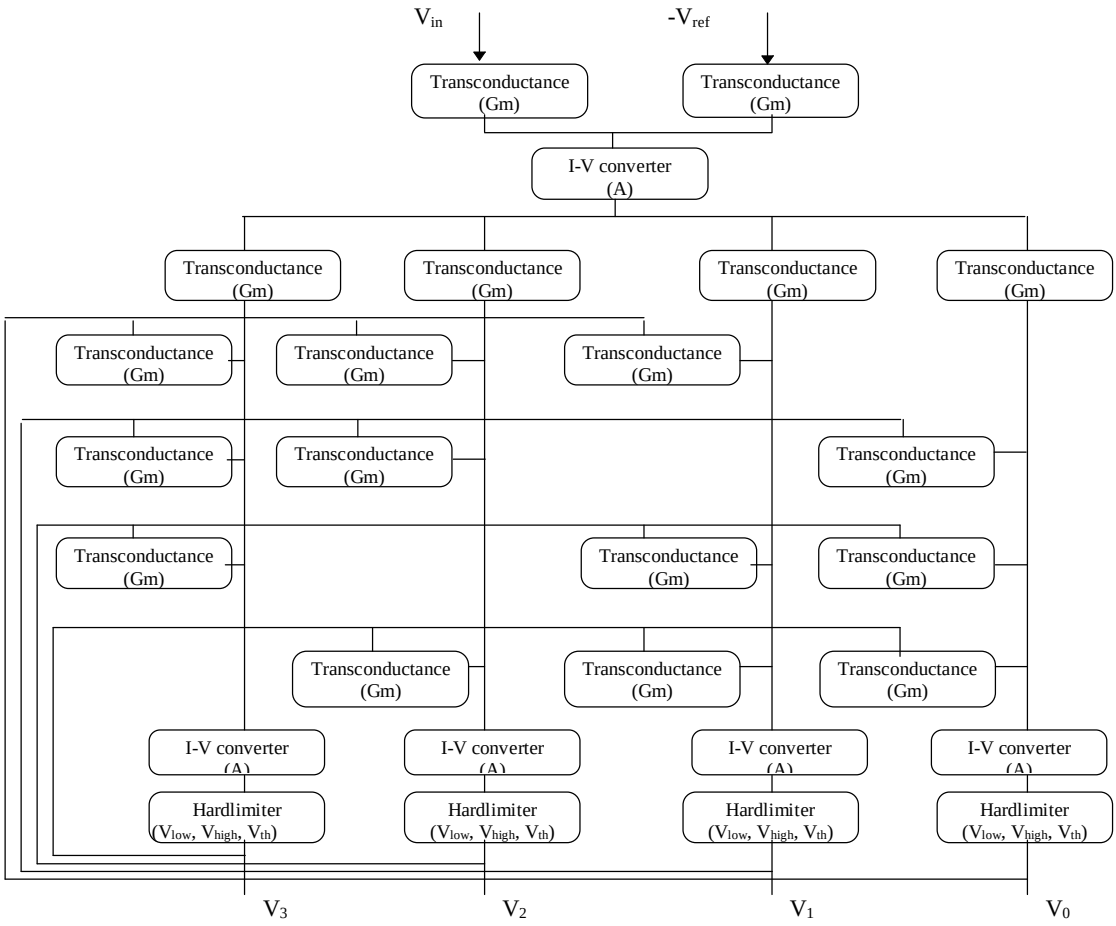



**Figure 8: Block diagram for the 4 bit Hopfield A/D converter[22]**

### 4.3. The Hopfield A/D converter

This circuit has gained its popularity in the mid-1980s thanks to its simplicity and regular structure. It has become now of limited importance due to the problems associated with its implementation and learning [3]. It is given as an example in this section just for the illustration of the model capabilities for high connectivity at the subsystem level. The 4 bit Hopfield A/D converter can be implemented in different ways [21,22]. The methodology used in [22] is well suited for the modeling process as the system is constructed of three major blocks (transconductance element, I-V converter, and hard-limiter). A block diagram for the circuit is shown in figure 8 with the three primitives described by their characterizing parameters. The parameters Gm and A have been described in the previous subsection, $V_{low}$ and $V_{high}$ are the two levels of the hard-limiter, and $V_{th}$ is threshold voltage after which the hard-limiter changes its output level. The models of the three primitive modules are simple, consisting of a single equation describing their function without including any nonidealities. This serves as a first order functionality check which can be beneficial for behavioral simulation . The number of transistors used in each circuit is small, thus the improvement in simulation time will not be appreciable(20% under DC analysis). The main improvement in this system when utilizing the models is the fact that the designer can change the parameters without going into the details of the circuit. Results provided by the model and the circuit are given in figure 9. The results are approximately the same and both show good agreement with the published results of Hopfield A/D converter[21].

## 5.Conclusion

In this paper a methodology is given for the analog behavioral modeling of ANN. Among the overall

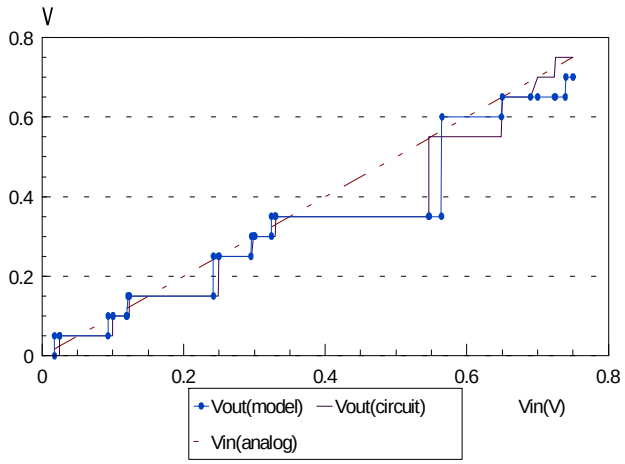



**Figure 9: Comparison between the simulation results of the circuit and the model of the 4 bit Hopfield A/D converter**

applications utilizing the analog VLSI technology, ANN seems to be the most suitable to be modeled and afterwards synthesized. Three examples were given, where in the first two, all the nonidealities which may be encountered in the actual circuit were modeled. Thus the model can be utilized by any optimizing or synthesis tool since the output voltage or current is given as a function in all the input signals and the effective parameters. Including these nonidealities tends to increase the simulation time and thus decrease the efficiency of the modeling process in accelerating simulation.

The final example illustrates the power of the modeling concept when being utilized as an ideal functionality check for a subsystem formed of a number of simple modules as the 4 bit Hopfield A/D converter .

The improvement in simulation time varies from one circuit to another regardless of its size because each model has itsown degree of complexity depending on its behavior rather than the size of its circuit

The simulation results provided by the models agree to a noticeable extent with that of the circuit. Analog libraries formed of these models can be very efficient in any possible future simulation, synthesis, or verification analog CAD tools.

## 6. Appendix I:The Architecture of the HDL-A model for the Guassian circuit

ARCHITECTURE *functional1* of *guassian* is
variable m : analog;
--A function to be multiplied by the ideal current to account for nonidealities effect [***m(v)***]
variable vth2 : analog;

BEGIN
RELATION

PROCEDURAL FOR INIT =>
Io := 10.0E-9;
sigma := 10.0E-1;
vth :=1.5;
K :=3.0;

PROCEDURAL FOR DC,TRANSIENT =>

vth2 := 2.0*in_weight.v - vth;
--The input weight signal is given the name $V_W$ in the text
--The input signal is given the name $V_{in}$ in the text

IF (in_signal.v > vth2) and (in_signal.v < vth) then
m := 1.0;

ELSE
m := exp(-k*((vth - in_weight.v)/sigma)*((vth - in_weight.v)/sigma))* exp(k*((in_signal.v - in_weight.v)/sigma)*((in_signal.v - in_weight.v)/sigma))*(1.0 -0.1*abs(in_signal.v));

END IF;

OUTP.I %= -m*Io* exp(-k*((in_signal.v - in_weight.v)/sigma)*((in_signal.v - in_weight.v)/sigma));

--The output current from the pin (outp) is given by its ideal equation multiplied by the function (m)

END RELATION;
END ARCHITECTURE *functional1*;

## 7.References


1. E. Liu, A. Vincentelli, “Behavioral representations for VCO and Detectors in Phase-Lock Systems,” *IEEE 1992 Custom Integrated Circuits Conference*.
2. M. Glesner and W. Pöchmüller, *Neurocomputers An overview of neural networks in VLSI*, CHAPMAN & HALL: London, 1994.
3. Jacek M. Zurada, *Introduction to Artificial Neural systems*, West Publishing company, 1992.

4. C. Mead and M. Ismail, eds., *Analog VLSI Implementation of Neural Systems*, Kluwer: Norwell, MA, 1990.
5. A. Cichocki and R. Unbehauen, *Neural Networks for Optimization and Signal Processing*, Wiley: New York, 1993.
6. J. Van der Spiegel, P. Mueller, D. Blackman, P. Chance, C. Donham, R. Etienne-Cummings, and P. Kinget, "An analog neural computer with modular architecture for real-time dynamic computations," *IEEE J. Solid-State Circuits*, Vol. SC-27, pp.82-92, 1992.
7. S. Gowda, B. Sheu, J. Choi, C. Hwang, J. Cable, "Design and Characterization of Analog VLSI Neural Network Modules," *IEEE J. Solid-State Circuits*, Vol. 28, pp.301-312, 1993.
8. J. Choi, S. Bang, B. Sheu, "A programmable Analog VLSI Neural Network Processor for Communication Receivers," *IEEE Trans. Neural Networks*, Vol. 4, pp.484-494, 1993.
9. A. Andreou, K. Boahen, P. Pouliquen, A. Pavasovic, R. Jenkins, K. Strohbehn, "Current-Mode Subthreshold MOS Circuits for Analog VLSI Neural Systems," *IEEE Trans. Neural Networks*, Vol. 2, pp.205-213, 1991.
10. B. Barranco, E. Sanchez-Sinencio, A. Rodriguez-Vazquez, J. Huertas, "A Modular T-Mode Design Approach for Analog Neural Network Hardware Implementations, " *IEEE J. Solid-State Circuits*, Vol. 27, pp.701-712, 1992.
11. F. Kub, K. Moon, I. Mack, F. Long, "Programmable Analog Vector-Matrix Multipliers," *IEEE J. Solid-State Circuits*, Vol. 25, pp.207-214, 1990.
12. P. Hollis, J. Paulos, "Artificial Neural Networks Using MOS Analog Multipliers," *IEEE J. Solid-State Circuits*, Vol. 25, pp.849-855, 1990.
13. N. Saxena, J. Clark, "A Four-Quadrant CMOS Analog Multiplier for Analog Neural Networks," *IEEE J. Solid-State Circuits*, Vol. 29, pp.746-749, 1994.
14. C. Mead, *Analog VLSI and Neural Systems*, Addison-Wesley:Reading, MA, 1989.
15. M. Ismail and J. Franca, eds., *Introduction to Analog VLSI Design Automation*, Kluwer: Norwell, MA, 1990.
16. A.M.Abdelatty, H.Haddara, and H.F.Ragaie, "Analog Behavioral Modeling of Artificial Neural Networks" *Proc. ICM'94*, pp. 140,143.
17. M. Elmasry, eds., *VLSI Artificial Neural Networks Engineering*, Kluwer: Norwell, MA, 1994.
18. E. Sanchez-Sinencio, "A Unified Enviroment for High-Performance Analog Signal Processing," *Proc. ICM'94*, pp. S49-S49.
19. B. Sheu and J. Choi, *Neural Information Processing and VLSI*, Kluwer: Norwell, MA, 1995.
20. S. Watkins, P. Chau, R. Tawel, "A radial basis function neurocomputer implemented with analog VLSI circuits," *Proc. IEEE/INNS Inter. Joint Conf. Neural Networks*, VOL. 2, pp. 607-612, 1992.
21. B. Kosko, *Neural Networks for Signal Processing*, Printice Hall, 1992.
22. M. Tan, " Synthesis of Artificial Neural Networks by Transconductances Only", *Analog Integrated Circuits and Signal Processing* 1, pp. 339-350, 1991.