

# Non-Refreshing Analog Neural Storage Tailored for On-Chip Learning

Bassem A. Alhalabi  
Computer Science & Eng. Dept.  
Florida Atlantic University  
Boca Raton, Florida 33431

Qutaibah Malluhi  
Computer Science Dept.  
Jackson State University  
Jackson, MS 39217

Rafic Ayoubi  
Faculty of Engineering  
University of Balamand  
P.O. Box 100, Tripoli, Lebanon

## Abstract

*In this research, we devised a new simple technique for statically holding analog weights, which does not require periodic refreshing. It further contains a mechanism to locally update the weights from the analog back-propagation signals for fast on-chip learning. In this circuit, the weight is stored as a 5-bit digital number, which controls the gates of five pass transistors allowing five binary-weighted (1,2,4,8,16) voltage references to integrate at a voltage adder. The output of the voltage adder is the analog weight. The 5-bit register is designed as an up/down counter so that every pulse on the up/down input will increase/decrease the weight by one level out of 32 possible levels. The learning circuit takes the analog graded error signal and generates two pulse streams for up/down counting depending on the sign of the error signal. The duration of the pulse stream is proportional to the magnitude of the error signal. This complete modular synaptic body (storage and learning technique) is appropriate for large scaleable analog VLSI neural networks because it handle recall and learning operations at the same speed with full parallelism.*

## Analog storage and refreshing

Synaptic weight storage has been the most challenging design issue in analog neural network paradigm especially if on-chip learning is required. In this case, the analog weight value (stored on a capacitor for example) must be made adjustable through some switching techniques (transistors) associated with the capacitor. This transistor association breeds an unpleasant leakage, which inevitably shortens the life of the weight storage, and

wherefore, refreshing mechanisms have evolved. Of course, the best storage to solve this problem is the digital static RAM that does not require refreshing hardware. However, if analog functional units are to be used for their superiority in speed and size, then all digital weights must be converted to analog. This conversion, in turn, necessitates that each synapse must be equipped with a DAC. Now, if learning is to be implemented in full parallelism, then an ADC must be provided for each weight to impose the analog graded update signal. The DAC/ADC pair solution for each synapse is rather expensive [4]. This trade off between digital storage with DAC/ADC and analog storage with refreshing brought up several schemes and techniques for weight storage and update handling [4].

The four basic schemes of hardware architectures are shown in Figure 1. Scheme A is all-analog including learning and refreshing mechanisms, An example is the hybrid system of [1]. Scheme B is analog except weight storage and update are digital, example in [6]. Scheme C is analog except refreshing which is done through RAM/ADC combination. Examples are in [2] and [5]. Scheme D uses analog operators and digital storage without refreshing nor learning, example [7]. The scheme of this paper is close to Scheme B except that the ADC and digital adder in the learning section are replaced by our pulsing mechanism.

## Static storage

The concept of non-refreshing storage is very simple and emerges from the basic idea of analog to digital conversion. The static digital storage in a form of a binary counter is utilized to hold a binary number, which is proportional to the actual value of the weight.

In this paper, we are using a 5-bit counter as shown in Figure 2. Although the same architecture can be easily implemented with more than 5 bits, simulations in previous literature have shown that 5 bits with 32 possible states provide sufficient accuracy [3].

The 5 bits which hold the normal binary positional weights, 1, 2, 4, 8, and 16, are connected to 5 pass transistors (T1, T2, T4, ..., T16) which are, in turn, connected to 5 binary weighted voltage supplies (V1, V2, V4, ..., V16) as shown in Figure 2. The summing operational amplifier SUM adds the different values of contributing power supplies and generates an output which is proportional to the binary value (0, 1, 2, 3, ..., 31). This voltage output is the actual analog weight. This methodology of static storage of analog weights has been utilized in one way or another in literature but none has been adapted for fully parallel on-chip learning.

## Learning technique

Learning on chip is the ability to dynamically modify the stored synaptic weights in accordance to the graded update errors, which are computed in the back-propagation procedure. The difficulty in analog paradigm is in imposing this analog error on the weights if they are statically stored in a digital form. In this research, however, we are storing the weight in a digital counter which facilitates for dynamically changing the weight value by simply pulsing the counter. Two issues need to be addressed here, the direction, up or down, and the magnitude, number of pulses. The direction of pulsing, up or down, depends on the sign of the graded update error signal, positive or negative respectively. The magnitude of update error signals is translated into multiple of pulses. To make a proportional relation between the number of pulses and the magnitude of the error signal, we used an analog technique as shown in Figure 3.

The computed weight update signal,  $\Delta w$ , is passed through 2 different circuitries to generate two pulses, PP for positive error and PN for negative error. See Figure 2 for schematics and Figure 3 for timing diagram. When  $\Delta w$  carries a positive signal, P1 goes high charging the associated RC circuit to a potential level proportional to the strength of the  $\Delta w$  signal. The PP inverted output is consequently an active-low pulse whose duration is proportional to magnitude of  $\Delta w$  signal. This PP pulse will then enable the NAND gate enabling the output UP to pulse the counter at a fixed rate, CLK signal, for a period of time proportional to the magnitude of  $\Delta w$  signal. At the same time, the DN output remains inactive.

In the other case, when  $\Delta w$  carries a negative signal, P2 goes high due to the inverting voltage follower, INVT. P2 then charges the associated RC circuit to a potential level proportional to the strength of the  $\Delta w$  signal. The PN inverted output is consequently an active low pulse whose duration is again proportional to magnitude of  $\Delta w$  signal. This PN pulse will then enable the lower NAND gate enabling the output DN to pulse the counter down for a period proportional to the magnitude of  $\Delta w$  signal. Here, the UP output remains inactive.

With this technique, the  $\Delta w$  signal, based on its sign, is translated into two exclusive trains of pulses, UP and DN, which will respectively, either increase or decrease the value of the weight stored in the counter. In either case, the amount of change is proportional to the length of the pulse train, which is proportional to the magnitude of the original  $\Delta w$  signal.

## Conclusion

In this paper, we devised a simple scheme of static storage for analog weights with the ability to modify their values in proportion to an analog feed back error without the need for ADC. This simple circuit can be easily replicated for each synapse body to achieve maximum parallelism.

## References

- [1] B.A. Alhalabi and M.A. Bayoumi. On-Chip Learning for Scalable Hybrid Neural Architecture. 1997 IEEE International Symposium on Circuits and Systems, Hong Kong, June 9-12, 1997.
- [2] B.E. Boser and E. Sackinger. An analog neural network processor with programmable topology. *IEEE Journal of Solid-State Circuits*, Vol.27, No.1, pp.67-81, January, 1992.
- [3] B.K. Dolenko and H.C. Card. Tolerance to analog hardware of on-chip learning in backpropagation networks. *IEEE Transactions on Neural Networks*, Vol.6, No.5, pp.1045-1052, September, 1995.
- [4] Y. Horio and S. Nakamura. Analog Memories for VLSI Neurocomputing. Chapter in "Artificial Neural Networks" ed. by Sanchez-Sinencio, IEEE Press, 1992.
- [5] S. Satyanarayana and Y.P. Tsividis. A reconfigurable vlsi neural network. *IEEE Journal of Solid-State Circuits*, Vol.26, No.12, pp.2017-2025, December, 1991.
- [6] T. Shima and T. Kimura. Neuro chips with on-chips back-propagation and/or hebbian learning. *IEEE Journal of Solid-State Circuits*, Vol.27, No.12, pp.1868-1876, December, 1992.
- [7] J.V. Spiegel and P. Mueller. An analog neural computer with modular architecture for real time dynamic computations. *IEEE Journal of Solid-State Circuits*, Vol.27, No.1, pp.82-92, January, 1992.

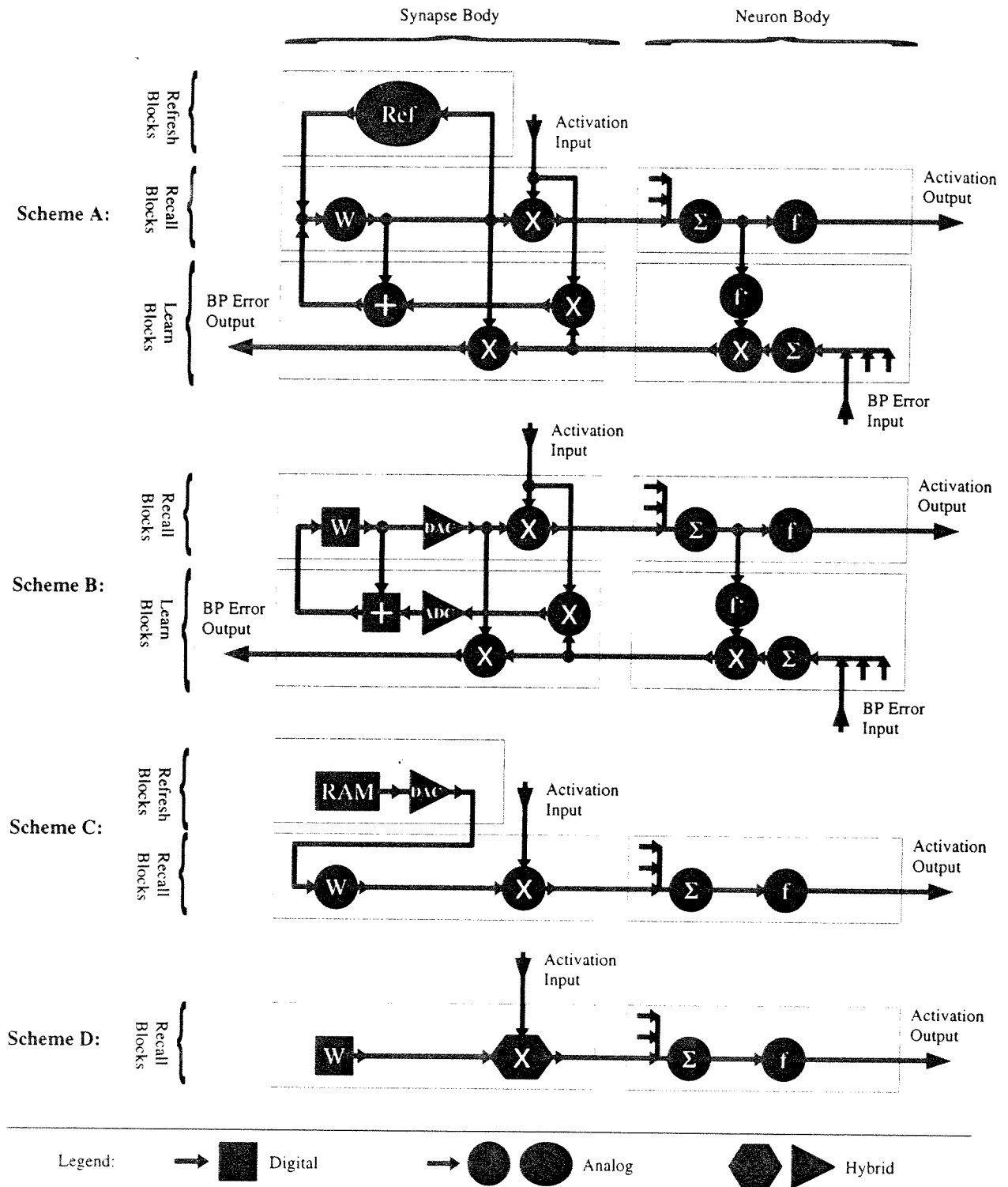


Figure 1: The Four Basic Schemes of Storage and Learning Mechanisms.

