CMOS Architecture of Synchronous Pulse-Coupled Neural Network and Its Application to Image Processing

Yasuhiro Ota

Image Information Products Hdqrs.
MINOLTA Co., Ltd.
Toyokawa, Aichi 442-8585 JAPAN
y-ota@tdc.minolta.co.jp

Bogdan M. Wilamowski

College of Engineering
University of Idaho
Boise, Idaho 83712 USA
wilam@ieee.org

Abstract

This paper presents a compact architecture for CMOS implementation of a PCNN and its application to image processing. A computational style described in this article mimics a biological neural network using pulse-stream signaling and analog summation and multiplication. Pulse-stream encoding technique utilizes pulse streams to carry information and control analog circuitry, while storing further analog information on the time axis. The structural form of the pulse-coupled neuron is presented first, then its application to image processing and the synchronization effect between neighboring neurons are demonstrated.

1. Introduction

In the vertebrate nervous system, communication between distant neurons is accomplished using encoded pulse streams [1],[2]. It was well into the 1930's before significant measurements of pulse-coded electrical activity in the brain had begun. Schmitt [3] devised a means of solving the equations proposed in theories of biological impulse propagation via vacuum tube circuits. In any event, the neuristor and its derivatives led to a large number of circuits being proposed for neural network realizations in mid 1960s through the mid 1970s. The idea of the neuristor is to abstract the five key axon properties of (i) threshold of excitation, (ii) refractory period, (iii) constant pulse-propagation velocity, (iv) pulse-shaping action during its propagation through the neuristor line, and (v) annihilation of pulses in case of their collision.

Pulse-stream encoding technique [4]-[8] uses pulse streams to carry information and control analog circuitry, while storing further analog information on the time axis. The firing rate of action potentials in biological neurons is roughly proportional to change in the original graded potential, which is categorized as *frequency modulation*.

Padgett, Werbos, and Kohonen [9] present overview of the use of PCNN in pattern recognition applications. PCNN models and their applications are also covered in more detail in several articles [10]-[13].

2. CMOS Architecture of PCNN

Inspired by biological models and the advantages of PCNN, a compact integrated circuit structure for a neuron with synaptic weight multiplication and summation is described in this section. The neuron circuit and its associated circuits function similar to a biological neuron with synaptic junctions. The presented neuron cell circuitry shown in Fig. 1 is an electronic analogy of a biological soma; i.e., it initiates reactions, with a given external stimulus, by generating a stream of electrical pulse waves. In this case, the external stimulus is current. The circuit structure is based on the current-driven simple neuron cells [5]-[7] and its voltage-driven circuit [8].

The presented neuron circuit in Fig. 1 functions as follows. The circuit has two capacitors, C_1 and C_2. The stored charge on capacitor C_1 corresponds to the charge of sodium ions (Na^+) accumulated on the external side of the *biological* neuron membrane, and the charge stored on C_2 corresponds to the potassium ions (K^+) accumulated inside the neuron cell [8]. The potential due to sodium ions changes at a faster rate than the potential due to potassium ions. Therefore, the time constant of the C_1 circuit is made smaller than that of the C_2 circuit. In a steady state, the MOS transistors (M_1-M_3) are cut off. As the potential on C_1 increases and exceeds the potential on C_2 by the threshold value of transistor M_1 at some point, then transistor M_1 change its state into active region of operation and further activates transistors M_2 and M_3 which form a current mirror. This positive feedback through transistors M_1, M_2 and M_3 is quickly terminated once capacitor C_2 is fully charged, and all the transistors become turned off. During the recovery process, known as *refractory period*, capacitor C_2 is slowly discharged by resistor R_2, and the neuron

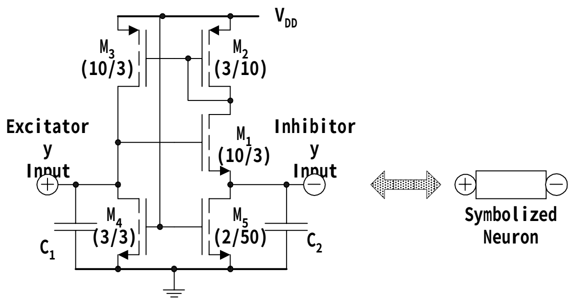

Fig. 1. CMOS circuit diagram of the presented neuron.

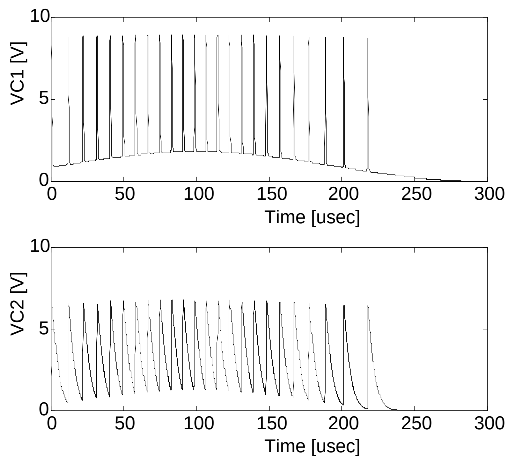

Fig. 2. Transient response of the neuron model simulated with SPICE. The neuron is excited with a shifted sinusoidal input. Notice that the discharge time on C_2 (bottom graph) is much slower than the discharge time on C_1 (top graph).

cell does not respond to any incoming excitations until the potential on C_1 exceeds the potential on C_2 by the threshold value of M_1. The transient response of the circuit of Fig. 1 for a shifted sinusoidal input excitation is illustrated in Fig. 2. Notice from Fig. 2 that the frequency of output pulses is proportional to the input excitation level (i.e., the highest firing frequency occurs at the peaks of the sinusoidal input signal); however, the maximum frequency of output oscillation is limited by this refractory period. One can observe this effect with an almost constant frequency for high input excitation level. ,

Notice that the presented circuit design has two input nodes in the neuron cell - one node at capacitor C_1 for an *excitatory* (positive) synaptic input, and the other at capacitor C_2 for an *inhibitory* (negative) synaptic input. Incoming input currents at the excitatory node are charging up capacitor C_1, yielding a positive effect on triggering transistor M_1. Incoming input currents at the inhibitory node are charging up capacitor C_2 that has a negative effect on triggering transistor M_1 by increasing its threshold value. In this scheme both excitatory and inhibitory synaptic weights are controlled, as in natural biological neural networks. By adjusting the resistance of the coupling resistors, the current, which flows through the axon, is controlled, yielding a corresponding rate of injecting charges into the input capacitor C_1. In the simplest case, NMOS transistors with their gates connected to positive power supply can be used to represent synaptic weights, as Fig. 3 illustrates.

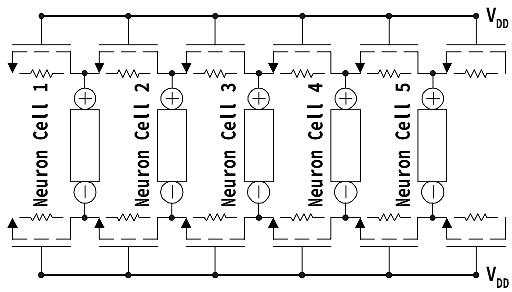

Fig. 3. Concept diagram of neuron cells connected in a chain to form an axon with coupling resistors. Each resistor (NMOS transistor) represents a synaptic weight.

3. Properties of PCNN and Its Model

The inputs of series of neural cells can be connected in a chain by simple coupling resistors to form an axon as shown in Fig. 3. In this section, the following basic properties of axons are emphasized and demonstrated:

1. threshold point of a pulse firing action
2. pulse shaping action during its propagation
3. refractory period
4. constant pulse-propagation velocity
5. annihilation of pulses in case of their collision

Several researchers have developed a neuromorphic delay lines [7],[12] demonstrating some of the above properties. The biggest merit of the presented design here, compared to other existing designs, is the small number of CMOS transistors with a compact design, allowing less power consumption and reduced silicon area. These five axon properties with the presented circuitry are discussed in below.

1. Threshold Point:

The first property, threshold point of a pulse firing action, was already demonstrated in Fig. 2 in the previous section. Recall that a pulse is fired when the potential on C_1 exceeds the potential on C_2 by the threshold value of transistor M_1.

2. Pulse Shaping Action:

Without a pulse shaping action, traveling pulses could be seriously attenuated and dispersed throughout the transmission. Thus, axons that have similar membrane structures should be able to regenerate the shape of transmitting pulses. Incoming pulses are regenerated and shaped as they transmit along the axon. Fig. 4 illustrates the pulse shaping action for a square input pulse. If an input pulse is too narrow, it will be annihilated. The shaping of the propagated pulse through the axon depends on the time constant of the output capacitor circuit

3. Refractory Period:

When the axon circuit is excited with a series of incoming pulses, those pulses can be transmitted through the axon if the incoming pulses are widely separated. On the other hand, some incoming pulses are skipped and not transmitted when the time interval between the incoming pulses is small. This property is demonstrated in Fig. 5. The refractory period of the delay line is caused by significant increase of the threshold voltage at the transistor M_1 after pulse firing.

4. Constant Pulse-Propagation Velocity:

Fig. 4 also demonstrates the property of constant pulse-propagation velocity. Neuron Cell 1 of Fig. 3 is initially stimulated with a voltage input in this simulation. One can observe that the resulting output pulse from Neuron Cell 1 activates Neuron Cell 2 and is seen to propagate at a constant velocity from left to right toward Neuron Cell 5. The extended refractory period of the excited pulses prevents the output pulses of neighboring units from reactivating a previous neural cell and insures that a single pulse is propagated.

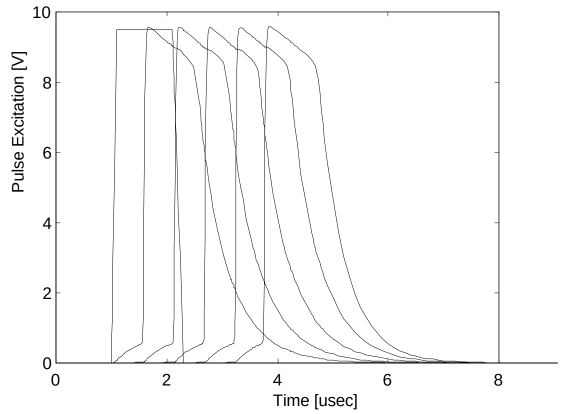

Fig. 4. SPICE simulated transient response to demonstrate the pulse shaping action and constant pulse propagation velocity. The square-wave pulse is applied to the input.

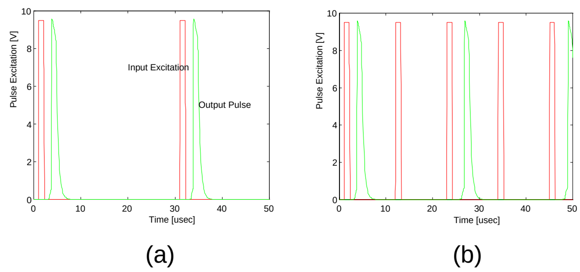

Fig. 5. Demonstration of effect of the refractory period. (a) With a large time interval between input excitation, all excitations are transmitted through the axon. (b) With a small time interval between input excitations, some of them are not transmitted due to the refractory period.

5. Mutual Pulse Annihilation:

Annihilation of pulses in case of their collision when pulses are propagating from opposite directions is a consequence of the existence of the refractory period, which causes pulse attenuation. By simultaneously stimulating Neuron Cell 1 and Neuron Cell 5 in Fig. 3, two analog pulses will collide at Neuron Cell 3 and annihilate each other since both Neuron Cell 2 and Neuron Cell 4 will both be in refractory period when Neuron Cell 3 fires a pulse.

4. Image Processing with the PCNN

Biological vision processing appears to require a number of parallel signals, containing color and intensity information, edge enhancement information which is provided by lateral inhibition between adjacent neurons. PCNN appear to manage combining such a wide variety of information into a coherent process. Eckhorn *et al*. [10] devised an integrate-and-fire neuron model based upon their studies of the visual cortices of cats. The Eckhorn dynamic model represents a visual neuron as a multi-input element with a single output. A leaky integrate-to-fire pulse generator [8], unlike the modulatory Eckhorn linking field, shows that its noise-smoothing capability is sperior to median filtering and average-filter smoothing. Most dynamic models of the neural computation suggest that *synchronization* between regions is the primary information carrier [8]. Johnson [11] states that the time signals are unique, object-specific and roughly invariant time signature for their corresponding input spatial image or distribution.

There is normally one-to-one correspondence between the neurons in the network and the pixels in the image. Therefore, the neuron $N_{i,j}$ in the network corresponds to the pixel $P_{i,j}$ in the image, and vice versa. The key features of image processing with the proposed pulse-coupled neurons are as follows. Suppose that the intensity range of the input image is mapped within [X_{min}, X_{max}]. The neurons with intensity X_{max} fire pulses naturally at the highest frequency. This intensity range is referred to as the *capture range* of $N_{i,j}$ with respect to the group of neurons pulsing at certain time. Then the number of pulses in certain time period is used to extract an image from the time signal back into the corresponding image signal. This image extraction depends on the strength of neuron couplings in the network. This coupling strength, expressed by *neuron coupling coefficient* β, of every neuron in the PCNN has the same value. The following example demonstrates image filtering (de-noising) and edge smoothing with the presented PCNN.

In general, the intensity of a noisy pixel is significantly different from the intensity of the neighboring pixels. Therefore, the intensity of the noisy pixel is unlikely to lie within its capture range with respect to its neighboring neurons. As a result, the noisy neuron is neither captured by its pulsing neighbors nor captures them. Fig. 6(a) shows the original image without any noise, and Fig. 6(b) is corrupted with additive random noise. Figs. 6(c) and 6(d) show the results of filtering the noisy image using the PCNN-based filtering technique with a neuron coupling strength of β=0.01 and β=0.05, respectively. First, note that PCNN-based filtering does not blur, dilate, or erode the edges of the image. If conventional median filtering or average filtering technique were to be used, the filtered image would have rounded corners and/or blurred the entire image [8]. On the other hand, the PCNN filtering retains the narrow features in the image significantly well.

It is also interesting to observe from Figs. 6(c) and 6(d) that the extent of the capture range increases as the neuron coupling coefficient β increases. In other words, a group of pulsing neurons can capture a neuron with considerably lower intensity if a sufficiently large neuron-coupling coefficient is used.

Hence, the idea of adjusting the neuron coupling coefficients is extended to the edge smoothing. That is, for some images where existence of sharp edges is not preferred or yields unnatural appearance to human eyes, an inclusion of edge smoothing in the original image is often preferred. In this case, a higher neuron coupling coefficient should be assigned. As can be seen from Fig. 6(d), which is resulted from assigning a higher coupling coefficient than one assigned in Fig. 6(c), the edges of the image feature is somewhat smoothed out from the background (base) intensity level. Furthermore, for a complex edge smoothing or edge enhancement and other feature extraction, multiple neuron-coupling coefficients can be assigned in a single PCNN architecture.

As can be seen from this image processing example, the capture phenomenon and the capture range play an important role in the image processing applications. That is, the neuron coupling strength determines the neuron synchronization effect. Recall that the coupling between neurons provides a global connection for all sub-regions in the image, and the neuron coupling enforces global synchrony between local regions. The effect of neuron synchronization is demonstrated by varying the neuron coupling coefficient and is seen in Fig. 7.

As the simulated result in Fig. 7 illustrates, the synchronization effect is stronger with a higher coupling coefficient, and vice versa. In fact, there is almost no synchronization between two neighboring neurons if β=0 in Fig. 7(a). On the other hand, a complete synchronization between two neighboring neurons is observed in case of β=0.95 in Fig. 7(c).

Furthermore, the existence of synchronization between neighboring neurons depends upon how these neurons are coupled together.

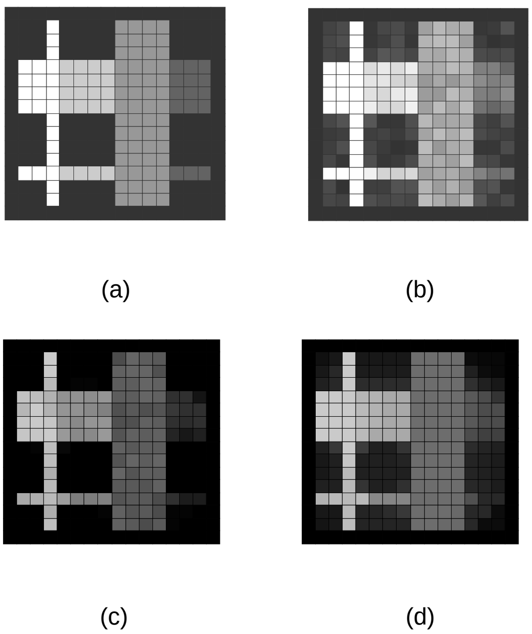

Fig. 6. An example of image filtering and extraction with PCNN. The original image (a) is added with some random noise (b). The results of PCNN-based image filtering with neuron coupling coefficient β=0.01 (c) and β=0.05 (d).

Fig. 8 shows the concept diagram of three basic types of neuron couplings. If there is no coupling between neighboring neurons, as seen in Fig. 8(a), each neuron oscillates at its own frequency that is proportional to the corresponding input excitation level. The second type, cross-coupled neurons as seen in Fig. 8(b), exhibits an averaging property among the neighboring neurons. Therefore, the output pulse excitation is determined by the weighted sum of the prime and neighboring neuron excitation levels. The third type is straight-coupled neurons, as shown in Fig. 8(c), which exhibits a synchronization effect between neurons.

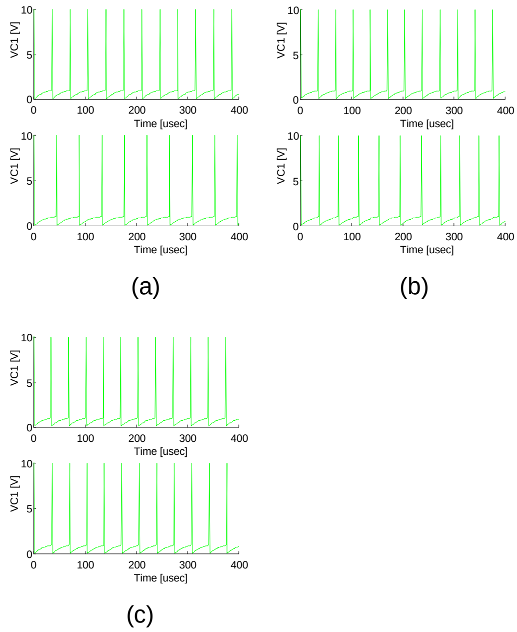

Fig. 7. Demonstration of the neuron synchronization effect on neighboring neurons. (a) β=0; (b) β=0.03; (c) β=0.95.

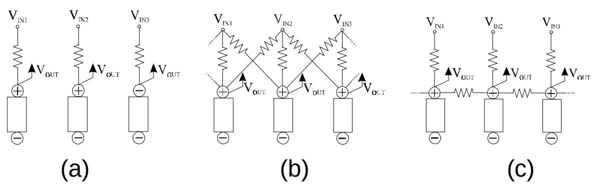

Fig. 8. Three basic types of neuron couplings. (a) No coupling between neurons. (b) Cross-coupled neurons. (c) Straight-coupled neurons which exhibit the synchronization effect and can be observed in PCNN.

5. Conclusions

In this paper, a CMOS hardware architecture that implements a pulse coupled neural network is described. The hardware design uses a leaky integrate-to-fire pulse generator, adaptive synaptic coupling, and dendritic delay-line propagation. The presented neuron circuit has its merits in hardware implementation owing to its simple structure, small size, and high speed of operation, yet achieving all the basic properties of natural biological neurons: (1) threshold-based firing (settable from external controls), (2) pulse shaping action and delay due to axonal propagation, (3) refractory period (i.e., showing a nonlinear sigmoidal characteristic), and (4) characteristic autowave behavior of mutual pulse annihilation. Another important feature of the proposed design is that the circuitry is robust to additive noise. Excitatory and inhibitory synaptic inputs are applied to the two capacitors (two input nodes), C_1 and C_2, respectively. The neuron cell circuitry which has been described here exhibits functional similarities to natural biological neurons.

The resulting PCNN architecture provides a practical near-term implementation leading to greatly enhanced image segmentation capability. It has also been shown that the PCNN has a unique synchronization property, and the capture phenomenon and the capture range play an important role in the image processing applications. That is, the neuron coupling strength determines the neuron synchronization effect.

References

[1] J. A. Anderson, *An Introduction to Neural Networks*, The MIT Press, Cambridge, MA, 1995.

[2] S. Deutsch and A. Deutsch, *Understanding the Nervous System: An Engineering Perspective*, IEEE Press, Piscataway, NJ, 1993.

[3] O. H. Schmitt, "Mechanical Solution of the Equations of Nerve Impulse Propagation," *Proc. American Physiological Society, 49th Annual Meeting*, Memphis, TN, April, 1937.

[4] G. Moon, M. E. Zaghloul, R. W. Newcomb, "VLSI Implementation of Synaptic Weighting and Summing in Pulse Coded Neural-Type Cells," *IEEE Trans. Neural Networks*, Vol. 3, No. 3, 394-403, May 1992.

[5] B. M. Wilamowski, Yasuhiro Ota, and Mary Lou Padgett, "CMOS Realization of a Pulse-Stream Artificial Neural Network using a Current Controlled Oscillator," *World Congress of Neural Networks*, San Diego, CA, September 1996, pp. 1317-1320.

[6] B. M. Wilamowski, Mary Lou Padgett, and Richard C. Jaeger, "Pulse-Coupled Neurons for Image Filtering," *World Congress of Neural Networks,* San Diego, CA, September 15-20, 1996, pp. 851-854.

[7] B. M. Wilamowski, Richard C. Jaeger, Mary Lou Padgett, and Lawerance J. Myers, "CMOS Implementation of a Pulse-Coupled Neuron Cell," *IEEE International Conference on Neural Networks*, Washington, DC, June 3-6, 1996, pp. 986-990.

[8] Y. Ota and B. M. Wilamowski, "Analog Implementation of Pulse-Coupled Neural Networks," *IEEE Trans. Neural Networks*, Vol. 10, No. 3, pp. 539-544, May 1999.

[9] M. L. Padgett, P. J. Werbos, and T. Kohonen, *Strategies and Tactics for the Application of Neural Networks in Industrial Electronics*, CRC Handbook for Industrial Electronics, J. D. Irwin Ed., CRC Press and IEEE Press, pp. 835-857, 1997.

[10] R. Eckhorn, H. J. Reitboeck, M. Arndt, and P. Dicke, "Feature Linking via Synchronization among Distributed Assemblies: Simulations of Results from Cat Visual Cortex," *Neural Computation*, Vol. 2, pp. 293-307, 1990.

[11] J. L. Johnson, "Pulse-Coupled Neural Networks," *Proc. Adaptive Computing: Mathematics, Electronics, and Optics*, S.-S. Chen and H. J. Caulfield Eds., Orlando, FL, Vol. CR55, pp. 47-76, April, 1994.

[12] W. Yang, "Neuromorphic CMOS Circuitry for Active Bidirectional Delay Lines," *Proc. IEEE International Symposium on Circuit and Systems (ISCAS)*, Atlanta, GA, Vol. 3, pp. 473-477, May, 1996.

[13] J. L. Johnson and M. L. Padgett, "OCNN Models and Applications," *IEEE Trans. Neural Networks*, Vol. 10, No. 3, pp. 480-498, May 1999.