A Field Programmable Neural Array

Ethan Farquhar*, Christal Gordon†, and Paul Hasler‡
farquhar@ece.utk.edu, christal_gordon@ncsu.edu, phasler@ece.gatech.edu
*Department of Electrical and Computer Engineering University of Tennessee, Knoxville, TN 37996
†Department of Electrical and Computer Engineering North Carolina State University, Raleigh, NC 27695
‡Department of Electrical and Computer Engineering Georgia Institute of Technology Atlanta, GA 30332

***Abstract*— An analog circuit capable of accurately emulating large complex cells, or multiple less complex ones is described. This circuit is termed the FPNA or the Field Programmable Neural Array. It is analogous to the more familiar FPGA, but is composed of biologically relevant circuit components including active channels, dendrites, and synapses. Taking each of these circuit models, and adding a routing structure capable of routing outputs from cells (or external inputs) to any individual synapse at any node yields a device which is capable of emulating complex biological circuits. This circuit will open doors to investigating what particular types of computation individual cells are performing, as well as small networks simpler cells.**

I. Overview

The efficacy of computing has been greatly increased by the concept of reconfigurability. Early electronic machines were capable of performing one task. They were also very expensive design and build, and what might be considered a simple change in the design today, would have proved a significant change then. True hardware reconfigurability is a relatively new phenomenon, but the ubiquitousness of microprocessors is due to the fact that software can be written for them that suits the application desired by the user. This can be thought of as a type of reconfigurability. While the hardware does not change, the software makes the generalized hardware do a specific task. The software can be easily rewritten to make the same physical processor do a completely different task.

Frequently, specialized hardware can perform a given computation more efficiently than general hardware that is running software (i.e. popular math co-processors). Reconfigurable hardware can significantly reduce the design cycle and allows for fast, low-cost systems. Recent advances in the field of hardware reconfigurability, spearheaded by the popularity of EPROMs, PLDs and FPGAs, have provided users with a platform whereby they can develop specialized hardware from generic building blocks. This is now a robust field, with many commercially available products. However, FPGA's operate mostly within the digital domain.

Field-programmable analog arrays (FPAAs) are just beginning to grow as a field. FPAAs allow the user to connect analog circuit blocks much like FPGAs allow a user to connect digital circuit blocks. Example blocks include amplifiers, multipliers, and single transistors [1]. These analog blocks can be arranged to form more complex circuits, such as matrix multipliers and filters in a manner that is similar to building adders and processors with FPGAs. There are some commercial products available, although the field of FPAAs is not as robust as its digital counterpart.

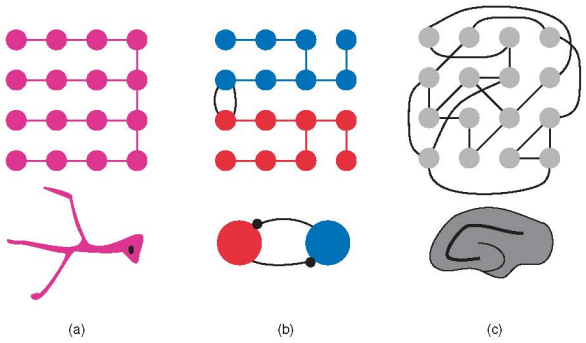

Fig. 1. Overview of concept. This figure is meant merely to illustrate types of networks that could possible be built using the FPNA.(a) One complex cell. (b) Two less complex cells forming a simple central pattern generator (half-coupled oscillator). (c) Many simple cells connected in a complex network.

This work seeks to introduce a new specialized FPAA, called the field-programmable neural array (FPNA). This device uses analog circuits as building blocks which have been designed to be biologically relevant. The blocks are connected to each other, but maintain a large degree of reconfigurability allowing future users to quickly design and experiment with neuro-inspired and neuro-mimetic systems. Similar to the way that DSPs are specialized microprocessors, FPNAs are specialized reconfigurable analog processors. Like DSPs, FPNAs are a tool with which users can create more powerful applications than could be realized with a more generalized device. Figure 1 is meant as an illustrative example of possible applications that the FPNA could be used to investigate. Since the amount of hardware on the chip is fixed, but the connectivity is not, the FPNA can model single complex neurons (those with large dendritic arborizations) such as Fig. 1 (a) or complex networks of simple neurons Fig. 1 (c), or a system with moderate complexity in both the computational unit and network connectivity. Note that the complexity of individual neurons in the network must be reduced proportional to the number of actual neurons being modeled in the network.

Some work has been previously done in the field of reconfigurable neural arrays [2], [3]. However, in both of these cases the neuron elements are of a fixed complexity. They do not have the ability to implement arbitrary dendritic

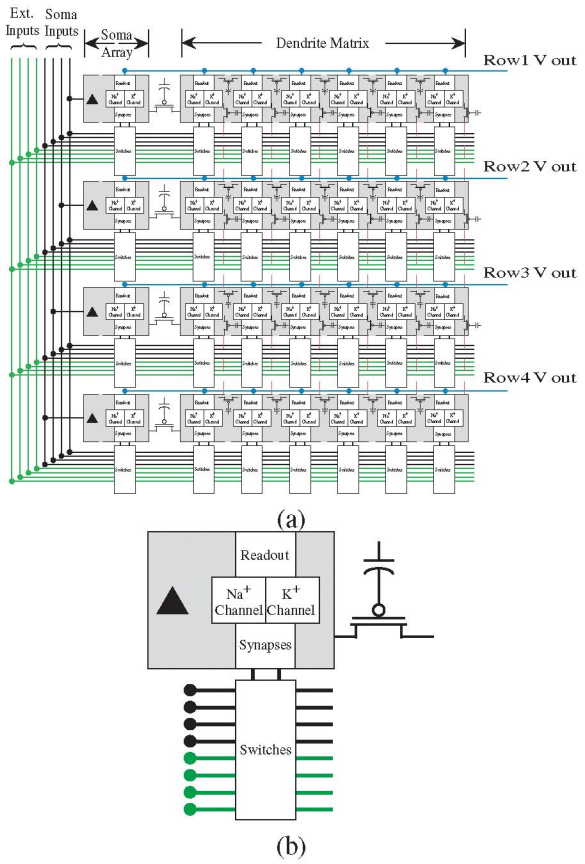

Fig. 2. (a) Schematic of the Field Programmable Neural Array (FPNA). Each node of the matrix is identical, with the exception of the triangle wave generator on the output of the soma. Each node has some circuitry for readout, 1 Na^+ channel, 1 K^+ channel, 1 inhibitory synapse, and 1 excitatory synapse. The switches are analog switches which allow for the off chip inputs and the outputs of the somas to be directed to any of the synapses. Each node of the dendrite matrix is connected to its nearest neighbor in 2 dimensions. These connections can be virtually turned off or connected to reconfigure the dendritic arborizations. (b) Shows a blowup of a single cell including the dendrite connection to the right (omitted are the dendrite connections to the left and the vertical connections).

arborizations. The reconfigurable nature of these circuits is in where the inputs/outputs go. Also, both examples utilize integrate and fire (or conceptually similar) neurons. Integrate and fire neurons are extremely useful for certain types of investigations, but they are not biologically relevant circuits. The authors acknowledge that the implementation of biologically relevant circuits is not necessary for every application, but one could very easily implement the FPNA described herein with integrate and fire neruons.

II. FPNA

The concept of an FPNA does not necessitate the following architecture or choice of building blocks. The architecture chosen was decided on because it allowed for maximum flexibility, and the building blocks that will be described were chosen because they are an area of active research for each of the authors.

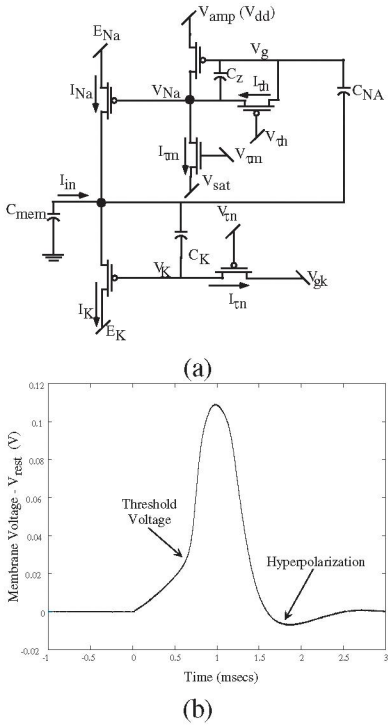

Fig. 3. (a) The Na^+ Channel and the K^+ channel. The currents generated by these circuits interact with each other on the membrane capacitor (C_{mem}) to generate an action potential. (b) An action potential generated by these active channels. The action potential is a stereotyped waveform present in the neurons of all species of animals.

A. Architecture

As seen in Figure 2, there is a full-crossbar connected matrix of elements. Thus, the output of any element can be connected to any other element. Every node consists of synapses, ion channels, dendritic sections, and readout circuitry. The latest version has two ion channels (1 Na^+ and 1 K^+), two synapses (1 excitatory and 1 inhibitory), two dendritic sections (horizontal connection and vertical connection), and circuitry for reading output voltages. It is important to note that any of these parameters can be adjusted in future versions. For instance, channel work is being done on various Calcium channels. In the future, these could be incorporated to add to functionality to this system. The authors also acknowledge that the implementation of a full-crossbar switch matrix may not be necessary (or desirable) for future versions. This switch implementation gives maximum routing possibilities, but does not scale well with increasing numbers of output nodes. These numbers were chosen to fit the following criteria in order of importance:

1) at least enough ion channels to create an action potential (Na^+ and K^+)
2) at least two synapses, preferably one excitatory and one inhibitory
3) pitch-matched to the smallest area possible
4) dendritic connections to connect in both dimensions of the matrix

5) at least one circuit to read voltage output

Within this set of criteria, there are many variables. Described here are the building blocks that were chosen for implementation.

B. Building Blocks

1) Active Channels: It has been shown that it is possible to build compact circuits which can emulate the dynamics of real channels [4]. As is the case with this previous work, just two types of channels have been implemented. A sodium (Na^+) behaves like a bandpass filter with very fast time constants (in biological terms). The potassium (K^+) channel is a lowpass filter, with a slower time constant. The Na^+ channel generates the very fast rise in voltage that can be seen in Fig. 3 (b). The K^+ channel works in the opposite direction of the the Na^+ channel, and is therefore responsible for returning the cell to rest. The Nobel prize winning work of the neuroscientists Hodgkin and Huxley first described the behavior of these two channel types. However, their circuit model (and their equations) made use of linearization that is not present in reality.

It turns out that the fundamental forces causing ion flow in biology are the same fundamental forces causing electron flow in a subthreshold MOSFET (at least at the macro-transport level). Using this relationship, the circuits seen in Fig. 3 (a) were developed. Using this compact circuit, action potentials (such as Fig. 3 (b) can be generated. This is important because the drive of this research has been to create circuits which are biologically relevant. Being able to generate the waveforms seen in biology with a circuit that utilizes the same transport phenomenon is paramount.

2) Electronic Synapse: The key to linking the blocks is synapses which not only provide biological output characteristics, but also can be tuned to provide a specific synaptic weight within a small footprint. The synapses used here, as in biology, are the smallest circuit of the neuronal block. Each synapse is a floating-gate pFET, which can be programmed to have a specific weight. This modification can be done through straight programming or adaptation. Programming, like supervised learning, entails knowing the desired synaptic weight. Programming algorithms allow for reaching synaptic weight targets within 10 μs or accuracy to the level of 100 fA [5], [6]. Adaptation, like unsupervised learning, requires the system to determine the proper weight by updating the weights in a manner that is dependent on the system inputs. These floating-gate synapses have been shown to have weight updates consistent with long-term potentiation and long-term depression as found in biology [7], [8].

Although there are many synapse types, we have chosen to use simple excitatory and inhibitory synapses to prove our concept. These two synapse types allow for networks with connections that augment or decrease a particular response. For instance a neuron that normally spikes at a 1 kHz rate may spike more quickly after an excitatory input, while it will spike more slowly after receiving an inhibitory input. Furthermore, the combination of these two synapse types allow

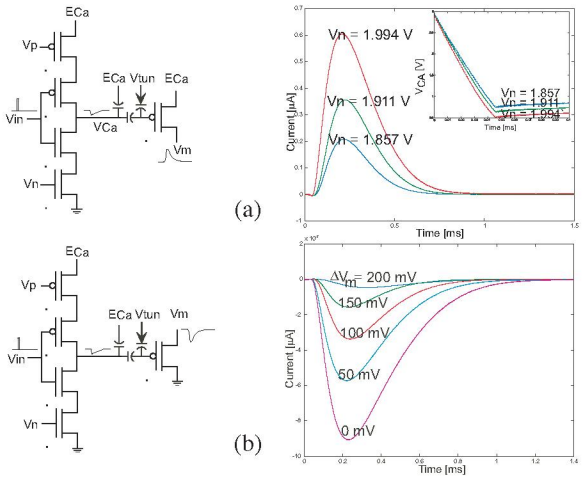

Fig. 4. (a) The excitatory synapse. The left side of the circuit behaves as a current limited inverter (a triangle wave generator). The output of this inverter is used to drive the input of a floating-gate transistor. The charge stored on the floating-gate can be modified to emulate the weight of the synapse. Current through the floating-gate transistor is exponentially related to the voltage on the gate. Data from this circuit illustrating that we can change the shape of the waveform can be seen on the right. (b) The inhibitory synapse. Notice that the circuit is the same as the excitatory synapse save for which node the current is being measured out of. Data from this circuit illustrating that we can achieve the same results as biological voltage clamp experiments can also be seen on the right.

for a biphasic output, which is useful for modified learning rules and prosthetic applications [9].

Both synapse types are similar in structure, as illustrated in Fig. 4. Both receive inputs from the soma and convert them to a post-synaptic potential (PSP). The PSP outputs for each synapse type precisely emulate the biological electrical response of a living neuron [10]. By modifying the synaptic weight and the biases for the soma, we can achieve PSP outputs that can be used to interface with living neurons [11].

3) Dendrite Matrix: Dendrites are frequently thought of as the inputs to a particular cell. However this distinction should belong to the synapses. Dendrites are the computation unit of the cell. Ions in a dendrite are able to diffuse either across the membrane, or axially along the length of the dendrite. Since diffusion is the macro-transport method of ion flow here, sub-threshold MOSFET transistors are being used to model the conductances seen along and across the membranes. The resulting single dimensional circuit looks very similar to the diffuser circuit described in [12]. However, a single diffuser circuit is unable to handle any of the branching seen in real dendritic arbors. Therefore, the diffuser circuit was extended to a 2D matrix. Using a circuit such as this, it is easy to see how particular dendritic morphologies can be programmed into a circuit, Fig. 5. A more thorough discussion of the dendrite can be found in [13].

Since the actual dendrite morphologies can be programmed into the circuit, investigations into actual dendritic computations can be performed [14]. The conductances of each of the gates can be individually programmed by using floating-gate transistors. With a large dendritic matrix, some conductances can be programmed to be virtually off, thus allowing the

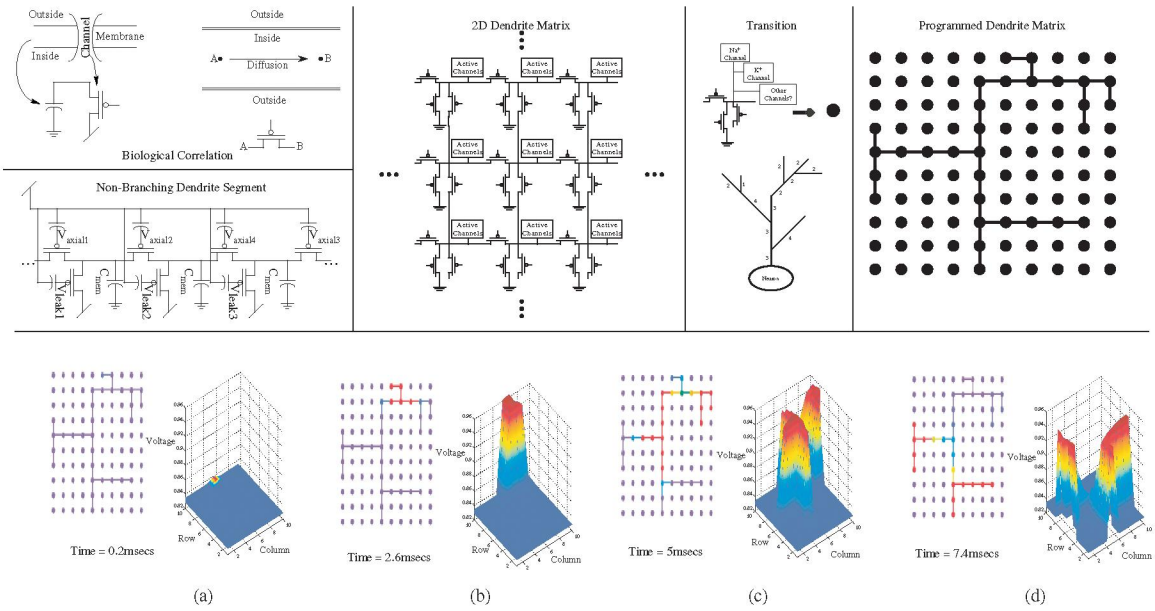

Fig. 5. Biologically, ions in a dendrite can either diffuse axially along the length of the dendrite, or through channels across the membrane. It has already been shown that a better model for diffusion is sub-threshold MOSFETs over the classical resistor models. Using this fact, a diffuser circuit is used as a better model of a dendrite segment. Extending this circuit to a 2D circuit allows the dendrite to model branching behaviors. The conductances of each of the MOSFETs can be individually tuned to program whatever dendritic morphology is desired. Data from a 2D dendritic matrix is shown in a-d. Each individual frame represents a different snapshot in time as an action potential is generated and then propagated throughout the programmed portions of the matrix.

dendrite matrix to emulate either complex large arborizations, or multiple smaller less complex ones. In the FPNA, this allows for the emulation of large complex single cells, or networks of smaller less complex ones.

III. Conclusion

This structure has been fabricated on a commercially available 0.35μ process and is currently being tested. This work provides investigators with a significant structure whereby they may begin to investigate a wide range of neuronal systems. Questions about the computation of a particular dendritic arbor can now be investigated, as well as networks of neurons. Complex models of cells can be made by modeling small sections of the cell in each sub-block of the FPNA. Small central pattern generator networks can be created by connecting a few approximate models of neurons. Large neuronal networks can be created by connecting hundreds of the individual blocks. As improvements are made both in the connectivity and functionality of each block a vast array of more complex neuro-mimetic systems can be built and studied.

References

[1] T. S. Hall, C. M. Twigg, P. Hasler, and D. V. Anderson, "Developing large–scale field–programmable analog arrays for rapid prototyping," *International Journal of Embedded Systems*, 2004.

[2] F. Tenore, R. J. Vogelstein, R. Etienne-Cummings, G. Cauwenberghs, M. A. Lewis, and P. Hasler, "A spiking silicon central pattern generator with floating gate synapses [robot control applications]," in *IEEE International Symposium on Circuits and Systems*, vol. 4, May 2005, pp. 4106–4109.

[3] G. Indiveri, E. Chicca, and R. J. Douglas, "A vlsi reconfigurable network of integrate-and-fire neurons with spike-based learning synapses." in *Proceedings of the Europan Symposium on Artificial Neural Networks*, 2004.

[4] E. Farquhar and P. Hasler, "A bio-physically inspired silicon neuron," *IEEE Transactions on Circuits and Systems I*, vol. 52, no. 3, pp. 477–488, March 2005.

[5] G. Serrano, P. Smith, H. Lo, R. Chawla, T. Hall, C. Twigg, and P. Hasler, "Automatic rapid programming of large arrays of floating-gate elements," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, vol. 1, Vancouver, BC, Canada, 2004, pp. 1373–1376.

[6] P. Smith and P. Hasler, "A kappa projection algorithm (kpa) for programming to femptoampere currents in standard cmos floating-gate elements," *Midwest Symposium on Circuits and Systems*, vol. 1, pp. 196–199, Aug. 2005.

[7] C. Gordon and P. E. Hasler, "Biological learning modeled in an adaptive floating-gate system," *International Symposium on Circuits and Systems*, vol. 5, pp. 609–612, May 2002.

[8] E. Kandel, J. Schwartz, and T. Jessell, Eds., *Principles of Neural Science*. New York: McGraw-Hill, 2000.

[9] W. F. Agnew and D. B. McCreery, Eds., *Neural Prostheses: Fundamental Studies*. Prentice Hall, Englewood Cliffs, 1990.

[10] C. Gordon, E. Farquhar, and P. Hasler, "A family of floating gate adapting synapses based upon transistor channel models," in *Proceedings of the 2004 International Symposium on Circuits and Systems, 2004. ISCAS '04.*, vol. 1, May 2004, pp. I–317–I–320.

[11] C. Gordon, A. Preyer, K. Babalola, R. Butera, and P. Hasler, "An artificial synapse for interfacing to biological neurons," *IEEE International Symposium on Circuits and Systems*, Submitted to 2006.

[12] K. Boahen and A. Andreou, "A contrast-sensitive retina with reciprocal synapses," in *Advances in Neural Information Processing Systems*, J. Moody, Ed., vol. 4. San Mateo, CA: Morgan Kaufman Publishers, 1991.

[13] E. Farquhar, D. Abramson, and P. Hasler, "A reconfigurable bidirectional active 2 dimensional dendrite model," in *Proceedings of the 2004 International Symposium on Circuits and Systems, 2004. ISCAS '04.*, 2004.

[14] P. Hasler, P. Smith, E. Farquhar, and D. Anderson, "A neuromorphic ic connection between cortical dendritic processing and hmm classification," *Digital Signal Processing Workshop*, pp. 334–337, August 2004.