A REVIEW ON THE DESIGN AND IMPLEMENTATION OF NEUROMORPHIC CHIPS

SAHANA MURTHY
R.V. College of Engineering
E-mail: sahana.murthy1317@gmail.com

Abstract— In recent years, there has been growing research in neuroscience and the application of neural computation in today’s technology. By incorporating neural elements in electronics, it is possible to have a more efficient design with faster computation speeds. Neurons are used to transmit and process information with the help of generated spikes in neuromorphic systems. Designers of neuromorphic chips can decide between various architectures, nanomaterials and approaches. This paper highlights the different design options available along with comparison between them and the application for which they are suitable.

Index Terms— LIF, Neuron, Neuromorphic, Synapse.

I. INTRODUCTION

Neuromorphic engineering, a highly interdisciplinary field, takes inspiration from neuroscience to build electronic systems. It is a combination of biology, computer science, mathematics and electrical engineering. Modern computers are linear, moving data back and forth over between a central processor and memory chips. On the other hand, the brain is highly interconnected with logic and memory intricately crossed. Neuromorphic chips aim to mimic the brain’s architecture, thus increasing the number of processes and responding power.

Current research in neuromorphic engineering aims to develop neural models efficient in area, power and cost. In [1] the main focus of the Human Brain Project is on brain inspired technology. The human brain can be treated as a supercomputer as it is efficient in power consumption, self-repairing and self-learning. In the same way, the Human Brain Project promotes data sharing and large scale collaboration to mimic brain functioning at different levels. There are six major development platforms, i.e., neuroinformatics, brain simulation, medical informatics, neuromorphic computing, neurorobotics and high performance computing.

Neurogrid is a neuromorphic system used for simulating neuron models in real time applications [2]. This hardware architecture was designed and built at Stanford University by the Brains in Silicon group. The designers of Neurogrid chose to make use of circuits that imitated all neural elements except for soma components. All circuits used analog neural models for the axon components as power and area were efficiently used and interconnected neural array in a tree network, rather than a mesh, so throughput was maximised. The major problem that occurs with brain simulation is that computers are sequential; this means that programs are executed step by step in a fixed process. Unlike computers, brains are able to run different processes in parallel. The main advantage of Neurogrid is that the simulations are detailed enough to cater to specific cell properties while still large enough to include multiple cortical areas.

TrueNorth, produced by IBM, is a similar neuromorphic CMOS chip [3]. This chip aims to overcome problems related to the memory bottleneck, by using custom electronic circuits dedicated to a signal neural element. As there are multiple design specifications, basic computer aided design tools are not sufficient to design and fabricate these chips. Design methodologies that included asynchronous-synchronous circuits are specifically developed for these neurosynaptics chips. In terms of connectivity and neural parameters, the TrueNorth chip is fully configurable allowing for various applications. The TrueNorth chip is energy efficient; using CAD placement tools logical networks to physical neurosynaptic cores. This chip serves as the base element for future neuromorphic designs.

The brain is made up of millions of neurons. Neurons are electrically excitable cells that can transmit and process information via synapses, where the axons of one neuron interact with the dendrites of another. The neural elements that are incorporated into hardware are a soma (cell body), dendrites and an axon. This paper focuses on the approaches to designing neuromorphic chips, followed by the implementation of nanoelectronics to emulate neurons and finally the various architectures to facilitate interneuron communication.

II. ABSTRACTION LEVELS OF BRAIN MODELLING

Modelling can be done to reduce the overall complexity of circuits and provide simplified versions of real time systems. There are seven different abstraction levels that allow a set of functionality, platform independence and interoperability.
Table I shows the equivalent neuroscience system for the corresponding electrical component at each abstraction level. Currently, there has not been much research done in the protein level of neuromorphic engineering. This level consists of the basic gene structure. Electron and ion exchange occurs at the membrane level. Transistors are simplified to the form of a switch with “on” and “off” states. They work in the subthreshold region where the relationship between current and voltage at the brain cells are represented in the V-I characteristics of the transistors. The main focus of the membrane level is selective attention. The communication between the neurons, occurring through the synapses, is implemented using CMOS circuits. Memristors can be used as an alternative to CMOS. At this level, the main focus is on memory modelling and learning. Perceptron, an algorithm to determine whether an input belongs to one class or another, is modelled at the component level. At the circuit level of abstraction, blocks or cells are described. This is equal to multiple neurons working together. At the system level, the working of each block is controlled by a macroblock. The working of the macroblocks is similar to its counterpart in neuroscience, the brain. The final level, behaviour level, defines the overall functionality of the system [4].

There are two different approaches to designing neuromorphic chips, as depicted in Fig. 1.

A. Top-down Approach
The top down approach focuses on the overall functioning of the system. It breaks down the system into various subcomponents to obtain the solution through reverse engineering. Each subcomponent is then further divided into smaller subcomponents until finally reducing it to the base elements.

B. Bottom-up Approach
In the bottom up approach, many base elements are designed at very specific detail. These elements are then joined together until the complete top layer is formed. This approach is suitable for local optimisation without focussing on meeting global purpose.

In practice, a combination of both methods, known as meet-in-the-middle approach is used. In this method, the actual designing of cell blocks is performed with the bottom-up approach while the specification synthesis process uses the top-down approach.

II. IMPLEMENTATION OF NANOELECTRONICS
Nanoelectronic materials have been crucial in the development of low area and power efficient circuits, which can be used to implement the functionality of neurons and synapses.

A. Analog Neuron
A neuron is surrounded a nd filled by water containing ions, and hence carrying electrical charge. Each neuron is bound by an insulating cell membrane which can maintain a certain concentration of charged ions on either side. This concentration of ions determines a capacitance C. When neurotransmitter cause ion channels on the cell membrane to open, there exists a movement of ions. This phenomenon is known as “firing of a neuron”. Firing of a neuron can be mathematically described by a time dependant current, I(t), and a change in voltage, resulting in a voltage spike. The voltage spike travels along the length of the cell and triggers the release of more neurotransmitters.

The leaky integrate-and-fire (LIF) model is used to replicate the behaviour of the neuron cell body. The addition of a “leak” at the membrane potential, allows for time dependant memory by reflecting the diffusion of ions when cell equilibrium is not reached. This can be described by

\[ C \frac{dV(t)}{dt} = -g_L(V(t) - E_L) + I(t) \]

Here, C is the membrane capacitance, I(t) is the external input current and gL is the leakage conductance term. When V(t) increases beyond a set threshold, it resets back to EL. Here, the voltage across the capacitor represents the membrane potential, the transistor is used to implement leakage and threshold detector is used to resent the spike.
When a neuron spikes, voltage is generated along the axon which decreases exponentially. This information is then sent to other neurons through their synapses.

Another factor to consider is information passing in both directions of the neuron. So, while a voltage spike is sent along the axon, a corresponding spike is sent along the dendrites, decaying exponentially. Hence, when voltage spikes occur at different neurons, the overlap between them causes the synapse to get programmed. To generate the programmed waveforms, there is a specific driver circuit required, as seen in Fig. 3, consisting of the driver and predriver stages.

Capacitors C1 and C2, current source I0 and switches S1 and S2 belong to the predriver stage. The predriver stage generates the voltage waveform while the driver stage, using transistor M1 and switches S3-S5, drives the dendrites and the axon. The input to the switches is given by the pulse controlled signal which is generated from the pulse generator block and the comparator.

B. Digital Neuron

The digital scheme of the neuron model makes use of CMOS logic circuits to form neurons and SRAM circuits for synapses as seen in Fig. 4. First, inputs from the synapse array and external inputs are given to the integrate-and-fire circuit. Excite, inhibit, external and leak input are multiplexed and given to a 16-bit adder. The output value of the adder is then compared with a predefined threshold. The input of the learning circuit is given as the comparator value, which is also given as the output to the axon. The learning circuit consists of a pre synapse counter and post synapse counter, which is used to determine the difference between consecutive voltage spikes during neuron information transmission.

The digital model can be described by (2).

\[
C \frac{V(n+1) - V(n)}{T} = -g_L(V(n) - E_L) + I(n)
\]  

In (2) C, gL and I(t) represent membrane capacitance, leakage and input current respectively. When a spike is detected, the neuron circuit reads the synapse value from the axon of the respective neuron. The value obtained by the counters is then programmed into waveforms, similar to the working of analog neurons.

C. Comparison between Analog and Digital

In analog systems, the area occupied can be limited by using deep trench capacitors, leaving a majority of the area for the integrator and comparator circuits. In digital neuron circuits, in addition to the area occupied by the LIF circuit, space for control circuits is required as well. Thus, the area occupied by analog neuron circuits for a fixed number of neurons can be at least ten times smaller than that of digital systems. In addition to this, the total power required for analog systems is two times lower than that of digital systems.

IV. NEUROMORPHIC ARCHITECTURE

Neuromorphic hardware, which consists of axons, dendrites, synapse and soma elements, can be classified based on hardware.

A. Fully dedicated

In fully dedicated type of architecture, all of the elements are fully dedicated, as seen in Fig. 5. The neurons are represented by the triangles and the squares represent individual synapses. With N neurons, a fully connected network requires N2 synapse elements. There are specific hardware elements for each part of the neuron (axon, dendrites and soma). For this reason, there is a one to one correspondence between elements of neural networks and chip elements.
A Review on the Design and Implementation of Neuromorphic Chips

B. Shared Axon
Fig. 6. depicts the shared axon architecture. Here, every neuron is assigned a particular address. Once a neuron spikes, its address is encoded by the transmitter, sent over a digital bus and decoded at the receiver. There is a fixed set of wires that are used for connecting to all of the axons. Hence, the number of wires in this system reduce from N to log2N and the number of synapse elements to connect N neurons is N^2.

C. Shared Synapse
In this architecture, depicted in Fig. 7, there is a dedicated circuit forming connections between the synapses of the neurons. Voltage spikes are then directed towards this circuit. The branching of the axon is realised with the help of SRAM. Here, only N synapse elements are required to connect N neurons. For situations when connectivity is sparse, shared synapse architecture is preferred.

D. Shared Dendrite
As seen in Fig. 8, there exists a shared resistive network connected to the dendrites. Similar to shared synapse architecture, there are only N synapse elements.

V. APPLICATIONS OF NEUROMORPHIC ENGINEERING
With advantages like efficient energy consumption and high processing speed, neuromorphic technology can be incorporated in a variety of applications. This includes robotics, imaging, sensing, security, and so on. For example, drones can be programmed with improved visual cue recognition. Modern computers currently work in preprogrammed ways but by
incorporating neuromorphic chips, they can soon react on their own.

CONCLUSION

The brain is capable of processing and computation at unthinkable speeds. By incorporating the architecture of the brain into electronic circuits, the overall efficiency, power consumed and area occupied can be optimised. There exist various methods to design neuromorphic chips and the most suitable one as per the specific application must be selected. This paper provides an analysis on the various possible approaches, nanoelectronics and architectures available for neuromorphic chips and compares the merits and demerits of each.

REFERENCES


***