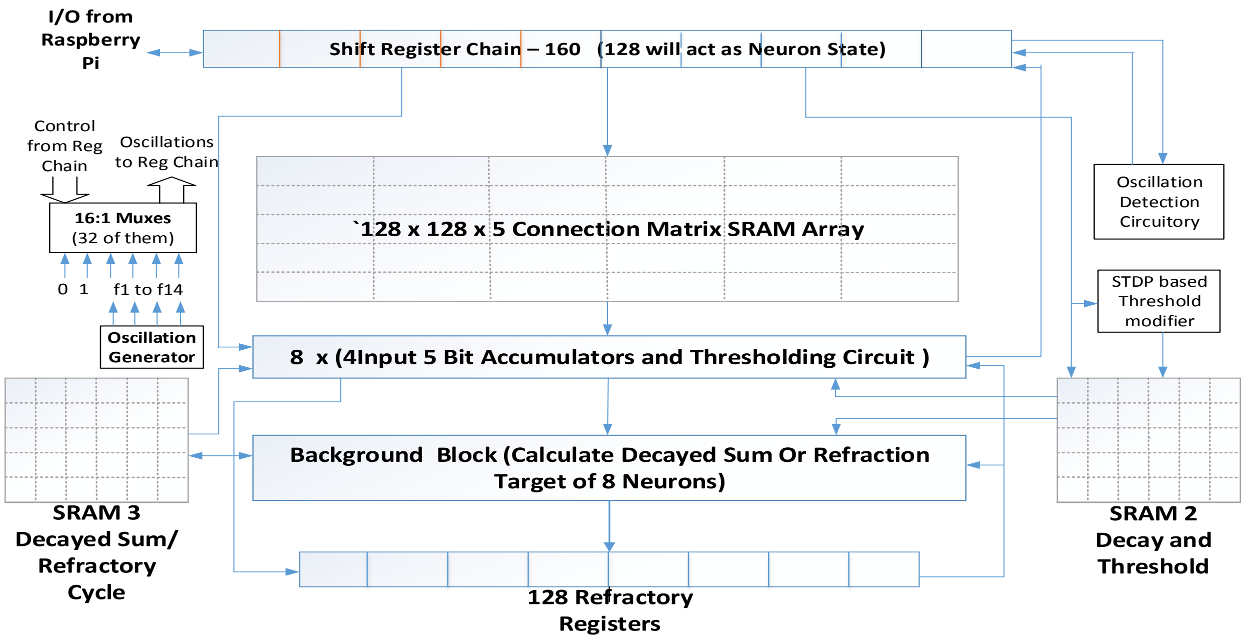
Introduction

The neuromorphic chip being designed in this project has 128 spiking neurons that can be configured as arrays of oscillators or used in more conventional neural network architectures.

Neuromorphic Chip - Medulla



I/O from Raspberry Pi
Shift Register Chain – 160 (128 will act as Neuron State)
Control from Reg Chain
Oscillations to Reg Chain
16:1 Muxes (32 of them)
0 1 f1 to f14
Oscillation Generator
`128 x 128 x 5 Connection Matrix SRAM Array
Oscillation Detection Circuitory
STDP based Threshold modifier
8 x (4Input 5 Bit Accumulators and Thresholding Circuit)
Background Block (Calculate Decayed Sum Or Refraction Target of 8 Neurons)
SRAM 3 Decayed Sum/ Refractory Cycle
SRAM 2 Decay and Threshold
128 Refractory Registers

Figure 1: Block Diagram of the Chip

The chip can be configured to have up to 32 oscillators each of which can be programmed to one of the 16 possible frequencies. These digital oscillators couple to different nodes in accordance with the coupling coefficients. This would result in new oscillations in rest of the 96 nodes in the pool. Each output node behaves as in a regular spiking neural network. The integrate and fire model based neurons have programmable decay, thresholds and refractory period. The SRAM array stores the coupling coefficients each of 5 bits. Threshold adaptability based on STDP is also being explored, but not yet finalized.

The 32 oscillators must be assigned frequencies externally. These are given as 4 bit control inputs from the raspberry pi. Thirty-Two 16:1 muxes, based on these control signals, give oscillations to the 32 input nodes from the oscillation generator circuits. The coupling coefficients are stored in the main SRAM array which is 128*(128 *5) cells big. Threshold (12 bits) and decay rate (4 bits) of each neuron will be stored in SRAM 2. If the neuron does not fire, the accumulated sum decayed by the specified rate, is stored in SRAM 3. Else, the iteration cycle up to which the neuron will be refracting is stored in SRAM3. The chip processes 8 neurons at a time in the 8 accumulators and thresholding circuitry. Background Block calculates the decayed sum based on decay rate simultaneously. It also calculates the target iterative cycle till which neuron would refract in case neuron does fire. This block will also have the comparator to check if current iterative cycle is the one till which the neuron refracts. The chip has another 128 registers to store the refractory information of each neuron. The oscillation detection circuitry keeps track of the status of nodes across iterations and detects oscillations.

The chip can be reconfigured for performing pattern recognition and other more conventional neural network applications. The spiking neural network would perform real-time computations on continuous streams of data. The information is stored as spatio-temporal patterns inside the recurrent neural network during training phase. This can be used like an associative memory for pattern recognition based applications.

Project Status

In software, we have the pattern recognition application working and are currently working on programming methods. Simulations are being done to train weights and to have STDP with minimal hardware implications. On the hardware side, we have integrated the memory components to the design and verified the functionality to match software simulations. Currently we are modifying the RTL design to include the blocks necessary for coupled oscillator functions.

Some Notes on Programming the Chip

The physical implementation of this system will be on a neuromorpohic chip with 128 binary nodes, 32 of which are dedicated to generating one of 8 sets of 4-bit frequencies. The other 96 can be used wherever they are needed. Since there are only 8 available frequencies, principle components analysis will be performed on the feature space prior to teaching to determine which basis would be best. Once the input frequencies, output nodes, and internal weights have been determined, the software model is trained over the data set until a certain accuracy threshold is reached. At this point, the output weights of the software model (and subsequently the rest of the predefined data) is transfered onto the chip, where the same calculations can be performed more efficiently and expediently.

Within the chip itself is the original implementation of, say, a Hopfield network along with a continuous buffer of neuron spike times in the last t seconds so that the threshold change due to STDP can be calculated. Although this will create an accurate representation of the threshold rising and falling, there will be a t second delay on the threshold reduction as the buffer calculated the total difference of the thresholds of all of the spikes within t seconds of a spike when that spike reaches either end of the buffer. Meaning, that when the spike's time enters the buffer, all of the other spikes currently within the buffers are used to calculate threshold raising, and after t seconds, when the spike leaves the buffer, all other spikes currently in the buffer are used to calculate threshold reduction of the leaving spike. While this creates a delay, it is much simpler in memory. The change in threshold will be calculated using a predefined and hard-coded piecewise function based on the exponential function although polynomial and logistic functions will also be tested.