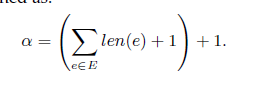
The neurons are partitioned into blocks of neurons to facilitate future pipelining. Connectivity can be established between any neuron to any other neuron. The network is established with an array of defparam statements that define how the output spikes of the neuron array are connected to the synapses. This can be easily automatically generated from a {?name of standard graph format?} file that describes the graph. Each neuron can also be individually configured with defparam statements if desired.

Once the network is established, a neuron can be stimulated to begin the search for the multiple-destination shortest paths. The simulation is run for alpha time steps.



Each neuron records the time step that it was triggered. This is then read out of the neurons using AXI4 streaming protocol. All the outputted data from the neurons are concatenated into one stream to be read out at the top level. An open source library, <https://opencores.org/projects/qaz_libs>, was used to support AXI4 streaming and other miscellaneous functions.

An ultra-small graph of 4 interconnected nodes, ultraSmallKronecker, was successfully simulated. The correct output was observed for various stimulated neurons. A larger graph of 32 interconnected nodes, is currently being debugged.

Unfortunately time ran out before a fully pipelined SNN could be implemented. The Verilog HDL is structured to ease the addition of pipelining and synchronizing hand shaking in the future. This is needed to bring the code to life on a FPGA. The design can support 1024 neurons partitioned in to 32 blocks of 32 neurons. The adder tree needs to be replaced with streaming multiply accumulators like the one found in, <https://opencores.org/projects/keras_to_fpga>. Also the all the synchronizing signals from each of the neurons would need to be fanned out and pipelined. Additional state machines would be required for coordination.