SuperNeuronBench - A Computational Benchmark for Comparing Time-Multiplexed Neuromorphic Architectures on Open-Source FPGAs to Parallel Biological Software Models

Neksha DeSilva
neksha-52950@anandacollege.edu.lk

International Youth Neuroscience Association
IYNA Summer, June-July, 2026
Capstone II

Git SNB Website

Introduction, Research Aims

The real time simulation of biological neurons is a main, practical problem in computational neuroscience and neuromorphic engineering, and is now increasingly common due to hardware cost, lab tools shortages, reduced research funding spending gazettes, etc. For example, one cortical column in a brain of a mammal includes approx. tens of thousands of neurons. (depending on the species.) And it includes hundreds of millions to billions of neural synaptic connections. In a biological brain, those neurons and connections have to update in that same clock cycle (refer Sievers et al., 2024). On the other hand, current computing hardware is unable to provide this parallel process for an ordinary laboratory with a relatively small fund or budget. Information Theory, time complexity and computation in general, used an already existing method in computer science and telecommunication for achieving a remarkable efficiency in computational neuroscience, named as time multiplexing. Meaning that, one processing element cycles through the states of several simulated neurons in a sequence, in the sequential algorithmic order. Time multiplexing architecture does not allocate one separate physical processing element (or logic elements/LEs) to every neuron in the simulation process. In this research, this combined processing element is named as a SuperNeuron. In addition to that, this is not solely an abstract concept. The University of Manchester's SpiNNaker system uses several artificial neurons (hundreds of neurons) through each physical processor core, and this process enables real time biological neuron simulation at a greater scale (refer Furber et al., 2014). Loihi 1 and 2 by the Intel Corp. and the IBM TrueNorth also process a greater amount of simulated neurons by using the method called Time Division Multiplexing (or TDM) and event driven processes (refer Davies et al., 2018, Merolla et al., 2014). IBM Corp's TrueNorth achieved about 16 million neurons by using the NS16e system with several

cores (refer DeBole et al., 2019).

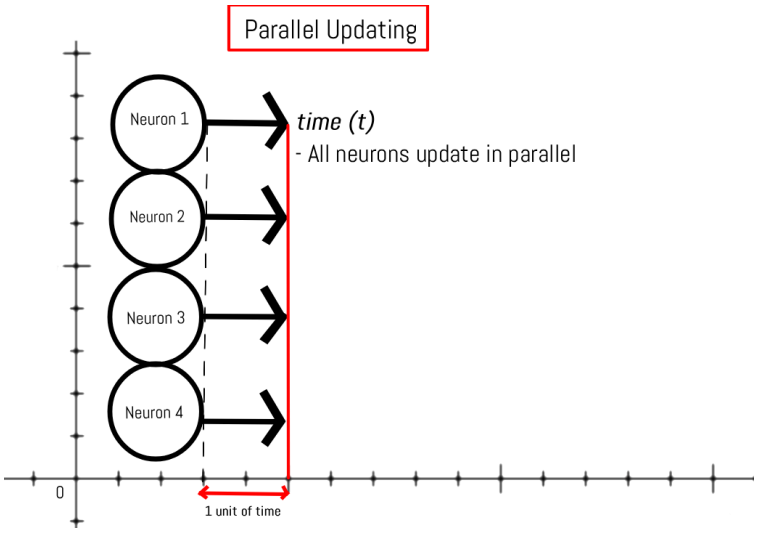

Figure 1: Biological Parallel Updating

As with any computational method, some disadvantages do exist. Here we are focusing on the effects of time multiplexing updates on the simulated neurons in a sequential process. Biological neurons, conversely, do not rely on this sequential processing architecture. Biological neuronal processing uses parallel processing. Due to this reason, there can be timing gaps or resting states between the expected update timestamp and the real update timestamp value. The correlation is, when there is a greater clock speed, the probability of a time gap reduces based on the instrumental error of that hardware (Should not be confused with zero error). A smaller gap in timing is not equal to the real parallel processing ability in biological neuronal processes, and these variable gaps may result in different mismatches in function. For example, different neural computing processes such as the Spike Timing Dependent Plasticity (or STDP), Temporal Gating (or TG), Temporal Coding (or TC), Anomaly Detection algorithms preventing false positive states, are depending on the specific relative timing between two spikes. Another set of processes is depending on the total rate of neuronal firing. Pattern recognition and fast retrieval is an example for this type.

Already established research minimally and partially addresses the impact of time multiplexing on different neural computation processes. And specifically, it is unclear about which specific computation types resulted in a reduced accuracy relative to the biological process they are based on due to the time multiplexing method. This includes neuromorphic architectures such as Spiking Neural Networks, Echo State Networks and models such as Izhikevich SNN model. And that early research does not output the magnitude of sequential processing a superneuron can successfully process before the accuracy reduces or dilutes as a percentage. This problem is visible directly in open source neuromorphic models or architectures. On the other hand, Loihi, TrueNorth and Northpole, etc. are closed source, proprietary and patented systems. Open source systems such as NeuroCoreX are configurable and are accessible for researchers with lower resources. Even further, a benchmark which isolates the results and effects of time multiplexing is not currently

available. As an already existing benchmark, NeuroBench mainly measures the accuracy and efficiency of the total system (refer Yik et al., 2025). It does not isolate the time multiplexing process into a separate evaluation environment.

The main hypothesis of this research is that as long as the system relies on a time multiplexed architecture, the reduction of biological accuracy is not equal across every computational task being simulated electronically, regardless of the hardware used. In this research, there is an expectation of failure for tasks that are dependent on the timing levels at a lower multiplexing density than tasks that are rate coded. Recognizing patterns, rapidly retrieving them, is expected to be functional through a greater density than STDP, TG, TC, and False Positive Detection.

Aim 1 of this research produces a result even if a later hardware result is different from the expected result. And aim 3 depends on the data output of aim 2. If the expected task ordering is not found, this is also a relevant result. Meaning that the error can be related to hardware's precision or the quantization process instead of timing. In both scenarios, regardless of other outputs, the aim 1 and aim 2 will provide the data related to a benchmark framework and a separate hardware dataset, which are not currently available for open source systems.

The Background and Significance

Obstacles in the process of scaling and simulating biological neural networks

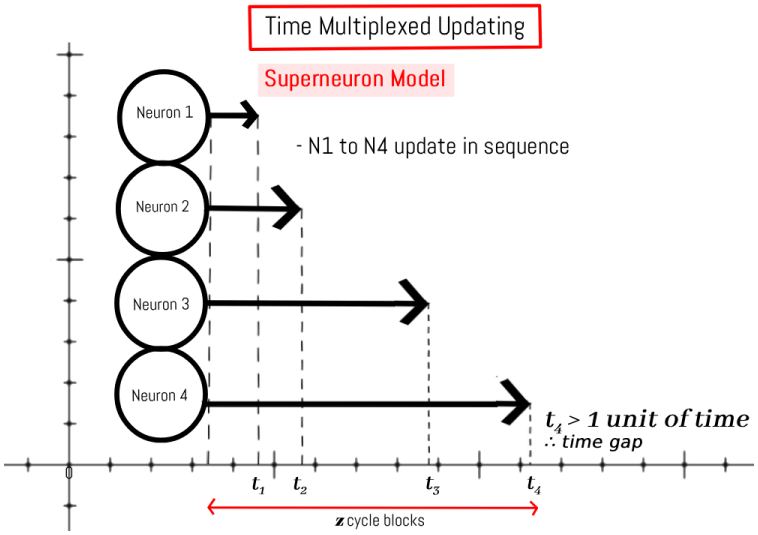

Figure 2: Time Multiplexed Updating (Sequential)

As the field of computational neuroscience expands, simulations of Neural Nets enable researchers in distinct and interconnected dimensions, including brain computer interfaces, artificial intelligence, computer vision, high performance computing and more. They enable researchers to test abstract mathematical concepts, bitstreams or simulations written using programming languages in environments similar to real biological tissues. This reduces the cost of wet labs, invasive medical tools, human errors, and more, which directly benefits the advancement of the large domain of neuroscience. The main disadvantage is the ability to scale. Animal neurons operate in a parallel process. Every biological neuron receives inputs, changes internal states and configurations the same moment the other neurons in the simulation are reacting, producing input and changing its internal state. So it is not a system with a traditional sequential task queue. Recent studies showcase statistics related to these biological systems. One cortical column of a human includes approx. 10,000 to 100,000 neurons with hundreds of millions to billions of synaptic connections. (refer Sievers et al., 2024). In a computational system, those synaptic states, weights, edge connections to other neurons, etc. have to be stored and updated in every clock cycle. Modern computing architectures with separate RAM units and processors, this constant data travels from CPU to registers to cache memory to SDRAM as a sequential cycle is time consuming and is not capable of obtaining the true parallel execution.

The Process of Time Multiplexing in Detail

Time multiplexing method allocates one processing unit for multiple logic tasks. This is achieved through the division of time, into separate time slots. In a neural computation, one superneuron cycles through the states and data of a higher number of neurons. This is an already existing engi-

neering process. The SpiNNaker system is such an example based on the ARM processing units. (refer Furber et al., 2014). Loihi and TrueNorth processors are capable of simulating approx. one million neurons on a single chip (refer Davies et al., 2018, Merolla et al., 2014). IBM TrueNorth (version NS16e) system is capable of simulating approx. 16,000,000 neurons using multiple processing cores. (refer DeBole et al., 2019).

These systems are based on the theory of event driven computing. Meaning that, a neuron is updated when a relevant event is received. In that case, it does not update states of all the neurons at once. This reduces the processing cost by a large margin, compared to traditional GPUs, CPUs, or TPUs. But these are closed source, commercial systems. In order to research about this topic, we are focusing on open source neuromorphic hardware models and designs. NeuroCoreX is an example. (refer Gautam et al., 2025, Oak Ridge National Laboratory, n.d.).

The Time Gap in Sequential Updating Process

A superneuron updates and changes the states of simulated neurons one after the other. And there is a timing gap between the update time in the real biological process and the update time in the time multiplexed system. A more advanced, high performance neuromorphic processor is capable of executing this updating process at greater speeds that reduce that time gap. But a reduced time gap does not prove that it does not cause any disadvantage. For example, STDP depends on the correct order and the interval time between a presynaptic spike and a postsynaptic spike event. Even a microscopic timing delta could change the direction or the intensity values of the update. On the other hand, a rate coded task uses a total amount of spikes across a timeframe. Storing patterns and memory retrieval can be given as an example. This second type is capable of tolerating this difference more than the first type and this difference is the main reason of separating them into multiple task groups. Because, clearly, time multiplexing does not reduce every neural computation accuracy equally. Superneuron benchmark is the array of testcases that tests the time dependent and rate dependent task groups separately.

Gaps in the Test Cases of Current Neuromorphic Benchmarks.

A majority of modern evaluation benchmarks output the energy efficiency, watts per hour, or the final accuracy across thousands or more clock cycles. (refer Ostrau et al., 2022). This outputs the total efficacy of the system. But the average is a smoothed value. It does not include those data of individualized states. This output is not capable of identifying the origin clock cycle(s) of the error or the process types which were disadvantaged because of this error.

NeuroBench is a benchmark that proposes a new standard for evaluating neuromorphic systems. (refer Yik et al., 2025). Its algorithm and system tracks compare the final task accuracy and the efficiency of computing in different platforms and algorithms. NeuroBench system compares different hardware platforms on a specific task. But Superneuron Benchmark does not change the task, model or the architecture. But rather, our benchmark changes the magnitude of time multiplexing. This enables us to research at the micro level rather than the macro level, to see the specific effect of one change on the biological accuracy. On the other hand, SNABSuite and the

related benchmark work use independent backend based SNN tasks which can be connected to multiple neuromorphic platforms (reference: Ostrau et al., 2022). It is a static group of tasks with a set of measurements and a hardware connection algorithm. But the testing method is different.

The Scientific Value and Outcomes

This research supports two main scientific communities. The first division includes hardware designers, systems researchers and open source hardware developers. The benchmark can depict the types of tasks a specific superneuron configuration advantages and which task types it disadvantages. As an example, a system would be able to use a lower amount of neurons per superneuron for a real time circuit while using a higher multiplexing for a SNN. The second group includes neuroscientists, artificial intelligence researchers and brain computer interface researchers who use CAD software for modeling neurons. In addition to that, this benchmark can identify the accuracy of analysis types and route errors for unsupported, malfunctioned models or architectures by providing clear error codes with documentation.

Research Strategy

Testing Process

The main requirement in this proposal is a biological reference accuracy which does not come from the system being tested. The effect from time multiplexing cannot be measured without a reference for the parallel process. Due to that reason, the same spiking neural network model will run through two conditions. The first condition is a complete parallel software simulation. NEST or Brian2 will be used for this process (refer Gewaltig and Diesmann, 2007, Stimberg, Brette, and Goodman, 2019). In this software condition, every neuron updates at the same simulation time. This produces the biological accuracy spike trains. The second condition is the time multiplexed superneuron implementation on the FPGA hardware. The difference between those two outputs gives the measured disadvantage from time multiplexing for that configuration and task.

Three values describe a superneuron configuration.

1. x denotes the number of neurons in the simulation which are assigned to one superneuron. Those neurons update as a sequence in a clock cycle.
2. z denotes the number of time blocked sub divisions used to update the x number of neurons during one biological simulation timeframe. Otherwise in general, $z \geq x$. Some neuronal updates require multiple programmed steps, such as the Synaptic Input Accumulation and the Threshold Checking process.
3. C is the number of superneurons operating in the parallel system. Hence the total amount of simulating neurons in that unit system is $x \cdot C$.

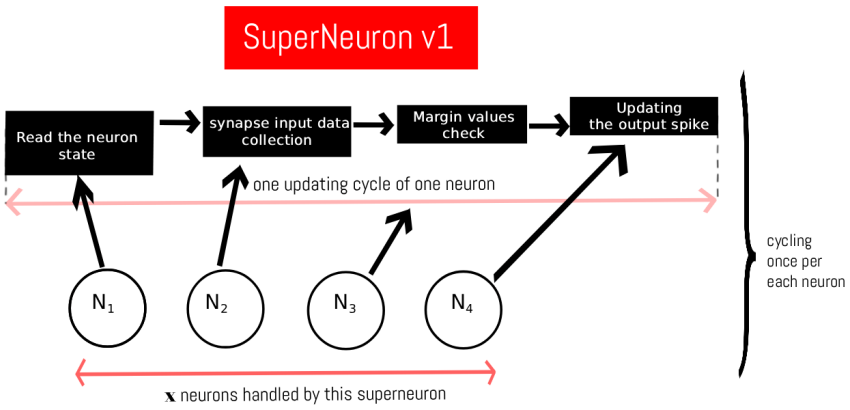

Figure 3: Superneuron (purposed)Internal Pipeline

Creating a Benchmark for Measuring Biological Accuracy

This section is relevant to the aim 1 of our research. This aim includes four types of tasks across the timing range which was discussed in the previous section. The first is storing patterns and retrieving them from that library. This is a rate coded task and is directly dependent on the number of spikes. The second one is Spike Timing Dependent Plasticity, or STDP. STDP depends mainly on the order of the commands. It depends on another factor. Which is the time gap between the pre synaptic and post synaptic spikes. The third task is the anomaly or coincidence detection. The last task is the temporal gating process (or sensory gating) and gate locking. For each of those task types, a separate small circuit will be created in NEST Framework or Brian2. These circuits will include approx. hundreds to thousands of computing neurons. This software outputs the biological reference spike sequences for it. In here, the difference between the biological reference and the time multiplexed output of artificial neurons will be measured using the theory "Van Rossum Spike Train Distance" (refer van Rossum, 2001).

$$f(t) = \sum_i H(t - t_i)\, e^{-(t-t_i)/\tau}, \tag{1}$$

This above equation is a direct representation from Professor Mark Rossum's research paper. t_i denotes the spike times. And H denotes the Heaviside step function. After the filtering process, the distance between the two spike trains f_1 and f_2 is

$$D_{vR}(s_1, s_2) = \frac{1}{\tau} \int_0^\infty [f_1(t) - f_2(t)]^2 \, dt. \tag{2}$$

The τ value controls which specific time difference is greater and then assigns it the greater position role. And a lower τ value assigns a greater position to a close spike coincidence. A larger τ value assigns a greater position to the difference in the spike rate. Due to that reason, this distance process is important for correctly separating the time depending and rate depending task groups. This equation is also a direct representation of Professor Rossum's paper.

Using the van Rossum distance, this will result in a normal, real biological accuracy value for the configuration (x, z, C). This normal biological accuracy equation is an invented mathematical equation used from here onward

$$\text{naccuracy}(x, z, C) = 1 - \frac{D(x, z, C)}{D_{\max}}. \tag{3}$$

$D_{\max}$ represents a separate normalization ceiling for every task. It is calculated as the van Rossum distance between the biological reference accuracy and a fully shuffled spike train from that similar task. This shuffled spike train represents the loss of the task related timing information.

A configuration passes the benchmark for a task when

$$\text{naccuracy}(x, z, C) \geq \theta. \tag{4}$$

The θ value represents the pass and fail margin value. It will be selected separately for every task type from the minimum functional tolerance reported in the relevant spike timing research. Due to that reason, the proposal does not use one arbitrary margin value for every task.

The Victor Purpura distance is another already existing spike train measurement process (refer Victor and Purpura, 1996). This calculates the minimum processing amount for inputting, deleting or moving spike values in order to change one spike train into another. We selected the Van Rossum distance algorithm because of its mathematical properties. more specifically, euclidean properties. And Van Rossum distance supports better for task averaging as well.

Expected Results The direct output of this is the benchmark specification. This includes the task group, the normal accuracy measurement process (naccuracy), pass and fail boundary values. These results do not depend on the success of the AIM 2. Which means, this aim will be already completed at the moment when the software biological accuracy simulations and those benchmark values are ready and already checked.

Measuring the Hardware Performance, Power and Others.

This testing will use a Japanese variant of Intel Cyclone IV EP4CE15 and Cyclone V SoC FPGA boards. These boards are already available for this project. Intel Quartus Prime Lite software will be used for the programming, hardware analysis processes and synthesizing processes. The Cyclone V board contains approx. 150,000 logic elements. The Cyclone IV EP4CE15 contains approx. 15,408 logic elements (refer Intel Corporation, n.d.).

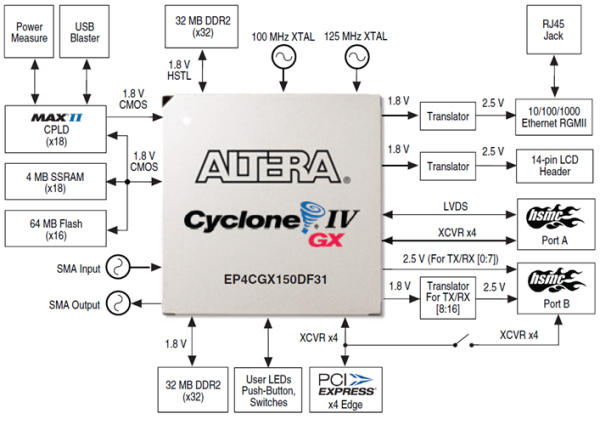

Figure 4
Intel Cyclone IV EP4CE15 board

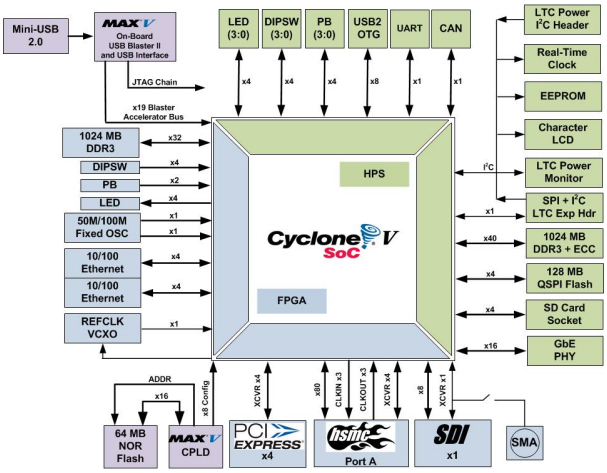

Figure 5
Intel Cyclone V SoC board

The tested models will come from open source software code repositories. NeuroCoreX is included as the main example (refer Gautam et al., 2025, Oak Ridge National Laboratory, n.d.). The x, z, and

C values will be changed in a systematic order. While during one parameter sweep, the other two values are assigned to the baseline config. state. Which means that, one main value changes during that process of testing. The tested amount starts from a smaller and more densely interconnected network with thousands of artificial neurons. This is relevant for memory retrieval algorithms and interconnected processing tests. The ”high density state” is approx. 50,000 neurons with a constant value of distribution or radiation of approx. 128 synapses. According to Intel Cyclone documentation, this is nearby the maximum processing limit of the Cyclone V board through a layered config. And the exact neuron and synapse amount for the lower density condition will depend on the connectivity process already supported by the selected open source model.

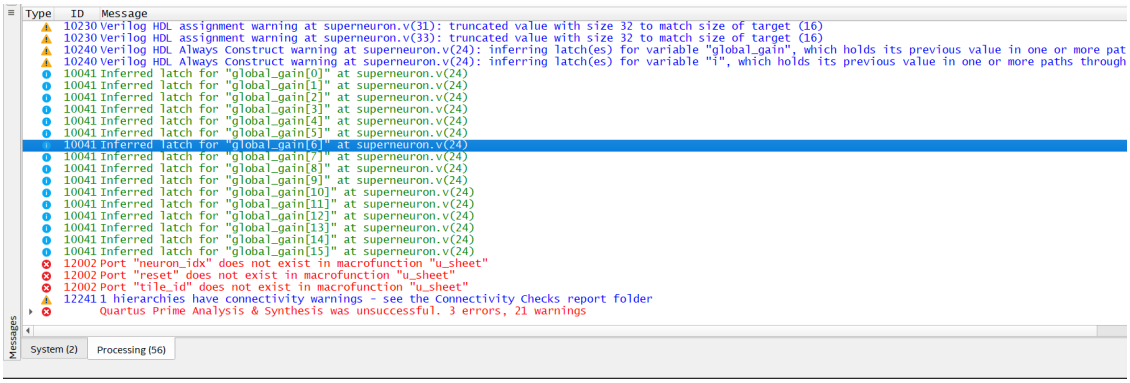

Figure 6: Intel Quartus Prime Lite Edition - Terminal Output

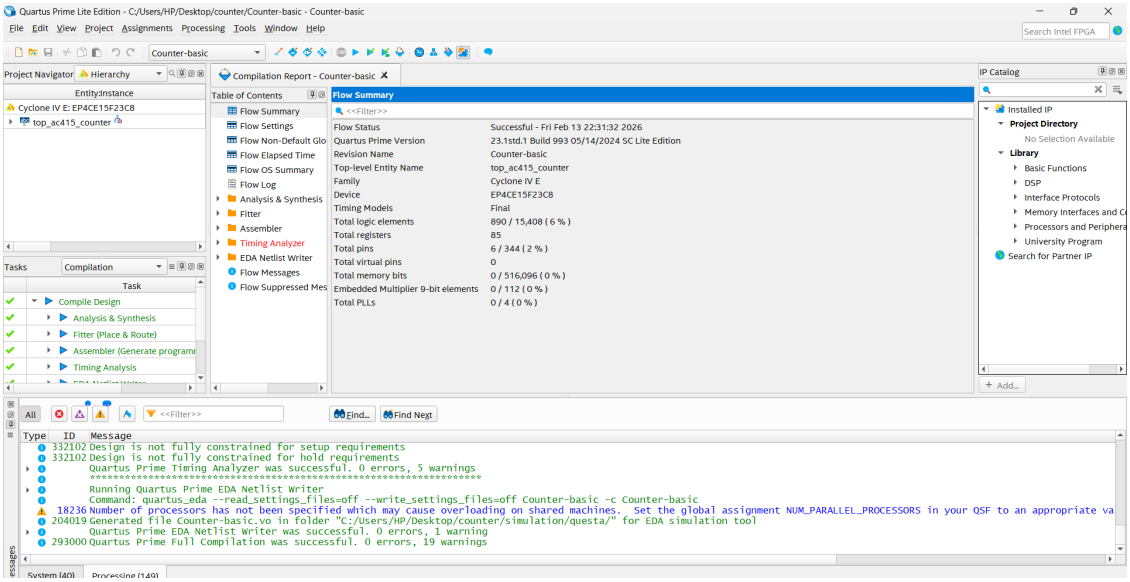

Figure 7: Intel Quartus Prime Lite Edition - Analysis and Synthesis, Fitter, Assembler, and Timing Analysis flows

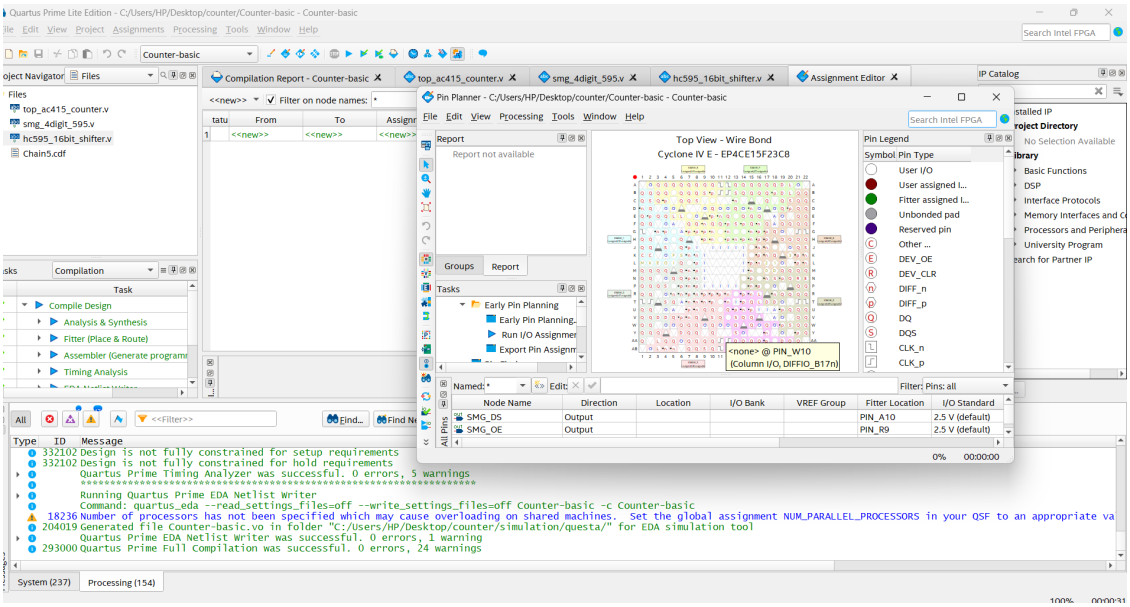

Figure 8: Intel Quartus Prime Lite Edition - Custom Pin Planner with Configs enabled.

Every config. will run the four task types previously mentioned. Four main outputs will be saved to the windows computer.

1. The biological accuracy values compared with the NEST or Brian2 software biological accuracy values.
2. The normal clock cycle time.
3. This will be consuming power dynamically. The QPF Power Analyzer in Intel Quartus Prime will give the software value. A physical multimeter measurement will be used as a second checking process (or an alternative physical measurement method).
4. The Real Time Ratio - This is defined as the simulated biological time divided by the normal time.

Every config. will run a greater amount of cycles. We will also need to average the output values in order to reduce the noise and the gamma. But the logs per each cycle will be stored as well. This is important because a temporary or rare degradation process can return and disappear inside the average value. (reference Ostrau et al., 2022).

The complete parallel software simulation is the zero multiplexing control for every hardware configuration. At a smaller scale, one physical processing element for every simulated neuron will be used when the hardware allows it. This assigns to a second control condition. It separates a general hardware or fixed point quantization difference from a difference which is created by this time multiplexing method.

This section is for expected results. The biological accuracy is expected to reduce when x and C variables increase when compared to variable z. Meaning that, a greater number of neuron

updates will be compressed with a compression algorithm into a similar amount of sequential time blocks. The power usage will be increased when C var increases. The power usage per each simulated neuron will be reduced when x increases. As mentioned, this is the advantage of the time multiplexing. If the biological accuracy is constant and similar in the array of tested instances, this is also a relevant result. This result will be able to depict that the tested open source systems have not reached their practical multiplexing limit in that range. Another important point is that, the Quartus Prime power value and multimeter reading values can be different. If this difference is very noticeable, both of those values will be added to the text records.

Assessing Performance Limits for each Task Type

The dataset from aim 2 will be analyzed separately for the four task types. For each task, the biological accuracy trend will match to the configurations. The analysis finds the configs where the fitting accuracy value first moves below the Research Aim 1 margin value θ.

For each task type, a monotonic sigmoid curve will connect the biological accuracy value with the multiplexing density. The multiplexing density is written as $x \cdot C$ while z is kept at its feasible minimum value. The degradation margin value is the density where the fitted curve crosses θ.

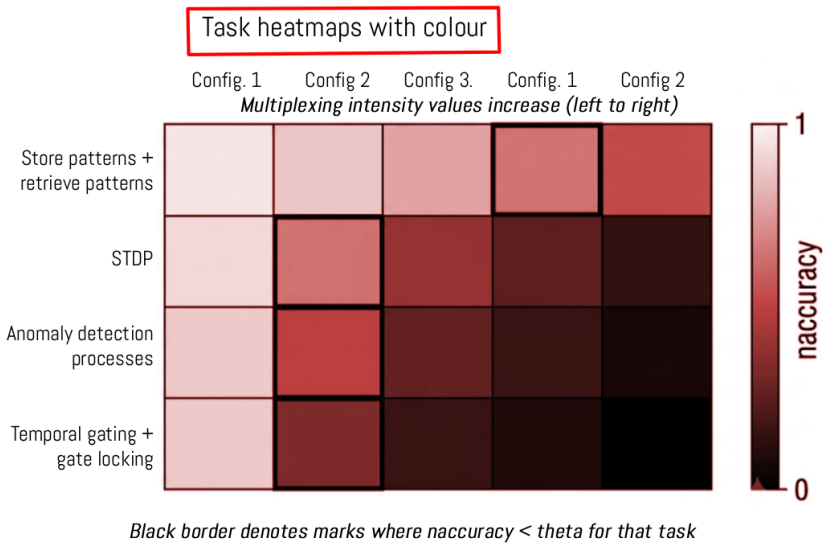

Figure 9: Configs By Task Heatmap

Timeline of Creating the SuperNeuronBench

In this research, access to hardware, software is available and is sufficient.

The Intel Altera Cyclone IV and Cyclone V FPGA boards are already available as Hardware. A Windows 11 computer will be used to measure and project results, as well as handle benchmark website updates, statistics, publications, etc. In terms of software resource availability, The Intel Quartus Prime Lite is already installed on a Windows 11 computer.

This has three separate phases. The first phase creates the benchmark and the software reference. The other task types, accuracy measures, and margin values will be completed in NEST or Brian2. In the second phase, we collect the data outputs from hardware. Then the parameter sweeping processes will be completed on the Intel Cyclone IV EP4CE15 board and the Cyclone V SoC board. Lastly, we analyze them and publish all the outputs. Degradation margin values will be measured and calculated, and the open benchmark ranking will be published on 🌐 `https://superneuronbench.nekshadesilva.com`.

Data Output From the Research

Every config will have a stored accuracy value, time log CSV file, power usage (per cycle) log .CSV file. At the same time, the Intel Quartus Prime project files folder will be uploaded with all .QSF files. This will enable another researcher to identify how those results were produced. The raw aim 2 datasets will also be released. In addition to that, the Quartus prime fitting and analysis fabric files will also be released. Those outputs will be available on the Git repository to download at 🌐 `https://superneuronbench.nekshadesilva.com/git`. This method allows another researcher to add a separate task, test another open source neuromorphic model, etc.

Limitations and Disadvantages

There are multiple limitations in the SuperNeuronBench. The first is the hardware dependency. These results are more native to the Intel Cyclone chip set family and the other open source models previously mentioned. This benchmark may not be compatible with other FPGA families such as AMD Versal, Intel Agilex, AMD Virtex, AMD UltraScale+ and many others. The second reason is the number of open source and actively maintained neuromorphic models. Unlike other industries, this is very limited. Due to that reason, these margin values might not represent other model architectures. The third reason is the power measurement. Software based power values are not always equal to the readings from analog or digital multimeters, ammeters, voltmeters, etc. especially when this switching activity is greater.

References and Citations

Davies, M., Srinivasa, N., Lin, T.-H., and others. 2018. Loihi, a neuromorphic manycore processor with on chip learning. *IEEE Micro*, 38(1), 82–99. https://doi.org/10.1109/MM.2018.112130359

DeBole, M. V., Taba, B., Amir, A., and others. 2019. TrueNorth, accelerating from zero to 64 million neurons in 10 years. *IEEE Computer*, 52(5), 20–29. https://doi.org/10.1109/MC.2019.2903009

Furber, S. B., Galluppi, F., Temple, S., and Plana, L. A. 2014. The SpiNNaker project. *Proceedings of the IEEE*, 102(5), 652–665. https://doi.org/10.1109/JPROC.2014.2304638

Gautam, A., Date, P., Kulkarni, S., Patton, R., and Potok, T. 2025. NeuroCoreX, an open source FPGA based spiking neural network emulator with on chip learning. *arXiv preprint*. https://arxiv.org/abs/2506.14138

Gewaltig, M.-O., and Diesmann, M. 2007. NEST, or NEural Simulation Tool. *Scholarpedia*, 2(4), 1430.

Intel Corporation. n.d. Cyclone V device overview. https://docs.altera.com/r/docs/683801/current/cyclone-v-device-datasheet

Intel Corporation. n.d. Cyclone IV device family overview. . https://www.intel.com/content/www/us/en/products/sku/210469/cyclone-iv-ep4ce15-fpga/specifications.html

Merolla, P. A., Arthur, J. V., Alvarez-Icaza, R., and others. 2014. A million spiking neuron integrated circuit with a scalable communication network and interface. *Science*, 345(6197), 668–673. https://doi.org/10.1126/science.1254642

Oak Ridge National Laboratory. n.d. NeuroCoreX GitHub repository. https://github.com/ORNL/NeuroCoreX

Ostrau, C., Klarhorst, C., Thies, M., and Rückert, U. 2022. Benchmarking neuromorphic hardware and its energy expenditure. *Frontiers in Neuroscience*, 16, 873935. https://doi.org/10.3389/fnins.2022.873935

Sievers, M., Motta, A., Schmidt, M., and others. 2024. Connectomic reconstruction of a cortical column. *bioRxiv*. https://www.biorxiv.org/content/10.1101/2024.03.22.586254

Stimberg, M., Brette, R., and Goodman, D. F. M. 2019. Brian 2, an intuitive and efficient neural simulator. *eLife*, 8, e47314. https://doi.org/10.7554/eLife.47314

van Rossum, M. C. W. 2001. A novel spike distance. *Neural Computation*, 13(4), 751–763. https://doi.org/10.1162/089976601300014321

Victor, J. D., and Purpura, K. P. 1996. Nature and precision of temporal coding in visual cortex, a metric space analysis. *Journal of Neurophysiology*, 76(2), 1310–1326. https://doi.org/10.1152/jn.1996.76.2.1310

Yik, J., and others. 2025. The NeuroBench framework for benchmarking neuromorphic computing algorithms and systems. *Nature Communications*. https://doi.org/10.1038/s41467-025-56739-4