Neuromorphic Architecture for Small-Scale Neocortical Network Emulation

Neuromorphic Architecture for Small-Scale Neocortical Network Emulation

Ziyao Zhang and Jayawan Wijekoon
The Department of Electrical and Electronic Engineering
The University of Manchester
Manchester, UK
Email: Ziyao.Zhang@manchester.ac.uk; Jayawan.Wijekoon@manchester.ac.uk

***Abstract*—The paper presents a neuromorphic platform that can emulate a small-scale cortical network with diverse types of neurons and synapses found in cortical circuits. The platform provides configurable neurons, which behave similarly to the electrophysiological behaviours of different classes of pyramidal and interneurons, and configurable long- and short- term dynamic synapses that can provide inhibition, excitation, weight depressing and facilitating and spike-time dependent plasticity (STDP) dynamics. The prototype of the platform presented in this paper uses a single Cortical Neural Layer (CNL) integrated circuit (IC), which facilitates a network of 120 neurons and 7560 synapses. The number of CNL ICs used in the proposed architecture can be increased to enable larger neural network emulation. The network connectivity is configured using an off-chip Field Programmable Gate Array (FPGA) device. The parameters of the neural elements of the network can be configured using a computer-controlled bias voltages generator. To prove the concept in hardware, Winner-Take-All and Synfire chain networks have been implemented on the platform, and the results are presented.**

Keywords—neuromorphic architecture; silicon neuron; silicon synapse; neural network; neocortex.

I. Introduction

Development of brain-inspired custom Very Large Scale Integration (VLSI) ICs and systems that implement hundreds to thousands of spiking neurons and synapses has become an increasing interest of many research groups [1-5]. The electronic models of neurons and synapses are implemented by mimicking biophysically realistic dynamics at different levels of abstraction to meet various hardware implementation constraints. These systems emulate functions of a cortical network in real-time or faster than real-time to demonstrate real-world behaviours to understand the underlying computational principles in brain computations. Currently, spiking neural network computations are used in neuroscience, robotics, and computer vision applications [3, 6-13].

In this paper, considering electrophysiology of neurons and synapses and details of the neocortical circuits of the brain, we propose a biologically plausible spiking neural network platform, particularly customised to emulate small-scale neocortical microcircuits. This architecture enables emulation of rich neural dynamics observed in cortical circuits, beyond conventional VLSI mixed-signal neuromorphic hardware implementations. The platform is capable of mimicking fast spiking, non-adapting non-fast spiking (i.e., regular spiking) adapting, irregular spiking and intrinsic burst firing neuron dynamics (for standard definitions of these dynamics, see Petilla terminology [14]), and synaptic depression, synaptic facilitation, spike-timing-dependent-plasticity, homeostatic synaptic plasticity, inhibitory and excitatory synaptic dynamics (see [15] for more details). The proposed platform uses a previously developed Cortical Neural Layer (CNL) integrated circuit (IC) to provide analogue neural circuits.

II. System Overview

The proposed neocortical network emulating platform consists of a CNL IC to provide neurons and synapses, an FPGA to facilitate the connectivity of the network, bias voltages generator to provide configurable parameters of the neural circuits and a computer-aided controller to monitor and coordinate the operations of the platform. Functions of the computer-aided controller include programming the neuron connectivity configuration on to the FPGA, configuring neural parameters of the network by tuning the bias voltages generator, acquiring and displaying the spike activities of the network and monitoring and storing the internal neural dynamics of the network by controlling an oscilloscope. The system overview of the network emulating platform is shown in Fig 1.

The hardware prototype demonstrated in this paper has 120 accelerated-time Izhikevich VLSI neuron circuits [17], 5460 different types of short-term synapses and 2100 long-term (STDP) synapses. Types of short-term synapses include 420 Excitatory Facilitating (EF) Synapses, 420 Inhibitory Facilitating (IF) Synapses, 1700 Inhibitory Depressing (ID)

Synapses, 300 somatic Inhibitory Depressing (sID) Synapses and 2620 Excitatory Depressing (ED) Synapses [16, 18]. These neuron and synapse numbers available to emulate a network can be increased by integer multiples up to sixfold by adding up to six CNL ICs to the proposed platform.

A neuron circuit can be configured to many different neuron types (such as fast spiking, non-adapting non-fast spiking, adapting, irregular spiking and intrinsic burst firing neuron dynamics, etc.) using its two externally tunable bias voltages, and the shape of the STDP curve of the long-term synapse can be configured using its four externally tunable bias voltages. In the short-term synapse circuits, the strength of weight depression or facilitation, speed of weight recovery, resting weight and post-synaptic current scale (synaptic scaling) can be configured using externally controllable voltages; Details of the synaptic dynamics and their respective tuning parameters can be found in [15-16, 18]. These neuron and synapse combinations can be used to emulate complex small-scale cortical networks in the hardware. Further, the internal dynamics of synapses belonging to eight neurons can be monitored and recorded using an external oscilloscope. The platform can emulate a cortical network with heterogeneous synapses and diverse types of neurons at a speed of up to five orders of magnitude faster than biological real-time.

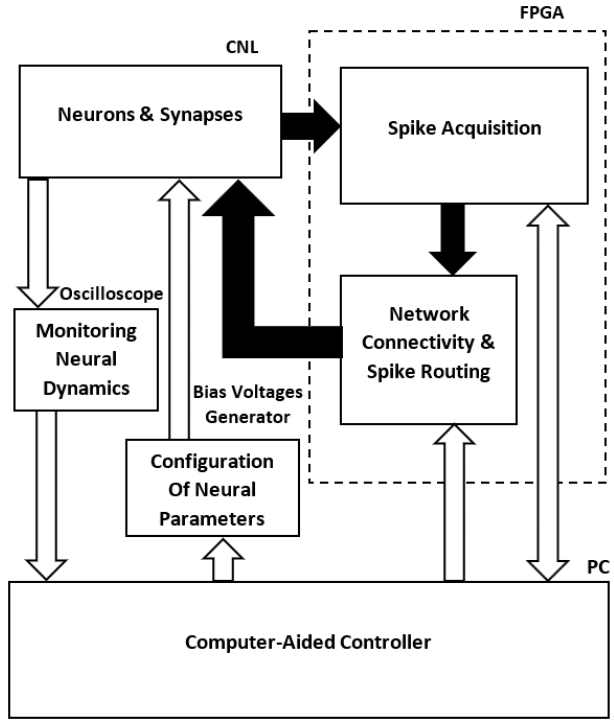

Fig. 1 Block diagram of the spiking neural network emulation platform.

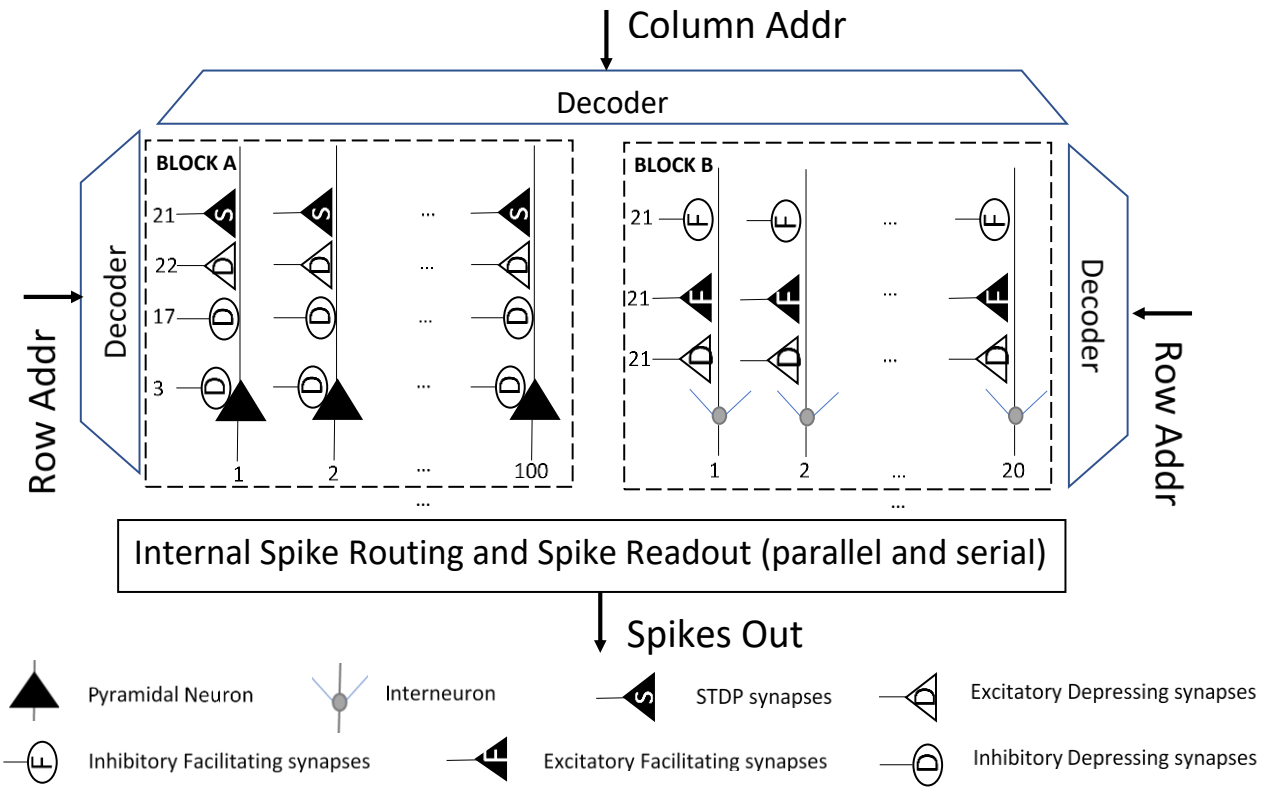

Fig. 2 Diagram showing the neurons and synapses on the CNL IC; The *Row Addr* and *Column Addr* used to select interested synapse group and generate presynaptic spikes. There are 100 excitatory neurons (Pyramidal Neurons) in Block A, and 20 inhibitory neurons (Interneurons) in Block B each neuron connected to 63 synapses; Note: Configurable parameters of the neurons and synapses are not shown here. Outputs of 20 Inhibitory Depressing synapses connected to a Block A neuron, 3 synapses are capable of generating higher postsynaptic currents to the membrane to mimic somatic Inhibitory Depressing (sID) synapses.

III. Implementation Details of The Proposed Architecture

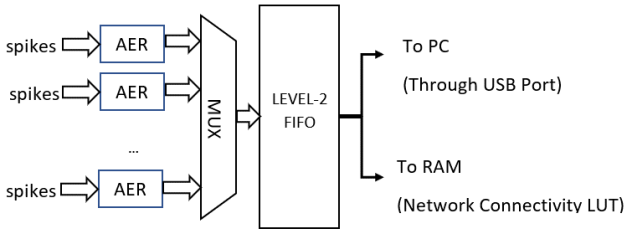

Fig. 3 Block diagram of the Spike Acquisition circuit which includes several AER (Address-Event Representation) cells, a round-robin MUX and a FIFO; details of an AER cell is shown in Fig 4. The FIFO output is sent to PC (through USB to record spike events of the network) and to the RAM-1 of the Network Connectivity and Spike Routing circuit (shown in Fig 5), which is responsible for sending spike events to connected synapses on the post-synaptic neurons.

The CNL IC provides 120 neuron units where each neuron receives input from 63 dedicated synapses, as shown in Fig 2. An FPGA facilitates the connectivity between these neuron units using the Spike Acquisition circuit and the Network Connectivity and Spike Routing circuit shown in Fig 3 and 5. In the prototype, the FPGA receives spikes from the neurons using parallel outputs. The incoming spikes received from the neurons (from the "axons of the presynaptic neuron") are acquired by the Spike Acquisition circuit, and they are routed to the designated synapses (on the "dendrites of the post-synaptic neurons") with the help of the Network Connectivity and Spike Routing circuit. When implementing a neural network on the platform, the connectivity data of the network is added to the look-up-table (LUT) of the Network Connectivity and Spike Routing circuit to facilitate intended spike routing.

A. Spike Acquisition

The Spike Acquisition circuit shown in Fig 3 is implemented on the FPGA to acquire spike outputs received from the CNL IC. It is comprised of Address-Event Representation (AER) cells, multiplexer (MUX) and a First-In-First-Out (FIFO) register. When a spike arrives at the AER cell, it generates an address event by capturing the addresses of the fired neurons and the time-stamps of the spikes.

A block diagram of an AER cell is shown in Fig 4. Each AER cell comprises 8 RS latches, MUX, Address-Event Generation (AEG) circuit and a FIFO register. The spikes received from up to 8 neurons are latched on RS latches and transferred to the AEG circuit using the round-robin technique implemented on the MUX. The AEG circuit generates the address event data corresponding to each spike received from the MUX. Finally, the data is transferred to LEVEL-1 FIFO.

Several AER cells are implemented on the Spike Acquisition circuit to receive outputs from many neurons. When there are 120 neurons in the network, 15 AER cells are used. The number of AER cells can be expanded to accommodate more CNL ICs on the platform.

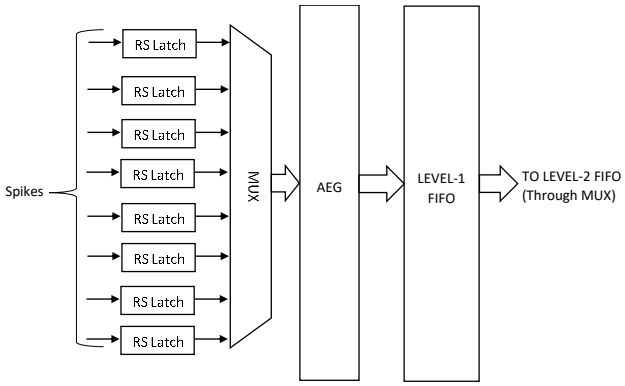

Fig. 4 Block diagram of an AER circuit: It comprises 8 RS latches which capture spikes received from 8 neurons, a MUX, Address-Event Generation (AEG) circuit and LEVEL-1 FIFO register. Output of the FIFO is sent to LEVEL-2 FIFO through the MUX shown in Fig 3.

The address events saved in the LEVEL-1 FIFOs are transmitted to the LEVEL-2 FIFO through the MUX using the round-robin technique. The data accumulated in the LEVEL-2 FIFO is used to send presynaptic spikes to the synapses of the post-synaptic neurons, based on the connectivity table implemented in the RAM-2 shown in Fig 5. Additionally, these incoming address events on the LEVEL-2 FIFO are sent to a Personal Computer (PC) to observe the firing activities of the network.

B. Network Connectivity and Spike Routing

When the incoming address event arrives from the LEVEL-2 FIFO, it is routed to the respective synapses using the circuit shown in Fig 5. The circuit consists of two random access memory (RAM) blocks (labeled as RAM-1 and RAM-2) and a Presynaptic Addressing Protocol circuit. A LUT that provides the mapping of neurons to the synapses on the postsynaptic neuron is stored in RAM-2. Some incoming spikes from a neuron need to be routed to many synapses. Hence, one or more LUT rows can be allocated to store a block of Destination Synapses (DS) addresses (format of a single DS address, where many synapses can be addressed using a single address, is discussed in the next paragraph) corresponding to each neuron. Each row on RAM-1 contains two entries related to the stored locations of a block of DS addresses. They are the location on the RAM-2 where the first DS address (data section denoted by 'a' in Fig 5) is stored, and the number of rows (data section denoted by 'b') allocated to the block. Hence, the number of elements in RAM-1 is equal to the number of neurons in the network, and the size of RAM-2 is determined by the size of the network and complexity of the network connectivity. In addition to providing the number of DS address rows for a single incoming spike event, each address on RAM-2 can deliver presynaptic spikes to multiple synapses using the Multiple Synapses Addressing strategy implemented on the CNL IC.

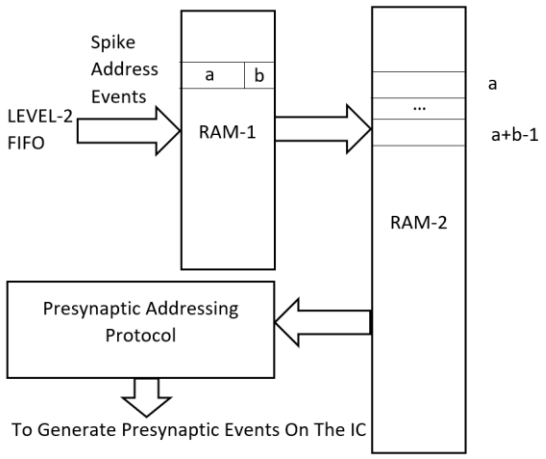

Fig. 5 Block diagram of the Network Connectivity and Spike Routing circuit. RAM-1 receives address events of spikes from a LEVEL-2 FIFO (shown in Fig 3); When an incoming address event is received from a neuron, its corresponding block of Destination Synapses (DS) addresses stored in the RAM-2 is read via RAM-1. Each row in RAM-1 holds two entries: the location of where the first address (e.g. 'a') of the block is stored in RAM-2 and the number of rows (e.g. 'b) allocated to the block. Once the DS addresses are read, the Presynaptic Addressing Protocol circuit sends signals to the CNL IC to generate a presynaptic event in all the DS.

The CNL IC uses an address-event representation framework with a Multiple Synapses Addressing strategy to receive presynaptic inputs efficiently. A single DS address received by the IC consists of 14 bits to identify the postsynaptic neuron/s (*Column Addr*) and 12 bits to identify synapse row number/s (*Row Addr*). When a single pair of *Column Addr* bits and *Row Addr* bits are received by the Decoders in the IC (shown in Fig 2), the corresponding batch of synapses is selected immediately, and the internal circuit of each selected synapse generates a presynaptic spike. As shown in Fig 6, the *Column Addr* consists of *Main Address* (the seven least significant bits) and *Don't care Address* (the seven most significant bits, where each bit corresponds to a bit in the *Main address*). At the Decoder circuit, if a *Don't care Address* bit is set to logic one, it replaces the corresponding bit of the *Main Address* with "*Don't care*" status (i.e the bit can be both logic 1 and 0) and concurrently selects both the neuron columns. The same strategy is used to select synapse rows using *Row Addr*. Many *Don't care Address* bits of *Column Addr* and *Row Addr* can be set in one DS address to send presynaptic spikes to many synapses simultaneously.

The off-chip DS address (that consists of the *Column Addr* and *Row Addr* of the designated synapses) is sent to the IC using three steps. The timing diagram of the addressing protocol used to send presynaptic spikes is shown in Fig 7. Initially, the target *Column Addr* is sent to the address (*Addr*) bus of the IC, and then a pulse is sent to the *Stb1_c* pin of the IC to register the *Column Addr*. Secondly, the target *Row Addr* is sent to the *Addr*, and a pulse is sent to the *Stb2_c* and *Stb_r* pins. Finally, a pulse is sent to the *Stb* pin of the IC to generate presynaptic input spikes in all the selected synapses. This presynaptic addressing protocol (shown in Fig 5) is implemented on the FPGA.

C. Delay Performance

The total time taken to send presynaptic spikes to the IC after receiving spikes from a neuron (i.e. routing delay) depends on the size of the network. The routing delay with worst-case input conditions can be estimated by assuming all the neurons fire simultaneously at the fastest rate, and the FPGA clock rate is set to 200 MHz. The routing delay is created by four main sub-delays. Firstly, capturing of incoming spikes from the RS latch takes from 10 ns to 80 ns depending on the size of the network. Then, the time taken to transfer the data from LEVEL-1 FIFO to the LEVEL-2 FIFO is proportional to the number of neurons in the network, where each neuron introduces an additional 5 ns transfer time (the worst-case delay to transfer address event between FIFOs). Thirdly, the time taken to read the target address in RAM-2 through RAM-1 is 10 ns; Each additional target address will introduce an additional 5 ns. Finally, the delay in transmitting a single DS address to the CNL IC takes 17.5 ns. Hence, assuming we could use a single DS address each to represent all the target synapses corresponding to each presynaptic neuron, the total worst-case delay can vary between 42.5 ns and 707.5 ns; if a postsynaptic neuron needs more than one DS row, each additional DS address will introduce an extra 10 ns delay. The total delay is much smaller than the refractory period of the neuron (which is approximately 1 µs) and time windows of STDP synapses. As the average spike activities of a network are much less than what is assumed above, the average delay should be much smaller. With 120 neurons in a network, the platform can handle up to 100 million address events per second; however, as Multiple Synapses Addressing strategy is used, one address event can send presynaptic spikes to many synapses simultaneously. Additionally, by carefully mapping the locations of neurons and synapses on the IC to the network and by identifying optimised network connectivity LUT, the total spike routing delays could be reduced.

D. Configuration of Neural Parameters

To configure the neural properties of the network implemented on the platform, neuron and synapse dynamics need to be configured using a computer-aided bias voltages generator. All the synapses of the same type are configured using the common set of externally tunable bias voltages. The short-term plasticity (weight facilitation or depression) of the short-term synapses can be switched-off to use as a simple weight dependent synapse. Depending on the types of synapses connected to a neuron, the neurons are divided into two blocks, block A and block B, as shown in Fig 2.

Further, for the purpose of configuring neurons to a particular type, the neurons are divided into 13 groups. All the neurons in a group can be configured to a specific type by setting the tuning parameters of neurons [16]. The parameters and spiking frequency of different types of neurons are given in [16].

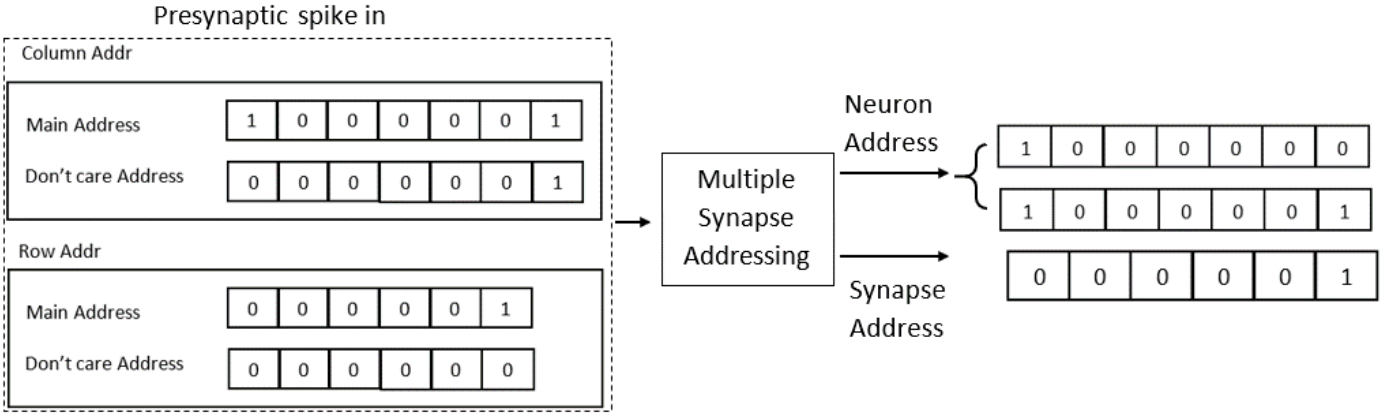

Fig. 6 Diagram illustrating the Multiple Synapses Addressing strategy used to send presynaptic spikes to the synapses; E.g. in the *Column Addr*, the 1st bit (least significant bit) of the *Don't care Address* corresponds to the 1st bit of the Main Address; If the 1st bit of the *Don't care address* is a logic one, the value of 1st bit in the *Main Address* is set to "*Don't care"* status. I.e the 1st bit of the *Main Address* can be both logic one and logic zero; Hence, the address will target synapses of two columns identified by addresses "1000000" and "1000001". Same strategy is used for the *Row Addr*.

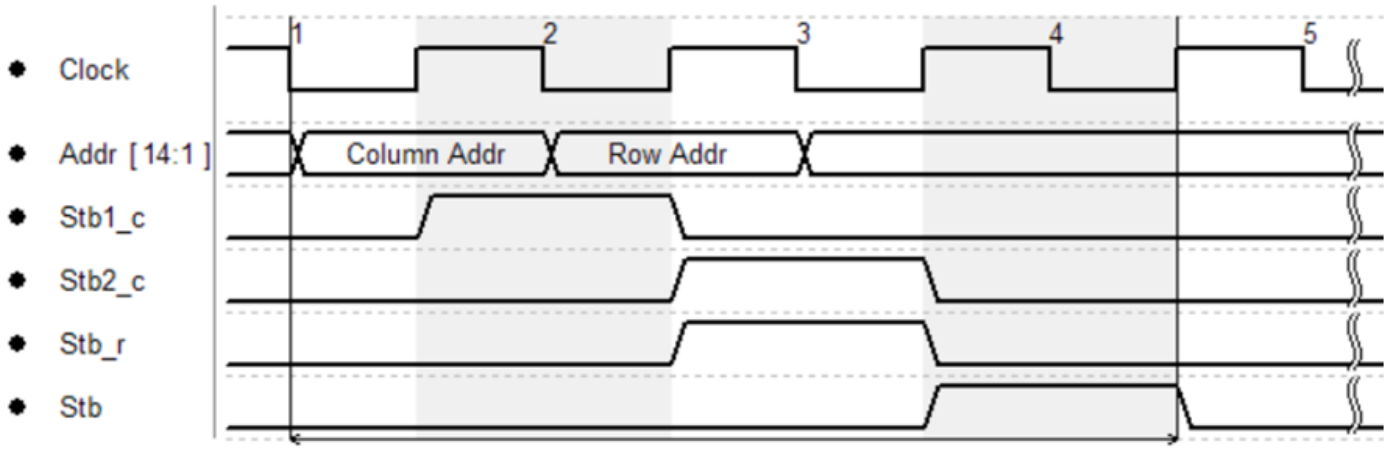

Fig. 7 The timing diagram of the off-chip addressing protocol used to send presynaptic input to the synapses. Two addresses, the *Column Add* and *Row Addr* contained the addresses of the synapses to which the presyanptic spike need to be sent. *Stb1_c*, *Stb2_c*, *Stb_r* and stb are pins in CNL IC that control the address bus (*Addr*). It takes 3.5 clock cycles to transmit a *Column Addr* and a *Row Addr* to the CNL chip. With a pipeline, the FPGA can transmit a *Column Addr* and a *Row Addr* every two clock cycles.

E. Limitations

The delay between the input and output of the FPGA is not constant, and the value depends on the size and complexity of the implemented network. As discussed, when 120 neurons are used in a network, the platform can handle up to 100 million address events per second within the refractory period of a neuron. However, as Multiple Synapses Addressing strategy is used, this value is much higher than 100 million, and the value depends on the connectivity of the network and how the DS addresses are represented in the LUT. To optimise FPGA routing efficiency of the platform, the DS addresses (network connectivity entries) in the RAM-2 (shown in Fig 5) need to be decided rationally. When the connectivity of a network is complex, it is hard to find the optimum representation manually. Hence, an algorithm for determining DS addresses is necessary.

In the prototype, the neuron spike activities observed by transferring them to the PC through a USB 2.0 port. When the number of spikes is more than 10 million spikes per second, the current design cannot transfer all the data. To solve this problem, a USB 3.0 port can be used to replace the USB 2.0, or some data can be saved in DDR2 SDRAM before transferring to a PC.

IV. NETWORK

This section presents implementations of Winner-Take-All (WTA) and Synfire chain networks on the prototype to demonstrate the emulation of networks on the proposed platform.

A. Winner-Take-All

The WTA network is a fundamental computational block used in many models of cortical processing [19-20]. The topology of the implemented network is shown in Fig 8. Each Excitatory Neuron has excitatory connections to two nearby Excitatory Neurons and to three Inhibitory Neurons in the centre. The three Inhibitory Neurons have inhibitory connections to all four surrounding Excitatory Neurons. Excitatory Neuron 1 and 4 receive external inputs from the Input Neuron A and B, respectively. Excitatory Neuron 1 and 4 are output neurons of the network.

In the initial experiment, as shown in Fig 9, the Excitatory Neuron 1 is given a fixed input spiking rate of 100 kHz; Excitatory Neuron 2 is given 25 kHz, 400 kHz and 25 kHz input spiking rates to observe the WTA network activities. All three Inhibitory Neurons have similar behaviour, so only the behaviour of one Inhibitory Neuron is shown in Fig 9. Excitatory Neuron 2 and 3 are inhibited during the operation and they do not exhibit noticeable firing activities. Excitatory Neuron 1 and Excitatory Neuron 4 have higher spike firing activities in turn, denoting the winning neuron of the WTA network. While the winning neuron exhibits an amplified firing rate, the other output neuron activities are suppressed. Fig 10 shows the behaviour of the output neurons of the network when the firing rate of one of the input neurons is fixed, and the other is varied continuously.

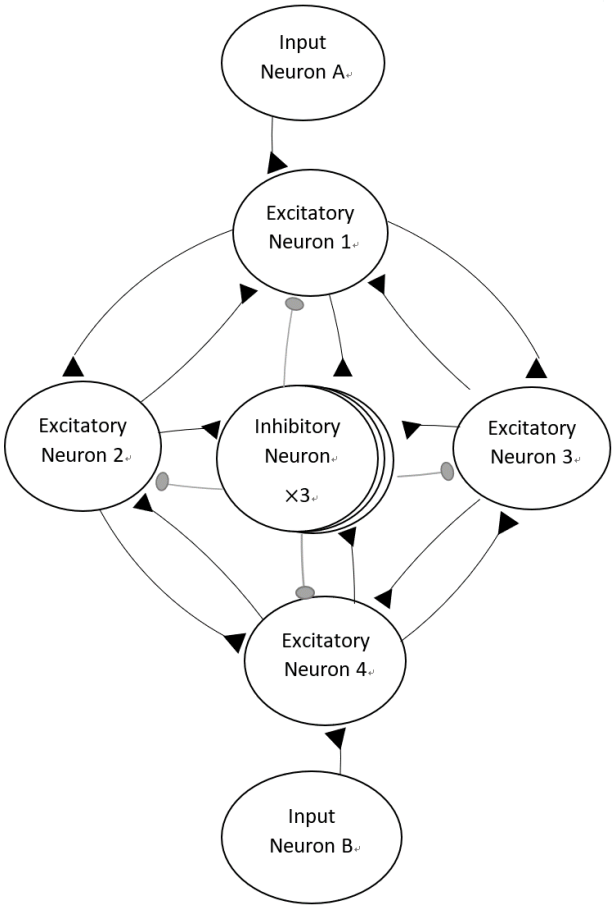

Fig. 8 Winner-Take-All network: Excitatory Neuron 1 and 4 receive inputs from Input Neuron A and B respectively. Three Inhibitory Neurons at the centre inhabit all four neighbouring Excitatory Neurons. Excitatory Neuron 1 and 4 are output neurons of the network.

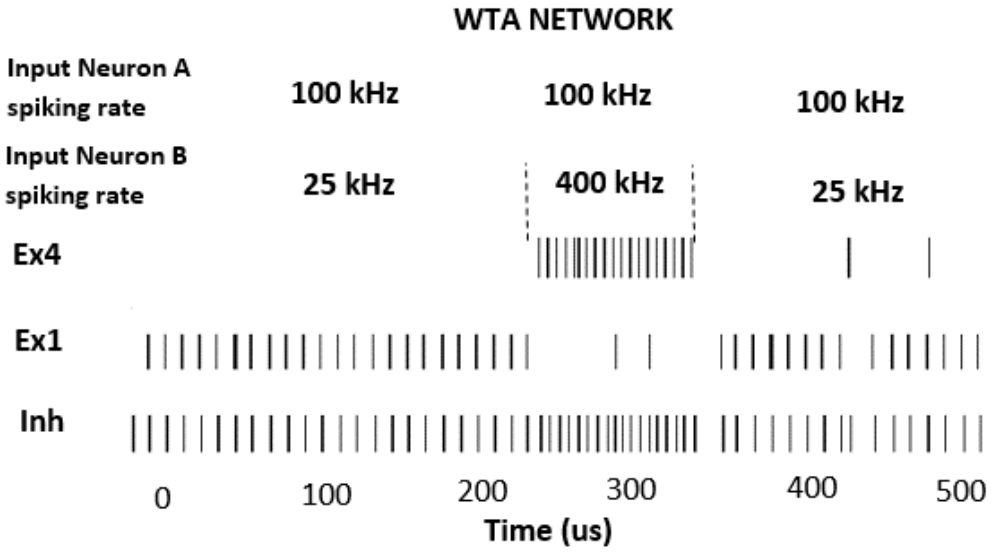

Fig. 9 Raster plot of the WTA network: 25 kHz, 400 kHz and 25 kHz input spiking rates are given to the Excitatory Neuron 4 and a 100 kHz fixed input spiking rate is given to Excitatory Neuron 1 to observe the WTA network activities. Ex1: Excitatory Neuron 1, Ex4: Excitatory Neuron 4, Inh: Inhibitory Neurons.

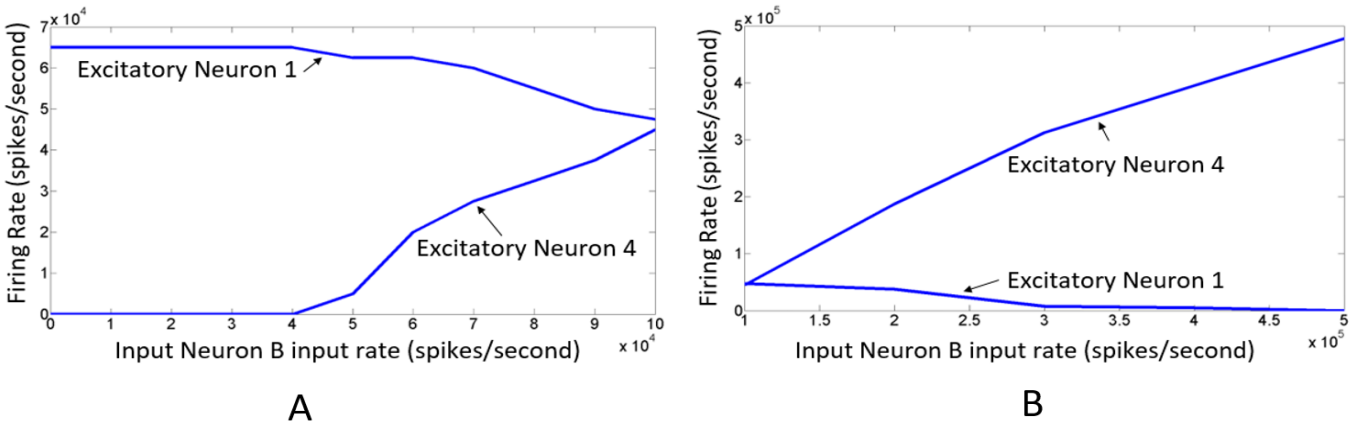

Fig. 10 Graphs shows the firing rate variation of the output neurons (Excitatory Neuron 1 and 4) of the WTA network, when neuron firing rate of the Input Neuron A kept at 100 kHz and firing rate of the Input Neuron B is varied; Graph A: the firing rate of Input Neuron B is lower than 100 kHz; Graph B: the firing rate of Input Neuron B is higher than 100 kHz.

B. Synfire Chain

An eight neuron Synfire chain [21] is implemented on the platform as shown in Fig 11 using four levels (L1, L2, L3, and L4). Input spike train is provided to the neurons at the lowest level (L1). Each level has two excitatory neurons. Each neuron at the lower level provides feed-forward excitatory connections to the two neurons at the higher level. For the same input spike train, if the neurons are configured to two different firing patterns (Regular Spiking and Intrinsic Bursting), Synfire chain spike propagations can be observed with distinct patterns and delays, as seen in Fig 12.

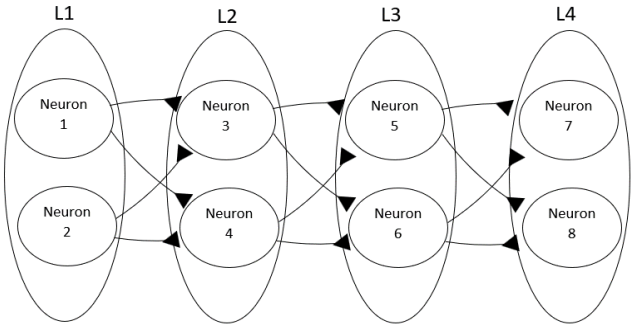

Fig. 11 Eight-neuron Synfire chain topology using four levels (L1 to L4); each neuron at the lower levels sends feed-forward excitatory connections to the two neurons at the higher level.

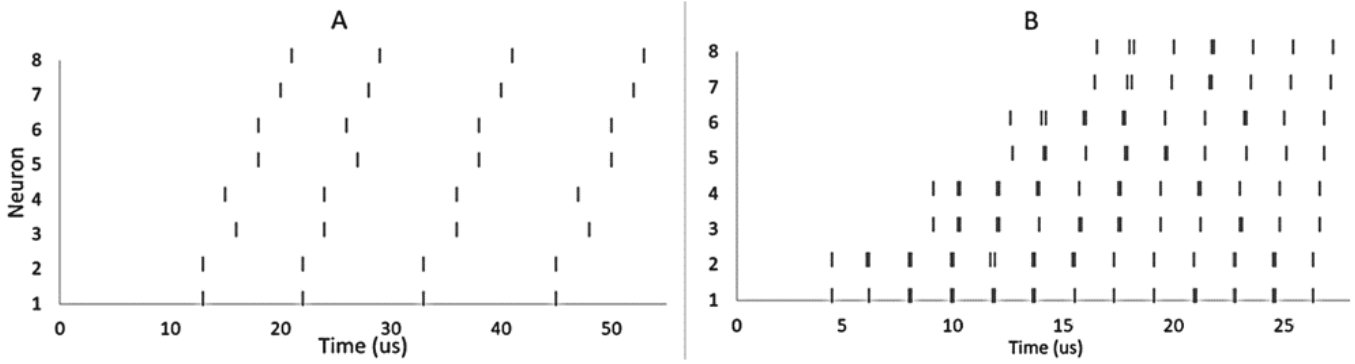

Fig. 12 Raster plots of the Synfire chain topology shown in Fig 11 for two different neuron parameter configurations. The non-adapting non-fast spiking neurons are used to obtain graph A, and the intrinsic burst firing neurons are used to obtain graph B; Other conditions remain the same.

V. CONCLUSIONS

This paper presents a small-scale neural network emulation platform. The functions of the platform have been experimentally verified using simple WTA and Synfire Chain networks. The platform can emulate real-time spiking neural networks utilizing diverse reconfigurable neural dynamics implemented on the CNL IC. The size of the network implemented on the platform can be scaled further by incorporating many CNL ICs on the platform.

REFERENCE

[1] J. H. Wijekoon and P. Dudek, "VLSI circuits implementing computational models of neocortical circuits," *Journal of Neuroscience Methods*, vol. 210, no. 1, pp. 93–109, 2012.

[2] S. Schmitt et al., "Neuromorphic hardware in the loop: Training a deep spiking network on the BrainScaleS wafer-scale system," *In 2017 International Joint Conference on Neural Networks (IJCNN)*, 2017.

[3] S. B. Furber, F. Galluppi, S. Temple and L. A. Plana, "The SpiNNaker Project," *Proceedings of the IEEE*, vol. 102, no. 5, pp. 652–665, 2014.

[4] Henry Markram, Karlheinz Meier, Thomas Lippert, Sten Grillner, Richard Frackowiak, Stanislas Dehaene, Alois Knoll, Haim Sompolinsky, Kris Verstreken and Javier DeFelipe, Seth Grant, Jean-Pierre Changeux, Alois Saria, "Introducing the Human Brain Project," *Procedia Computer Science*, vol. 7, pp. 39-42, 2011.

[5] T. Serrano-Gotarredona, T. Masquelier, T. Prodromakis, G. Indiveri and B. Linares-Barranco, "STDP and STDP variations with memristors for spiking neuromorphic learning systems," *Frontiers in Neuroscience*, vol. 7, 2013.

[6] C. D. Schuman, T. E. Potok, R. M. P., J. Douglas Birdwell, M. E. Dean, G. S. Rose and J. S. Plank, "A Survey of Neuromorphic Computing and Neural Networks in Hardware," *arXiv:1705.06963*, May 2017.

[7] D. Roggen, S. Hofmann, Y. Thoma and D. Floreano, "Hardware spiking neural network with run-time reconfigurable connectivity in an autonomous robot," *In Proceedings NASA/DoD Conference on Evolvable Hardware*, 2003.

[8] A. Bouganis and M. Shanahan, "Training a spiking neural network to control a 4-DoF robotic arm based on Spike Timing-Dependent Plasticity," *In 2010 International Joint Conference on Neural Networks (IJCNN)*, 2010.

[9] Y. Cao, Y. Chen and D. Khosla, "Spiking Deep Convolutional Neural Networks for Energy-Efficient Object Recognition," *International Journal of Computer Vision*, vol. 113, no. 1, pp. 54–66, 2014.

[10] T. Masquelier, R. Guyonneau and S. J. Thorpe, "Competitive STDP-Based Spike Pattern Learning," *Neural Computation*, vol. 21, no. 5, pp. 1259–1276, 2009.

[11] P. U. Diehl and M. Cook, "Unsupervised learning of digit recognition using spike-timing-dependent plasticity," *Frontiers in Computational Neuroscience*, vol. 9, 2015.

[12] T. Masquelier and S. J. Thorpe, "Unsupervised learning of visual features through Spike Timing Dependent Plasticity," *PLoS Computational Biology*, vol. preprint, no. 2007, 2005.

[13] E. S. Fortune and G. J. Rose, "Short-term synaptic plasticity as a temporal filter," *Trends in Neurosciences*, vol. 24, no. 7, pp. 381–385, 2001.

[14] Petilla Interneuron Nomenclature Group et al. "Petilla terminology: nomenclature of features of GABAergic interneurons of the cerebral cortex," *Nature reviews*, vol. 9, No. 7, pp. 557-68, 2008.

[15] A. Morrison, M. Diesmann and W. Gerstner, "Phenomenological models of synaptic plasticity based on spike timing," *Biological Cybernetics*, pp. 459-478, 2008.

[16] J. H. Wijekoon and P. Dudek, "VLSI circuits implementing computational models of neocortical circuits," *Journal of Neuroscience Methods*, vol. 210, No. 1. pp. 93-109, 2012.

[17] J. H. Wijekoon and P. Dudek, "Compact silicon neuron circuit with spiking and bursting behavior," *Neural Networks*, vol. 21, No. 2-3, pp. 524-534, 2008.

[18] J. H. Wijekoon and P. Dudek, "Heterogeneous neurons and plastic synapses in a reconfigurable cortical neural network IC," *In 2012 IEEE International Symposium on Circuits and Systems*, 2012.

[19] M. Oster, R. Douglas and S.-C. Liu, "Computation with Spikes in a Winner-Take-All Network," *Neural Computation*, vol. 21, no. 9, pp. 2437–2465, 2009.

[20] M. Lundqvist, A. Compte and A. Lansner, "Bistable, Irregular Firing and Population Oscillations in a Modular Attractor Memory Network," *PLoS Computational Biology*, vol. 6, no. 6, 2010.

[21] S. Schrader, M. Diesmann and A. Morrison, "A Compositionality Machine Realized by a Hierarchic Architecture of Synfire Chains," *Frontiers in Computational Neuroscience*, vol. 4, 2011.

[22] F. Akopyan, J. Sawada, A. Cassidy, R. Alvarez-Icaza, J. Arthur, P. Merolla, N. Imam, Y. Nakamura, P. Datta, G. Nam, B. Taba, M. Beakes, B. Brezzo, J. Kuang, R. Manohar, W. Risk, B. Jackson and Modha, D. (). "TrueNorth: Design and Tool Flow of a 65 mW 1 Million Neuron Programmable Neurosynaptic Chip," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 10, pp. 1537-1557, 2015.

[23] S. Furber, "Large-scale neuromorphic computing systems," *Journal of Neural Engineering*, vol. 13, no. 5, p. 051001, 2016.

[24] B. Benjamin, P. Gao, E. McQuinn, S. Choudhary, A. Chandrasekaran, J. Bussat, R. Alvarez-Icaza, J. Arthur, P. Merolla and K. Boahen, "Neurogrid: A Mixed-Analog-Digital Multichip System for Large-Scale Neural Simulations," *In Proceedings of the IEEE*, 2014, pp. 699-716.

[25] E. Painkras, L. Plana, J. Garside, S. Temple, F. Galluppi, C. Patterson, D. Lester, A. Brown and S. Furber, "SpiNNaker: A 1-W 18-Core System-on-Chip for Massively-Parallel Neural Network Simulation," I*EEE Journal of Solid-State Circuits*, vol. 48, no. 8, pp. 1943-1953, 2013.

[26] P. Merolla, J. Arthur, R. Alvarez, J. Bussat and K. Boahen,. "A Multicast Tree Router for Multichip Neuromorphic Systems," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 61, no. 3, pp. 820-833, 2014.