A Scalable FPGA-based Cerebellum for Passage-of-Time Representation

Junwen Luo*, Graeme Coapes*, Terrence Mak, Tadashi Yamazaki, Chung Tin# and Patrick Degenaar#

***Abstract*— The cerebellum plays a critical role for sensorimotor control and learning. However dysmertria or delays in movements' onsets consequent to damages in cerebellum cannot be cured completely at the moment. To foster a potential cure based on neuroprosthetic technology, we present a frame-based Network-on-Chip (NoC) hardware architecture for implementing a bio-realistic cerebellum model with 100,000 neurons, which has been used for studying timing control or passage-of-time (POT) encoding mediated by the cerebellum. The results demonstrate that our implementation can reproduce the POT functionality properly. The computational speed can achieve to 25.6 ms for simulating 1 sec real world activities. Furthermore, we show a hardware electronic setup and illustrate how the silicon cerebellum can be adapted as a potential neuroprosthetic platform for future biological or clinical applications.**

I. Introduction

The cerebellum critically mediates the precise timing of muscle activations to achieve smooth and robust motor control. Such representation of the passage-of-time (POT) over a range of tens to hundreds of milliseconds is essential for organizing movements of different body parts into a coordinated action[1]. Errors in POT encoding consequent to cerebellum damages can lead to dysmetria or delays in movement onsets in these patients[2]. A complete cure for such condition is still missing at the moment, while it is impacting millions of patients worldwide. To foster a potential cure based on neuro-prosthetic technology, an efficient computational platform that can favorably mimic the complex function of the cerebellar neural network will be important. Fig. 1 shows a conceptual closed-loop system for a cerebellar POT functionality prosthesis.

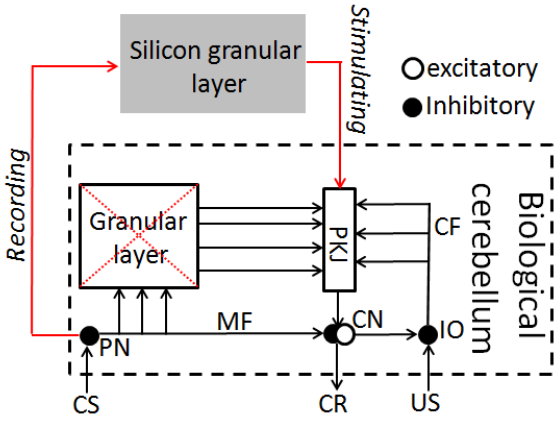

Fig. 1: The conceptual closed-loop system for cerebellum Passage-of-time (POT) prosthetic. CS is a conditional stimulus while The US is an unconditional stimulus. MF is the mossy fiber and CF is the climbing fiber, PKJ is the Purkinje cell.

Hardware implementations of cerebellum neural networks for neuro-prostheses have already attracted the interest of neuroscientists and engineers[3][4]. Bamford et al [3] has designed a VLSI field-programmable mixed-signal array to produce the eyeblink conditioning performances by modeling the cerebellum system. This has been fabricated as a core on a chip prototype intended for use in an implantable closed-loop prosthetic system aimed at rehabilitation of associated behavior. While they have demonstrated a proof-of-concept of success in their implementation, a highly simplified neural model with abstract modeling of cerebellar information processing is used in the work. Such simplification is convenient for hardware implementation, but when direct physiological correspondence for quantitative comparison with the biological system is required, such model becomes insufficient. In contrast, Yamazaki and Tanaka's model [5] is more biologically realistic and pays specific attention to the role of the granular-Golgi layer in timing and gain control by the cerebellar cortex to reproduce experimental results. However, this comes with the cost of a significant increase in the size and complexity of the computational model in order to produce a robust system behavior. As such, an efficient implementation is required to overcome these computational challenges, especially when real-time application is required.

In this work, we have developed an FPGA-based network-on-chip (NoC) hardware architecture for implementing the granular layer of random projection cerebellum model presented in [5]. Our design is potentially applicable for *in vivo* experimental or clinical application.

* Both authors contribute equally in this work.
# Corresponding authors
Junwen Luo is with School of Electrical and Electronic Engineering, Newcastle University, NE1 7RU, UK. He was also a visiting research student at Department of Mechanical and Biomedical Engineering, City University of Hong Kong, Hong Kong. (e-mail: j.w.luo@newcastle.ac.uk) .
Graeme Coapes and Patrick Degenaar are with School of Electrical and Electronic Engineering, Newcastle University, NE1 7RU, UK (e-mail:{ graeme.coapes; patrick.degenaar}@newcastle.ac.uk).
Terrence Mak is with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong (e-mail: stmak@cse.cuhk.edu.hk).
Tadashi Yamazaki is with The University of Electro-Communications, Tokyo, Japan (e-mail: ieee14@neuralgorithm.org).
Chung Tin is with Department of Mechanical and Biomedical Engineering; Centre for Robotic and Automation; and Centre for Biosystems, Neuroscience, and Nanotechnology, City University of Hong Kong, 83 Tat Chee Avenue, Hong Kong (e-mail: chungtin@cityu.edu.hk).

978-1-4244-7929-0/14/$26.00 ©2014 IEEE

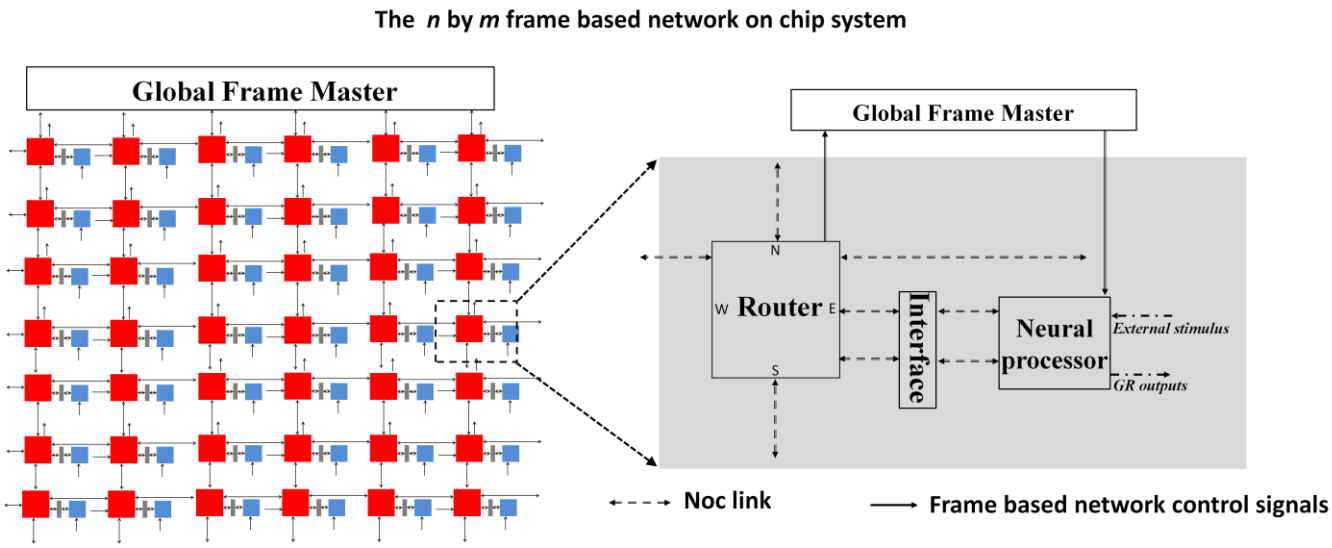

Fig. 2: A conceptual FPGA based network on chip hardware architecture. The figure on the left is the scalable n by m structure of frame based network on chip system. It contains n*m neural processors, n*m routers and one global controller. This architecture can be scaled up depending upon on the required model. In this paper, we implemented a network on chip system which contains 48 processors (100,000 neurons).

II. Hardware Architecture Design

To implement the POT model, we proposed a frame-based network on chip (NoC) hardware architecture on FPGA. The conceptual structure is shown in Fig.2. We implemented a NoC system containing 48 processors and a frame master.

A. Neural computing

The neural processor data path is shown in Fig. 3. Two types of neurons are implemented in the processor, the granule cell (GR) and the Golgi cell (GO). The neurons are modeled as conductance-based, leaky integrate-and-fire units. Both models use the same hardware architecture but with different parameters. Each granule cluster, containing 100 granule cells, connects to one Golgi cell. The activities (1 or 0) of all the 100 granule cells will be first calculated; whilst an accumulator will add all of them together and at the 100th clock cycle send the summated value to the Golgi cell model as an excitatory input.

Fig. 3B details the data path inside the neural model, which takes two computing stages: ion channel activities and integration. Each stage takes 4 clock cycles. Because of the parallel computational architecture, the latency in each individual path has to be consistent; therefore appropriate delay blocks (the rectangular blocks) are added as necessary. Fig. 3C shows the sub-component circuits, including the inhibition and excitation circuits and FIFO-based delay circuits. Since each neural processor implements 2000 granule cells and 20 Golgi cells, a pipelining technique is applied for reducing hardware resources. A long pipelining stage is required for storing granule cells calculation intermediate values. A First-In First-Out (FIFO) based delay circuit is designed for achieving long computational stages.

B. Network-on-chip

To manage the transmission of action potentials between Golgi cells and granule-cell clusters we have developed a NoC infrastructure. This system allows for arbitrary connectivity between Golgi cells and granule-cell clusters. Each processing element is connected to a router through which the action potentials are communicated. The routers are connected together in a mesh topology as shown in Fig. 4b.

When a Golgi cell produces an action potential the interface fetches a list of destination granule-cell clusters from memory, and an individual packet is generated to be sent to each of these destinations within the network. The connectivity of the neural network can be updated by adjusting the contents of the memory. A user may alter the contents of the memory to adjust the connectivity by injecting configuration packets into the network. This can be done at start-up or part way through simulation if required by halting the system using the global frame master.

The packet format is shown in the lower panel of Fig.4. Packets are classified by the setting of a 2-bit type identifier. The generated spike packet contains the address of the granular cell, allowing for the routers to direct the packet to the correct processing elements. Each granule-cell cluster summates the packets received. This value is used as an input into the granule-cell clusters. Packets are transmitted between routers using a 4-phase asynchronous protocol and a parallel data bus. The routers are output buffered using a 2-deep FIFO memory element.

C. Frame master

In order to maintain synchronicity within the system a frame master is used. The master is responsible for ensuring that all packets are transmitted to their destination before the

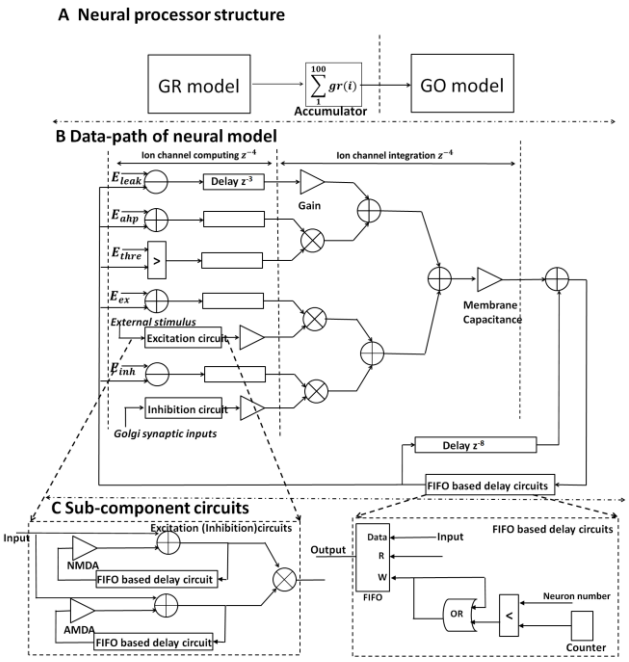

Fig. 3: The neural processor structure and the data path of neural model. (A) shows the conceptual structure of the processor and (B) shows the data path of the neural model. (C) shows the sub-component circuits: excitation (inhibition) circuits and FIFO based delay circuits. The triangle blocks denote the NMDA and AMPA receptor conductance. The mathematic model is described in equation(1-3) in [5]**.**

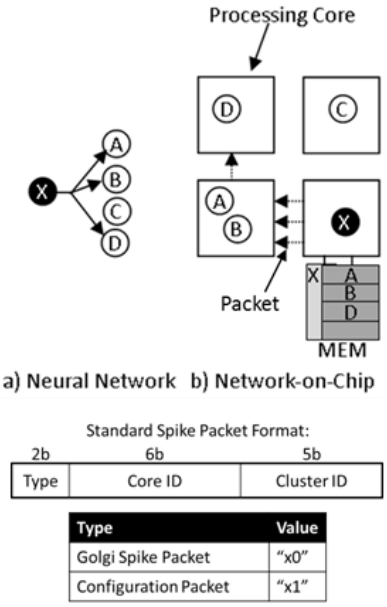

Fig. 4: Example of mapping of neural network to a network-on-chip; a) A sample Golgi neural network; b) Each core may model multiple Golgi cells. When the Golgi cell X, produces an action potential, individual packets are transmitted to teach connected granule-cell clusters which are distributed through the network.

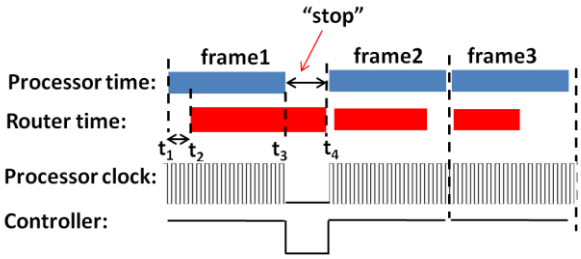

Fig. 5: The frame master performances. In the frame 1, the router processing time is longer than the processor's, so the frame master temporally disabled the neural processor at *t3-t4* periods until the router finished its current traffic loads. While in the frame 2 and 3, because of the routing time is shorter than the processor time, the processor clock was continuously running.

processing elements start to process the next time step. This ensures that the granule-cell clusters receive all their updates within the correct time period. For example, as is shown in Fig. 5, the duration of the network communication depends on the load of the network, which is determined by the frequency of Golgi cells spiking and the Golgi cell topologies. This varies for each frame. In each frame, once the first Golgi cell spike event is released (at time t2), the router starts to process the corresponding synaptic packages. After all 20 Golgi cell spike events are computed (at time t3), the processor's duty in frame 1 is finished. Then the neural processor needs to start computing the next 20 Golgi cell activities for frame 2. However, in frame 1 after time t3, the network is still processing the current 20 Golgi cell communication tasks. Therefore there is extra time allocated for the network to finish the first frame, before frame 2 begins. As results of this, the frame master generates a low level signal that disables the processor clock for the t3 – t4 period until the network has completed the current frame routing task. The frame master then enables the processor to allow it to start computing again.

III. Results

Fig. 6a shows the spike patterns of 40 granule cells randomly chosen from different granule-cell clusters. These granule cells show different temporal activity patterns. In contrast, the Golgi cells fire spikes rather regularly. Fig. 6b shows the similarity index of the activity pattern against the time shift Δt (Eq. (9) in [5]), which measures the temporal evolution of GR cell firing pattern. The similarity index monotonically decreases with $|\Delta t|$, indicating that the populations of active granule cells change gradually over time such that no active granule-cell clusters appear more than once throughout the stimulation, which reveals a one-to-one correspondence of GR population and time interval representation. The hardware simulation result is well comparable with software simulation with mean error being less than 5% (Fig. 6b). The error is mainly caused by hardware truncation errors. Fig. 6c shows the reproducibility index (Eq.(10) in [5]) from the hardware simulation, which compares the activity pattern generated by two different

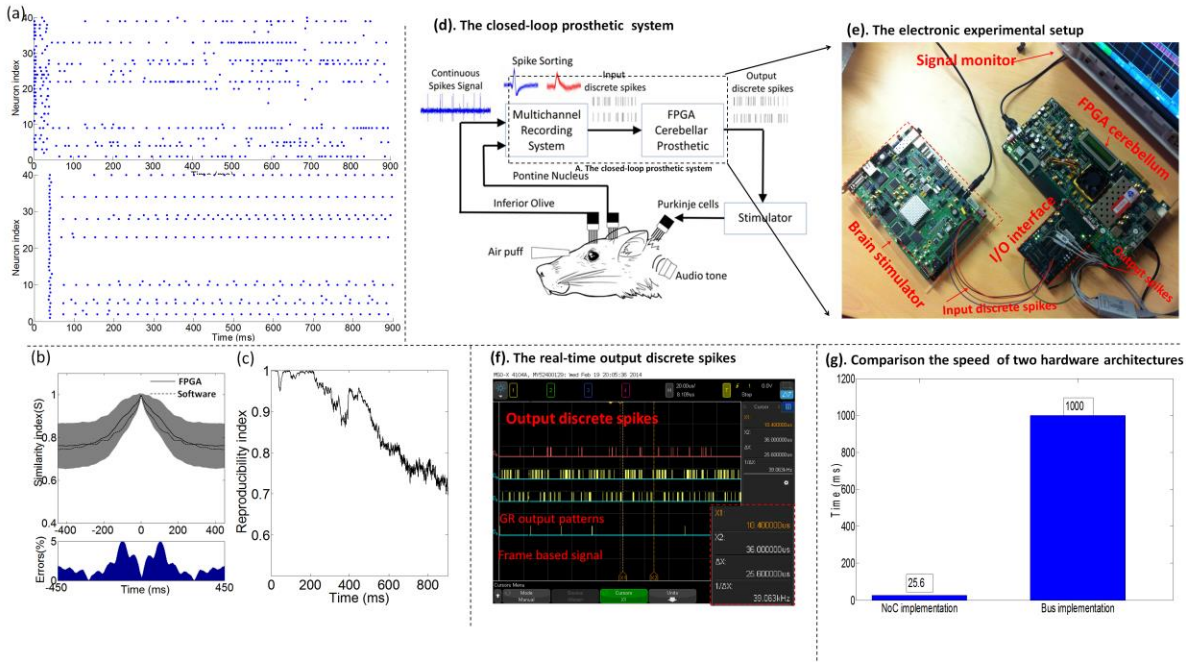

Fig. 6: (a): Spike patterns of 40 granule cells and Golgi cells chosen randomly in an implemented granular layer. (b): Comparison of similarity index between software and FPGA simulations. The grey areas are the standard deviations of the hardware results. The errors between the two results are shown at the bottom. (c): The reproducibility index is calculated by equation (5). (d): A hypothetical *in vivo* closed-loop experimental setup for cerebellum rehabilitation. (e): An electronic setup to demonstrate the feasibility of the *in vivo* experiment. (f): The real-time output discrete spikes and the frame-based signal. (g): The comparison of the speed of NoC and bus hardware architectures.

Poisson spike inputs. The reproducibility index remains high (>0.7), indicating that the POT encoding will remain robust despite of variability of signals in the two stimulating inputs. This shows that the neuron population can maintain consistent POT representation across trials when, for instance, learning of delayed eyeblink conditioning over multiple training sessions is to be incorporated in the model [5].

We illustrated a hypothetical *in vivo* experimental setup for closed-loop prosthetic application using our FPGA granular layer system in Fig. 6d. Fig. 6e shows an electronic system setup to demonstrate such an experiment. A Virtex-5 board is employed to simulate the biological spikes conveyed by MFs, the input discrete spikes are modeled as two 5Hz and two 30Hz Poisson spike trains in 4-bits signals. The proposed silicon granular layer is implemented on the Virtex-7 board with the I/O interface for displaying the system output on the oscilloscope in real-time (Fig. 6f). The displayed GR spikes were taken from three neural processors. The frame-based signal is also shown which is used to monitor and verify system processing behaviours. In Fig. 6g, our system can complete 1s of simulation in as little as 25.6ms, which is much faster than many core based bus implementation system (around 1s).

IV. Conclusion

The goal of the work has been to implement a real-time cerebellar granular layer model onto a FPGA hardware platform utilizing a NoC hardware architecture. The major contributions are: 1) An efficient FPGA-based NoC hardware architecture is proposed for implementing a large-scale cerebellar granular-Golgi layer model for POT encoding. 2) Our design can be a potential neuro-prosthetics tool for future experimental and clinical applications owing to its high computational power and high scalability.

Acknowledgements

Junwen Luo was partially supported by Research Grants Council of Hong Kong SAR (Project No. 138613) and Croucher Foundation of Hong Kong (Project No. 9500014).

References

[1] R. B. Ivry and R. M. C. Spencer, "The neural representation of time," *Curr. Opin. Neurobiol.*, vol. 14, no. 2, pp. 225–232, Apr. 2004.

[2] J. D. Schmahmann, "Disorders of the cerebellum: ataxia, dysmetria of thought, and the cerebellar cognitive affective syndrome.," *J. Neuropsychiatry Clin. Neurosci.*, vol. 16, no. 3, pp. 367–78, Jan. 2004.

[3] S. A. Bamford, R. Hogri, A. H. Taub, I. Herreros, F. M. J. Verschure, M. Mintz, and P. Del Giudice, "A VLSI field-programmable mixed- signal array to perform neural signal processing and neural modelling in a prosthetic system," pp. 1–14.

[4] M. Miwa, T. Hashiyama, T. Furuhashi, and S. Okuma, "Cerebellar model arithmetic computer with bacterial evolutionary algorithm and its hardware acceleration using FPGA," in *IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on*

[5] T. Yamazaki and S. Tanaka, "A spiking network model for passage-of-time representation in the cerebellum.," *Eur. J. Neurosci.*, vol. 26, no. 8, pp. 2279–2292, Oct. 2007.