Energy-efficient Spike-based Scalable Architecture for Next-generation Cognitive AI Computing Systems*

Ogbodo Mark Ikechukwu[1], Khanh N. Dang[2,1], and Abderazek Ben Abdallah[1]

[1] Adaptive Systems Laboratory, Graduate School of Computer Science and Engineering, The University of Aizu, Aizu-Wakamatsu, Fukushima 965-8580, Japan
{d8211104,benab}@u-aizu.ac.jp

[2] SISLAB, University of Engineering and Technology, Vietnam National University Hanoi, Hanoi, 123106, Vietnam
khanh.n.dang@vnu.edu.vn

Abstract. In recent years, neuromorphic computing systems have taken a range of design approaches to exploit known computational principles of cognitive neuro-biological systems. Profiting from the brain's event-driven nature modeled in spiking neural networks (SNN), these systems have been able to reduce power consumption. However, as neuromorphic systems require high integration to ensemble a functional silicon brain-like, moving to 3D integrated circuits (3D-ICs) with the three-dimensional network on chip (3D-NoC) interconnect is a suitable approach that allows for scalable designs, lower communication cost, and lower power consumption. This paper presents the design and evaluation of an energy-efficient spike-based scalable neuromorphic architecture. Evaluation results on MNIST classification, using the K-means-based multicast routing algorithm (KMCR), show that the proposed system maintains high accuracy with a small spike arrival window over various configurations.

Keywords: Spiking Neural Network · Scalable Architecture · Energy-efficient · Next-generation AI

1 Introduction

Spiking neural network (SNN) has gradually gained awareness by reason of its ability to process and communicate sparse binary signals in a highly parallel manner similar to that of the biological brain. Spike-based neuromorphic systems have leveraged this to exhibit rapid event-driven information processing and low power consumption. SNNs are modeled after the biological information processing of the brain, where information is communicated via spikes, and the processing of these spikes depends on their timing and the identity of the

* This work is supported by the University of Aizu, Competitive Research Funding (CRF), Ref. UoA-P6-2020

synapses used in communicating them. In contrast to multi-layer perceptrons where all neurons communicate by firing at every propagation cycle, SNN communication takes place only when the membrane potential of neurons are stimulated beyond a threshold [19]. Various ways can be employed when encoding information as spikes in SNN, and some of them include rate coding, population coding, and temporal coding [25]. There are various models of spiking neurons, and they describe the dynamics of a biological neuron at different levels. Some models which are broadly used include: the integrate and fire (IF) model [9], leaky integrate and fire (LIF) [9], and Hodgkin Huxley (HH) [14]. In general, their operation can be summarized as integrating currents from arriving spikes and the generation of new spikes whenever a threshold is exceeded. Typical spikes irrespective of their amplitude and shape are handled as similar events, and from the outset to finish, they last about two milliseconds [1] traveling down axonal lengths. The IF and LIF neuron models can easily be found in neuromorphic systems because of their simplicity and ease of implementation. However, the HH neuron model is not usually employed because its complexity makes it less suitable for large-scale simulation and hardware implementation.

SNN has successfully been used for tasks that range from vision systems [15] to brain-computer interfacing [27]. Performing software sim-ulation of SNN has shown to be a flexible approach to exploring the dynamics of neuronal systems. However, as SNNs become deeper, simulating it in software becomes slow and consume more power, making it less suitable for large scale and real-time SNN simulation. As an alternative approach, Hardware implemen-tation (neuromorphic system) provides the potential for rapid parallel real-time simulation of SNN, and holds an edge of computational acceleration over software simulations. Moreover, multi-neurocore neuromorphic systems can leverage the structure, stochasticity, parallelism, and spike sparsity of SNN to deliver rapid fine-grained parallel processing with low energy cost.

Over the years, Neuromorphic processors such as Loihi [7], MorphIC [11] and TrueNorth with two-dimensional (2D) architecture have been proposed. Loihi is a manycore 14-nm processor with on-chip learning capability. It occupies a silicon area of 60-mm^2 and communicates off-chip in a hierarchical mesh manner using an interface. MorphIC [11] is a quad-core processor with 512 LIF neurons per core and 528k synapses. It conducts learning using its on-chip stochastic spike-driven synaptic plasticity learning module. TrueNorth is the largest of these processors, with one million neurons and 256 million 1-bit synapses.

For three-dimensional (3D) architectures, the works in [31] and [30] both proposed multi-core 3D architectures, achieved by stacking several Stratix III FPGA boards. Inter-neuron communication was implemented using tree-based topology. This architecture, however, is not suitable as ASIC implementation, and because of the drawbacks of its topology, it seldom gets deployed in embedded neuromorphic systems [8].

The complexity of neural networks have increased over the years to inculcate multiple layers, each of which are expressed in 2D. These layers, when considered together, form a 3D structure. Mapping such structure on a 2D circuit generally results in several lengthy wires occurring between layers, or the occurrence of congestion points in the circuit [4, 5, 28]. 3D packet-switched NoC (3D-NoC), however, enables such structure to be mapped efficiently with communication between layers enabled via short through-silicon vias (TSVs). 3D-NoC also allows SNN to be scaled and parallelized in the third dimension by combining NoC and 3D ICs (3D-ICs) [2].

In designing a neuromorphic architecture that will support such deep SNN with many synapses, some challenges require attention. First, there is a need for a densely parallel multicore architecture with low-power consumption, light-weight spiking neuro processing cores (SNPCs) with on-chip learning, and efficient neuron coding scheme. Another major challenge that requires attention is on-chip neuron communication. Furthermore, we need to keep in mind that the number of neurons to be interconnected is immensely larger than the number of cores that require interconnection on recent multicore systems on chip platforms [12]. These challenges make the design of such a neuromorphic integrated circuit (IC) a demanding task [3].

This paper presents the design and evaluation of an energy-efficient spike-based scalable neuromorphic architecture. An extended version of this paper with fault-tolerant support and real hardware design is presented in [22]. The rest of this paper is organized as follows: Section II describes the architecture of the system's main building blocks. In section III, we present the evaluation, and in section IV, we present the conclusion and future work.

2 System Architecture

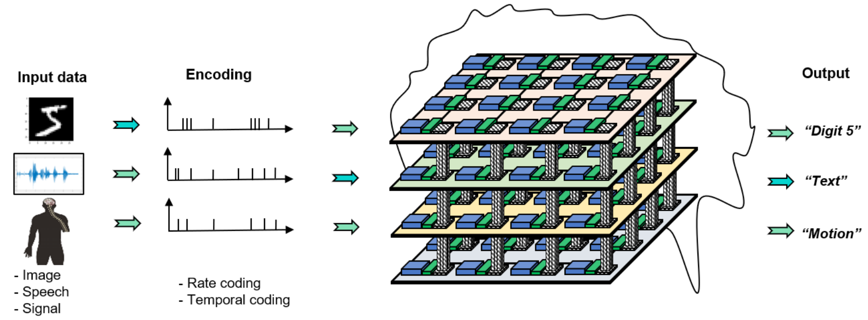


Fig. 1: High Level View of the System Architecture: (a) System architecture illustrated in a $4 \times 4 \times 4$ configuration.

A high-level view of our proposed 3D-NoC-based neuromorphic architecture [29] in a $4 \times 4 \times 4$ configuration is presented in Figure 1. This architecture integrates several nodes in a 3D-NoC interconnect to provide a scalable neuromorphic architecture. Each node consists of a spiking neuron processing core (SNPC) [23], Network Interface (NI), and a multicast 3D router (MC-3DR). The nodes are arranged in two-dimensional (2D) mesh layers stacked to form a 3D architecture. Communication between layers is enabled via through-silicon-vias (TSVs).

2.1 Spiking Neuron Processing Core (SNPC)

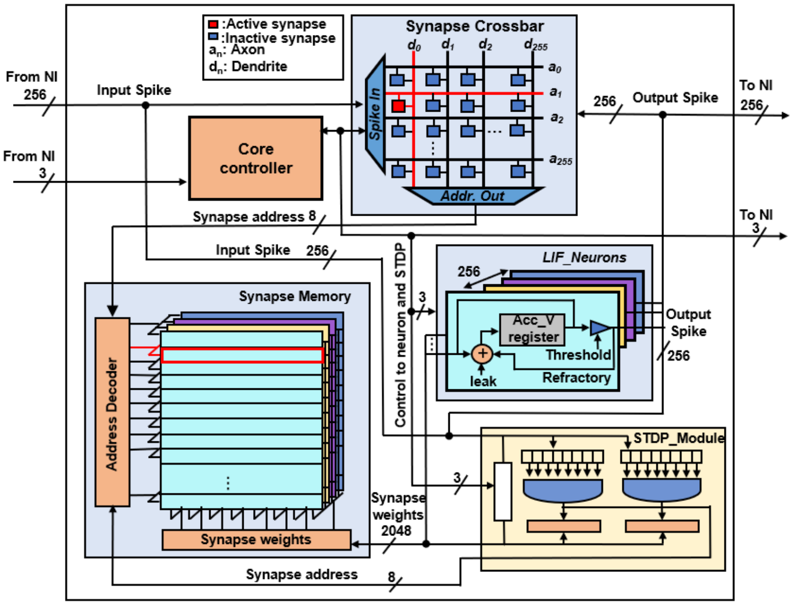


Fig. 2: Architecture of the SNPC Comprising of the Synapse Memory, Synapse Crossbar, LIF Neurons, Control Unit, and STDP learning module

The architecture of the SNPC is described in Figure 2. It is composed of a core controller, synapse Crossbar, Synapse memory, 256 LIF neurons, and STDP learning module. The SNPC multiplexes the state of all 256 neurons onto a single bus of 256 bits, each bit signifying the presence or absence of a spike event with 1 or 0, respectively. A total of 65k synapses are represented at the synapse crossbar. Spike processing operation on the SNPC is carried out in response to the core controller's control signals. The SNPC assumes an initial default state of idle. At the arrival of a presynaptic spike array which is preceded by a spike arrival signal, it downloads the presynaptic spike array to the synapse crossbar.

At the synapse crossbar, the presynaptic spike array is checked for the presence of spike events. If present, the crossbar determines the postsynaptic neuron and the synapse memory address of the associated synapses. This is done by performing one hot operation on the presynaptic spike array. When the synapse memory addresses are determined, the synapse weights stored at those addresses are fetched from the synapse memory and sent to the postsynaptic neurons for update. At the LIF neuron described in Figure 3, the synaptic weights received from the synapse memory are accumulated as the membrane potential. At the end of the accumulation, a leak operation that causes slight decay in the membrane potential value occurs. After that, the value of the membrane potential is compared with the neuron threshold value. If it exceeds the threshold, an output spike is fired. If not, no spike is fired. In the event of an output spike, the membrane potential value is reset to zero, and the neuron enters a refractory period that lasts a few time steps. While in the refractory period, the neuron cannot accumulate synaptic weights but resumes accumulation once the refractory period is over. The output spike array from the postsynaptic neuron is sent to the NI to be encoded into packets. The SNPC design enables the 256 neurons to be updated in one cycle. The accumulation of synapse weights by the LIF neuron is described in equation 1 as:

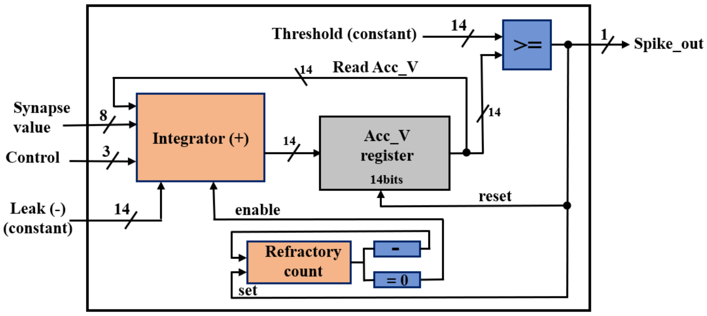


Fig. 3: Illustration of the Leaky Integrate and Fire Neuron Design.

$$V_j^l(t) = V_j^l(t-1) + \sum_i w_{ij}{}^{*} x_i^{l-1}(t-1) - \lambda \tag{1}$$

where V_j^l is the membrane potential of a LIF neuron j in layer l at a time-step t. w_{ij} is the synaptic weight from neuron i to j, λ is the leak and x_i^{l-1} is pre-synaptic spike from previous layer $l-1$.

On-chip learning in each of the 65k synapses of the SNPC is achieved with an efficient implementation of the trace-based spike-timing-dependent plasticity (STDP) Learning rule [24]. As described in Figure 4, the STDP module requires 16 presynaptic spike trace arrays, and the presence of postsynaptic spike arrays to carry out a learning operation. After an output spike array from the

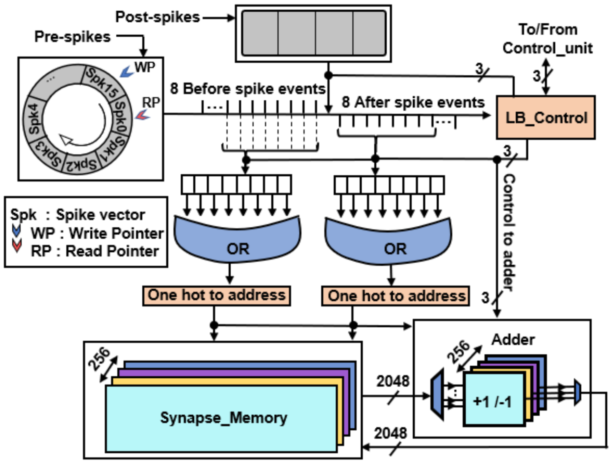


Fig. 4: Architecture of the STDP Learning Module.

LIF neurons has been sent to the MC-3DR, the SNPC checks if learning conditions have been met. If met, learning begins, if not, learning is skipped, and the SNPC returns to the idle state. The presence of 16 presynaptic spike traces, and postsynaptic spike trace arrays are verified to begin learning. If present, the presynaptic spike trace arrays are grouped into two, 8 $Before$ and 8 $After$, based on their arrival time relative to the postsynaptic spike trace array(s). An OR operation is further performed on the groups to obtain two arrays. Using one hot operation and the postsynaptic spike trace array, the associated synapses' memory addresses are obtained from the two arrays. The corresponding synapse values are then fetched from the synapse memory, increased for the $Before$ spike events, decreased for the $After$ spike events, and then written back to the synapse memory. The trace-based STDP design utilizes 256 adders, which enables parallel update of synapses.

2.2 Neurons Interconnect

The MC-3DR is responsible for spike communication among SNPCs in the 3D architecture. As described in Figure 5, it has 7 inputs and 7 output ports. Four of those ports connect to neighboring routers in the east, west, north, and south direction, two connect to the neighboring routers in the layers above and below, and the last connect to the local SNPC. The MC-3DR routes packets using four pipeline stages: buffer writing (BW), routing calculation (RC), switch arbitration (SA), and crossbar traversal (CT) [6]. It begins the first pipeline stage BW by storing the packet in the input buffer when it receives a packet from the NI or other routers. When BW is complete, the second pipeline stage RC begins. The packet's source address is obtained from the packet itself, and the next destination is calculated to arbitrate the right output port. After the right output

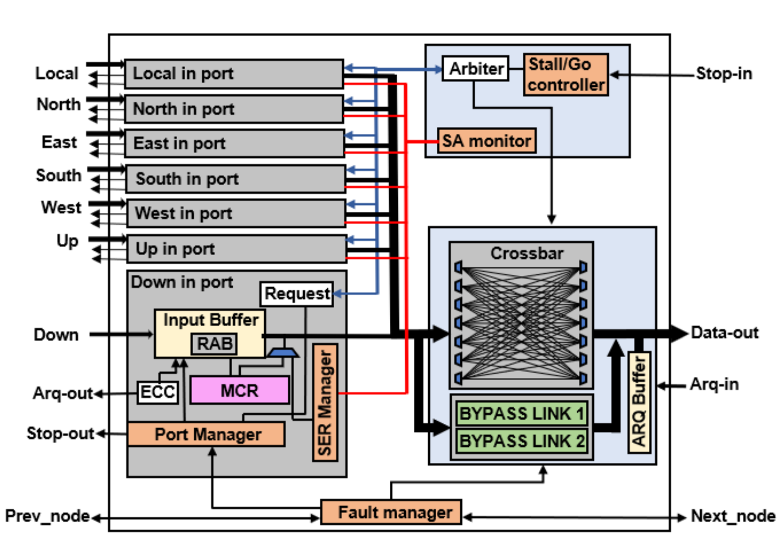
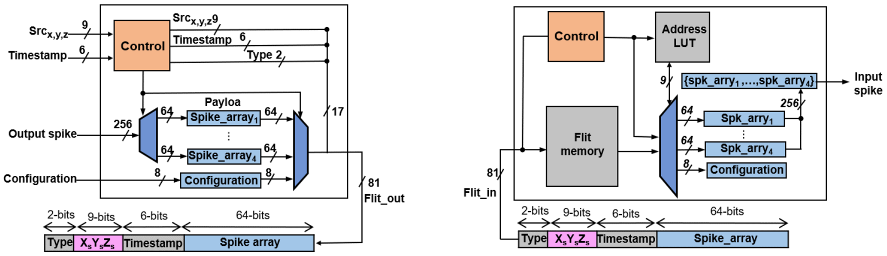


Fig. 5: Multicast 3D Router (MC-3DR) [29] Architecture, Illustrating its Ports and Internal Modules

port has been determined, the third pipeline stage begins. In this stage, the switch arbiter grants permission for the output port to be used. In the final stage, the packet is sent to the right output port through the crossbar.

The architecture of the NI is described in Figure 6. It is made up of two modules: Encoder and Decoder. The architecture of the encoder is described in Figure 6a, and its task is to pack spike arrays received from the SNPC into packets and send them to the router for transmission. The encoder packs spikes into flits of packet using an 81-bit flit format. The first two bits indicate the *"Type"* of the flit: "00" for configuration and "11" for the spike. The next 9-bits (3-bits each for X, Y, and Z dimensions) are used to represent the address of the source neuron. The following 6-bits are a record of the time in which the source neuron fired the spike. The last 64-bits are used for the spike array from presynaptic neurons. In contrast to the encoder, the decoder, which is described in Figure 6b unpacks packets that are received from the router into spike arrays before sending them to the SNPC.

To ensure efficient operation, we adopt and explore the shortest path k-means based multicast routing algorithm (KMCR) presented in [29]. In routing packets, the KMCR first partition destination nodes into subgroups, and from these subgroups, nodes with the least mean distance to other nodes in the subgroup are chosen to act as centroids. When the centroids have been chosen, the packets are routed from source node to the centroids, and then from the centroids to the destination nodes using a spanning subtree.



(a)

(b)

Fig. 6: Diagram of the Network Interface modules: (a) Encoder: packs presynaptic spikes into flits before routing to the destination SNPC, (b) Decoder: decodes flits received from source SNPCs into presynaptic spikes.

3 Evaluation Results

The proposed system was designed in Verilog-HDL, and synthesized with Cadence Genus. The NANGATE 45nm open-cell library [20] was used as the standard cell, the system memory was generated using OpenRAM [13], and TSV from FreePDK3D45 [21] was employed for inter-layer connection. To explore the efficiency of our proposed system, we evaluate it by carrying out MNIST data set [18] classification with on-chip and off-chip learning using SNN size of 784:400:10 and 784:100, respectively. The MNIST benchmark which contains 60K training and 10K testing images was used for evaluating the system because it is widely used, and therefore provides a basis for comparison with existing works.

3.1 Performance Evaluation

We evaluate the system performance by classifying MNIST dataset. The classification was done on our proposed system using 3 different network configuration sizes of 3×3×3 with a layer-based mapping scheme described in Fig. 7. The input layer of 784 neurons is mapped to the first layer of the system. The hidden layer of 400 neurons is also mapped onto the second layer. Finally, the output layer of 10 neurons is mapped to the third layer. The evaluation focused on classification accuracy and average classification time (ACT) on different configurations using the KMCR and the XYZ-UB algorithms, over various spike arrival windows (SAWs). The ACT is the average time taken to classify one MNIST image, and the SAW is the number of cycles allowed for all flits (spikes) from source SNPCs to arrive destination SNPC. After the first flit arrives, the SAW starts counting down till zero, and flits that do not arrive by the end of the countdown are not decoded. When the SAW countdown is over, the flits that arrived before the end of its countdown are decoded and sent to the destination SNPC, and it's value is reset.

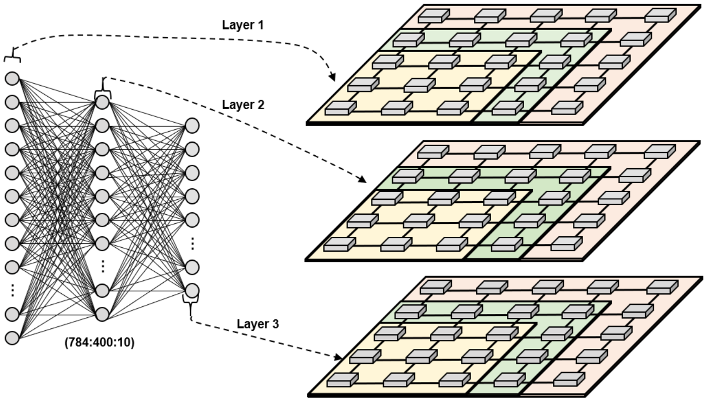


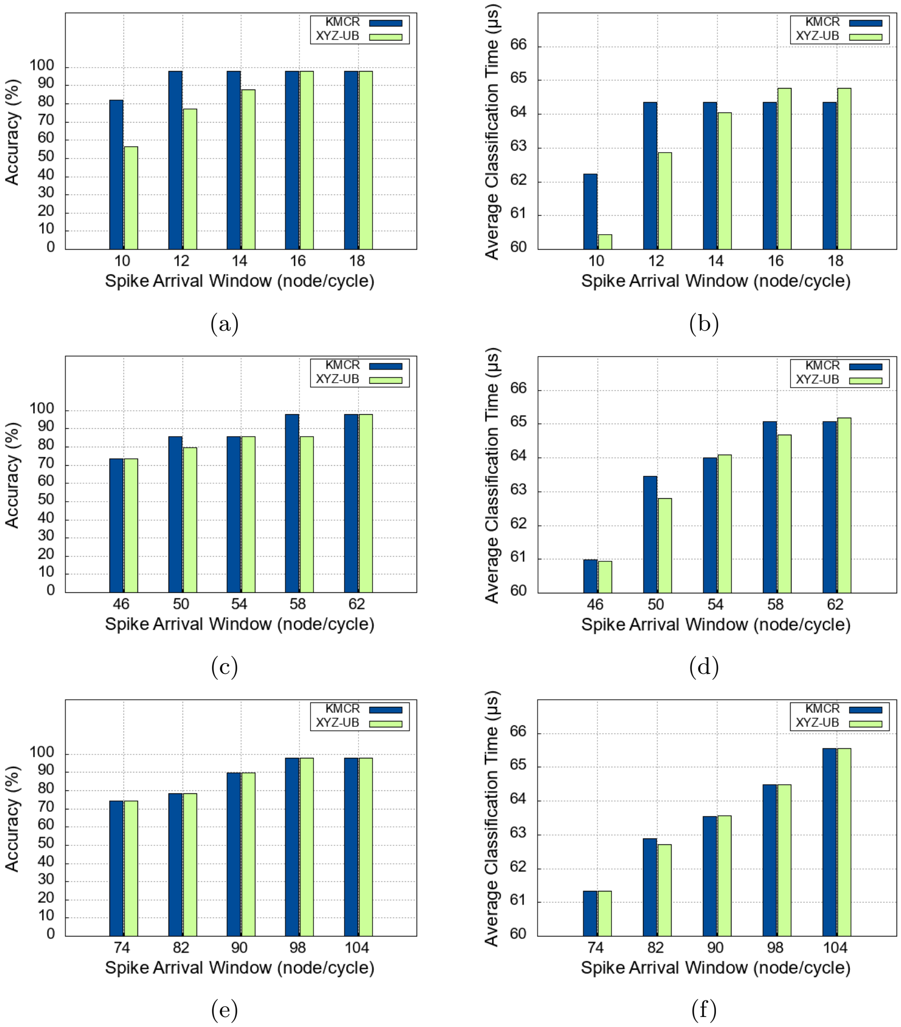
Fig. 7: SNN mapping for MNIST classification on $3\times3\times3$, $4\times4\times3$, and $5\times5\times3$ configurations: The first layer of 784 neurons without neural computation is mapped to Layer one, the second layer of 400 neurons is mapped to layer 2, and the third layer of 10 neurons is mapped to layer 3).

The accuracy and ACT of the evaluated system configurations over various SAWs are described in Fig. 8. For the $3\times3\times3$ configuration described in Fig. 8a and Fig. 8b, the KMCR shows better accuracy from SAW of 10 to 14 with 24.8%, 5.2%, and 9.9% better accuracy. However, as the SAW reaches 16, both algorithms reach same accuracy 98.2%. This is because the KMCR algorithm is able to service more spikes at lower SAW compared to the XYZ-UB, which reflects in the ACT, where the KMCR is lower than the XYZ-UB from SAW 10 to 12, due to the increased time taken by the KMCR to process more spikes that arrived within the SAW.

For the second configuration of $4\times4\times3$ described in Fig. 8c and Fig. 8d, the KMCR and XYZ-UB show similar accuracy at SAW 46, 54 and 62. This is because the similar number of spike packets were able to reach the destination SNPC for both algorithms. However, a slight difference can be seen in SAW 50 and 58, where the KMCR was able to utilize the little timing difference between it and the XYZ-UB, to deliver more spikes, which resulted in better accuracy compared to the XYZ-UB. However with more spike delivered at SAW 50 and 58, KMCR utilized more classification time compared to XYZ-UB. At SAW of 62, when the accuracy of 98.2% was reached, the KMCR utilizes 2.3% less ACT compared to XYZ-UB.

In the third configuration of $5\times5\times3$ described in Fig. 8e and Fig. 8f, both the KMCR and the XYZ-UB show similar performance in Accuracy and ACT over all the SAWs. It was observed that the performance of both algorithms gradually becomes similar as the size of system configuration increases.

With the SNPC able to update all neurons in parallel, the time taken to update neurons given same SNN size and number of spikes across the utilized



(a) (b)

(c) (d)

(e) (f)

Fig. 8: MNIST classification (with off-chip learning) on different system configurations over various SAW: using the KMCR and XYZ-UB algorithms. (a) Accuracy on a $3 \times 3 \times 3$ system configuration. (b) Average classification time on on a $3 \times 3 \times 3$ system configuration. (c) Accuracy on on a $4 \times 4 \times 3$ system configuration. (d) Average classification time on on a $4 \times 4 \times 3$ system configuration. (e) Accuracy on on a $5 \times 5 \times 3$ system configuration. (d) Average classification time on on a $5 \times 5 \times 3$ system configuration.

system configurations are the same. The difference in the ACT and accuracy among the configurations result from the time taken to route spike packets, and the number of packets delivered to the destination SNPCs. As can be observed in Fig. 8, the $4 \times 4 \times 3$ and $5 \times 5 \times 3$ configurations requires 3.5 and 5.8 times more SAW respectively, compared to the $3 \times 3 \times 3$.

In conclusion, we evaluated different configurations of the proposed system system by classifying MNIST hand written digits. We show that with off-line training, the system can successfully deliver reasonable accuracy across different configurations with increased SAW and ACT.

A comparison of the architecture and evaluation result of our proposed system and some existing works [10, 17, 26] is also performed. Compared to other works, our system utilizes a scalable 3D architecture which helps to provide increased parallelism and reduced communication cost. Also, the SNPC design utilizes parallel neuron and synapse update approach rather than the serial approach employed in other works. This enables all neurons to be updated in one cycle and synapses in two.

At an on-chip STDP learning accuracy of 79.4% on the MNIST dataset classification, our system achieved a higher accuracy than *Seo et al.* [26]. The higher accuracy is a consequence of the higher synapse precision utilized by our system. *ODIN* [11] and *Kim et al* [16] nonetheless achieved a higher accuracy, but employed some form of supervision and image pre-processing, to achieve it.

Table 1: Comparison between the proposed system and existing works.

Parameters/Systems	**Kim et al. [16]**	**ODIN [10]**	**Seo et al [26]**	**This work**
Accuracy (%)	84	84.5	77.2	79.4
Neurons / core	64	256	256	256
Neuron Model	IF	LIF and Izh.	LIF	LIF
Neuron Update	serial	serial	serial	parallel
Synapses /core	21k	64K	64k	65k
Synapse Precision	4, 5, 14	4-bit	1-bit	8-bits
On-chip Learning Rule	Stoch. grad. desc.	Stoch. SDSP	STDP	STDP
Memory Technology	SRAM	SRAM	SRAM	SRAM
Interconnect	2D	2D	2D	3D

4 Conclusion

In this work, we presented architecture and evaluation of an energy-efficient spike-based scalable neuromorphic architecture for next-generation cognitive AI computing systems. The proposed approach immensely benefits from the scalability, high parallelism, low communication cost, and high throughput advantages that 3D-NoC provides. Leveraging these benefits, our system can support large-scale SNN with an enormous amount of synapses. Evaluating with MNIST dataset classification, our method achieved 98.2% and 79.4% accuracy with off-chip and on-chip learning, respectively. The evaluation results show that with different configurations, our system can maintain high accuracy. Although the energy merit cannot be clearly seen in this evaluation due to the type and size of the used benchmark, we expect a higher energy efficiency and accuracy when the proposed system is benchmarked with large biological application.

References

1. Bear, M.F., Connors, B.W., Paradiso, M.A.: Neuroscience: Exploring the Brain, 4th Edition, pp. 81–108. Lippincott Williams and Wilkins, Lippincott Williams and Wilkins, 351 W Camden St, Baltimore, MD 21201, United States (2016)
2. Ben Abdallah, A.: Advanced Multicore Systems-On-Chip: Architecture, On-Chip Network, Design, chap. 6, pp. 175–199. Springer Singapore, Singapore (Sep 2017)
3. Carrillo, S., Harkin, J., McDaid, L.J., Morgan, F., Pande, S., Cawley, S., McGinley, B.: Scalable hierarchical network-on-chip architecture for spiking neural network hardware implementations. IEEE Transactions on Parallel and Distributed Systems **24**(12), 2451–2461 (Dec 2013). https://doi.org/10.1109/tpds.2012.289
4. Dang, K.N., Ahmed, A.B., Okuyama, Y., Abdallah, A.B.: Scalable design methodology and online algorithm for TSV-cluster defects recovery in highly reliable 3D-NoC systems. IEEE Transactions on Emerging Topics in Computing **8**(3), 577–590 (Oct 2017). https://doi.org/10.1109/TETC.2017.2762407
5. Dang, K.N., Ahmed, A.B., Okuyama, Y., Abdallah, A.B.: Scalable design methodology and online algorithm for tsv-cluster defects recovery in highly reliable 3D-NoC systems. IEEE Transactions on Emerging Topics in Computing **8**(3), 577–590 (2020)
6. Dang, K.N., Ahmed, A.B., Tran, X.T., Okuyama, Y., Abdallah, A.B.: A comprehensive reliability assessment of fault-resilient network-on-chip using analytical model. IEEE Transactions on Very Large Scale Integration (VLSI) Systems **25**(11), 3099–3112 (Nov 2017). https://doi.org/10.1109/tvlsi.2017.2736004
7. Davies, M., Srinivasa, N., Lin, T.H., Chinya, G., Cao, Y., Choday, S.H., Dimou, G., Joshi, P., Imam, N., Jain, S., Liao, Y., Lin, C.K., Lines, A., Liu, R., Mathaikutty, D., McCoy, S., Paul, A., Tse, J., Venkataramanan, G., Weng, Y.H., Wild, A., Yang, Y., Wang, H.: Loihi: A neurophic manycore processor with on-chip learning. IEEE Micro **38**(1), 82–99 (January 2018). https://doi.org/10.1109/MM.2018.112130359
8. Ehsan, M.A., Zhou, Z., Yi, Y.: Modeling and analysis of neuronal membrane electrical activities in 3D neuromorphic computing system. In: 2017 IEEE International Symposium on Electromagnetic Compatibility Signal/Power Integrity (EMCSI). pp. 745–750 (Aug 2017). https://doi.org/10.1109/ISEMC.2017.8077966
9. Fourcaud-Trocmé, N.: Encyclopedia of Computational Neuroscience: Integrate and Fire Models, Deterministic, pp. 1–9. Springer New York, New York, NY (2013)
10. Frenkel, C., Lefebvre, M., Legat, J.D., Bol, D.: A 0.086-mm^2 12.7-pj/SOP 64k-synapse 256-neuron online-learning digital spiking neuromorphic processor in 28-nm CMOS. IEEE Transactions on Biomedical Circuits and Systems **13**(1), 145–158 (Feb 2019). https://doi.org/10.1109/TBCAS.2018.2880425
11. Frenkel, C., Legat, J.D., Bol, D.: Morphic: A 65-nm 738k-synapse/mm 2 quad-core binary-weight digital neuromorphic processor with stochastic spike-driven online

learning. IEEE Transactions on Biomedical Circuits and Systems **13**, 999–1010 (Oct 2019). https://doi.org/10.1109/TBCAS.2019.2928793

12. Furber, S., Temple, S.: Neural systems engineering. Journal of The Royal Society Interface **4**(13), 193–206 (nov 2006). https://doi.org/10.1098/rsif.2006.0177
13. Guthaus, M.R., Stine, J.E., Ataei, S., Brian Chen, Bin Wu, Sarwar, M.: Openram: An open-source memory compiler. In: 2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD). vol. 34, pp. 1–6 (2017). https://doi.org/10.1145/2966986.2980098
14. Hodgkin, A.L., Huxley, A.F.: A quantitative description of membrane current and its application to conduction and excitation in nerve. Bulletin of Mathematical Biology **52**(1), 25–71 (Jan 1990). https://doi.org/10.1007/BF02459568
15. Hopkins, M., García, G., Bogdan, P., Furber, S.: Spiking neural networks for computer vision. Interface Focus **8**(4), 128–136 (Jun 2018). https://doi.org/10.1098/rsfs.2018.0007
16. Kim, J.K., Knag, P., Chen, T., Zhang, Z.: A 640m pixel/s 3.65mw sparse event-driven neuromorphic object recognition processor with on-chip learning. In: 2015 Symposium on VLSI Circuits (VLSI Circuits). pp. C50–C51 (2015). https://doi.org/10.1109/VLSIC.2015.7231323
17. Kim, Y., Zhang, Y., Li, P.: A reconfigurable digital neuromorphic processor with memristive synaptic crossbar for cognitive computing. ACM Journal on Emerging Technologies in Computing Systems **11**(4), 1–25 (Apr 2015). https://doi.org/10.1145/2700234
18. LeCun, Y., Cortes, C., Burges, C.: MNIST handwritten digit database. http://yann.lecun.com/exdb/mnist/, (accessed 23.02.2021)
19. Maass, W.: Networks of spiking neurons: The third generation of neural network models. Neural Networks **10**(9), 1659–1671 (Dec 1997). https://doi.org/10.1016/s0893-6080(97)00011-7
20. NanGate Inc.: Nangate Open Cell Library 45 nm. http://www.nangate.com/, (accessed 05.05.2021)
21. NCSU Electronic Design Automation: FreePDK3D45 3D-IC process design kit. http://www.eda.ncsu.edu/wiki/FreePDK3D45:Contents, (accessed 05.05.2021)
22. Ogbodo, M., Dang, K., Abdallah, A.: On the design of a fault-tolerant scalable three dimensional noc-based digital neuromorphic system with on-chip learning. IEEE Access **9**(1), 64331–64345 (2021). https://doi.org/10.1109/ACCESS.2021.3071089
23. Ogbodo, M., Vu, T., Dang, K., Abdallah, A.: Light-weight spiking neuron processing core for large-scale 3D-NoC based spiking neural network processing systems. In: 2020 IEEE International Conference on Big Data and Smart Computing (BigComp). pp. 133–139 (2020). https://doi.org/10.1109/BigComp48618.2020.00-86
24. Rahimi Azghadi, M., Iannella, N., Al-Sarawi, S.F., Indiveri, G., Abbott, D.: Spike-based synaptic plasticity in silicon: Design, implementation, application, and challenges. Proceedings of the IEEE **102**(5), 717–737 (May 2014). https://doi.org/10.1109/JPROC.2014.2314454
25. Rodrigues de Oliveira Neto, J., Cerquinho Cajueiro, J.P., Ranhel, J.: Neural encoding and spike generation for spiking neural networks implemented in FPGA. In: 2015 International Conference on Electronics, Communications and Computers (CONIELECOMP). pp. 55–61 (2015)
26. Seo, J., Brezzo, B., Liu, Y., Parker, B.D., Esser, S.K., Montoye, R.K., Rajendran, B., Tierno, J.A., Chang, L., Modha, D.S., Friedman, D.J.: A 45nm cmos neuromorphic chip with a scalable architecture for learning in networks of spiking neurons.

In: 2011 IEEE Custom Integrated Circuits Conference (CICC). pp. 1–4 (Sep 2011). https://doi.org/10.1109/CICC.2011.6055293

27. Valencia, D., Thies, J., Alimohammad, A.: Frameworks for efficient brain-computer interfacing. IEEE Transactions on Biomedical Circuits and Systems **13**(6), 1714–1722 (Dec 2019). https://doi.org/10.1109/TBCAS.2019.2947130
28. Vu, T.H., Ikechukwu, O.M., Ben Abdallah, A.: Fault-tolerant spike routing algorithm and architecture for three dimensional NoC-based neuromorphic systems. IEEE Access **7**, 90436–90452 (2019)
29. Vu, T.H., Okuyama, Y., Abdallah, A.B.: Comprehensive analytic performance assessment and k-means based multicast routing algorithm and architecture for 3D-NoC of spiking neurons. ACM Journal on Emerging Technologies in Computing Systems **15**(4), 1–28 (Dec 2019). https://doi.org/10.1145/3340963
30. Yang, S., Deng, B., Wang, J., Li, H., Lu, M., Che, Y., Wei, X., Loparo, K.A.: Scalable digital neuromorphic architecture for large-scale biophysically meaningful neural network with multi-compartment neurons. IEEE Transactions on Neural Networks and Learning Systems **31**(1), 148–162 (2020). https://doi.org/10.1109/TNNLS.2019.2899936
31. Yang, S., Wang, J., Deng, B., Liu, C., Li, H., Fietkiewicz, C., Loparo, K.A.: Real-time neuromorphic system for large-scale conductance-based spiking neural networks. IEEE Transactions on Cybernetics **49**(7), 2490–2503 (2019). https://doi.org/10.1109/TCYB.2018.2823730