# A 2048-Neuron Spiking Neural Network Accelerator with Neuro-Inspired Pruning and Asynchronous Network on Chip in 40nm CMOS

Sung-Gun Cho [1], Edith Beigné [2], and Zhengya Zhang [1]
[1] University of Michigan, Ann Arbor, MI, USA
[2] Université Grenoble Alpes, CEA-LETI MINATEC, Grenoble, France

***Abstract*—A 40nm, 2.56mm$^2$, 2048-neuron globally asynchronous locally synchronous (GALS) spiking neural network (SNN) chip is presented. For scalability, we allow neurons to specialize to excitatory or inhibitory, and apply distance-based pruning to cut communication and memory. An asynchronous router limits the latency to 1.32ns per hop. The reduced traffic and lower latency allow the input channel to be parallelized to achieve 7.85GSOP/s at 0.7V, consuming 5.9pJ/SOP.**

***Index Terms*—spiking neural network, asynchronous network-on-chip, distance-based pruning, deadlock handling.**

## I. Introduction

Bio-inspired spiking neural networks (SNN) have been demonstrated to perform versatile cognitive tasks. The size of a SNN, i.e., number of neurons, determines the capability of the SNN. To enable the efficient mapping of large-scale SNNs, hardware accelerators have been designed using modular tiles and network-on-chip (NoC) [1]–[3]. However, the synapse count scales quadratically with the number of neurons. Scaling up a SNN eventually saturates NoC and on-chip memory bandwidths, leading to a higher latency and energy. A high latency, in particular, affects the dynamics of a SNN and degrades its performance [4].

In this work, we allow neurons to specialize to either excitatory or inhibitory and apply distance-based pruning to cut the communication traffic and memory size. An asynchronous router design limits the average latency to 1.32ns per hop, 3.1× lower than the state-of-the-art design [1]. The results are demonstrated in a 2.56mm$^2$, 2048-neuron, 16-tile globally asynchronous locally synchronous (GALS) SNN chip in 40nm CMOS. The reduced traffic and lower latency allow the input channel to be parallelized by 4× to achieve 7.85GSOP/s at 0.7V, consuming 5.9pJ/SOP, where a SOP denotes a synaptic operation that conveys a spike from a neuron to another through a nonzero unique synapse [1], [2].

## II. Neuron Specialization and Distance-Based Pruning

Biological neurons are not homogeneous. Rather, they are specialized to excitatory neurons (E neurons) that are stimulated by inputs, and inhibitory neurons (I neurons) that are stimulated by E and I neurons [5]. Illustrated in Fig. 1, E neurons, which are the majority, do not suppress E neurons directly; whereas I neurons, the minority, suppress both E and I neurons. An $N$-neuron fully-connected SNN requires nearly $N^2$ synapses and $N^2$ weights. By specializing $M$ ($M \ll N$) neurons to I neurons, and $N - M$ neurons to E neurons, the number of synapses and weights are reduced by approximately $N/M$. Known algorithms, such as E-I Net [5], have shown competitive classification performance using networks of specialized neurons.

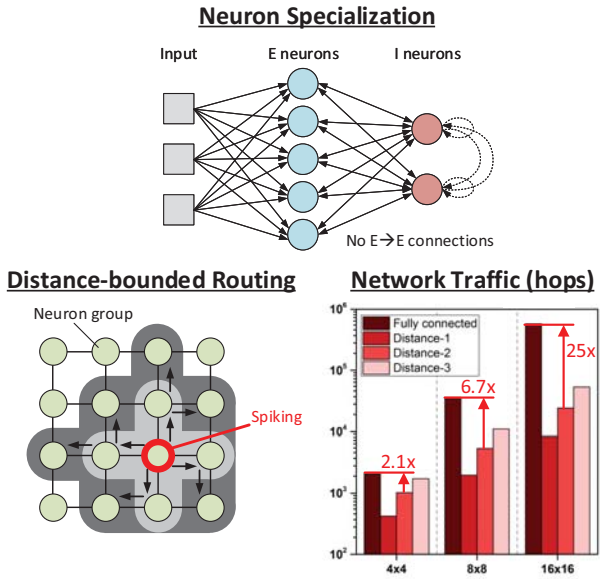


Fig. 1. Neuron specialization to E and I neurons; and distance-bounded routing and the resulting traffic reduction.

Connections between biological neurons are distance-dependent with dense connectivity between nearby neurons and sparse connectivity between distant ones [6]. This property motivates a design to constrain the connectivity between neurons based on distance. As shown in Fig. 1, neurons within a local group are fully connected and the connections between distant groups are pruned. The distance-based pruning reduces spike traffic by 52% to 95%, with a higher reduction for a larger network. The pruned SNNs can be trained to perform the same tasks with high efficacy based on our experiments.

* 978-1-5386-9395-7/19/$31.00 ©2019 IEEE

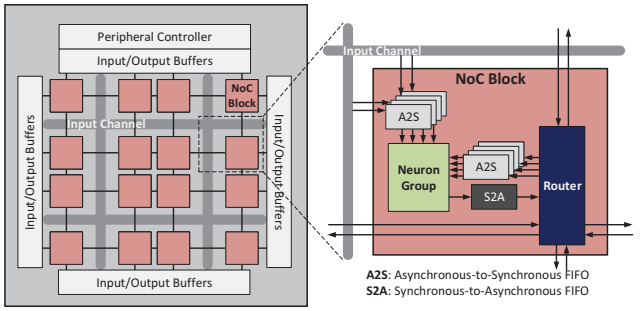


Fig. 2. GALS SNN architecture consisting of $4 \times 4$ neuron groups, asynchronous NoC and input channels.

## III. SNN Accelerator Design

A 2048-neuron SNN prototype is constructed as shown in Fig. 2 by tiling 16 neuron groups in a $4 \times 4$ grid. As one example, we choose 1/4 of the neurons to be configured as I neurons and 3/4 to be E neurons, cutting the number of connections and weights by $4\times$; and we apply a distance bound of 2 hops to further reduce the spike traffic and the number of weights by 24%. These choices are made to keep the accuracy degradation of the SNN below 1% for a common set of classification tasks.

Following the GALS architecture, each neuron group is placed in a separate clock domain, and the groups are connected to an asynchronous NoC to propagate sparse and stochastic spike traffic. Asynchronous-to-synchronous (A2S) and synchronous-to-asynchronous (S2A) FIFOs bridge synchronous clock domains with the asynchronous NoC. To reduce latency and the backpressure from the NoC, we allocate dedicate asynchronous input channels for the efficient broadcast of dense input data.

By neuron specialization, distance-based pruning and dedicate input channels, the processing bandwidth can be quadrupled with the support of 4 input channels, leading to a $4\times$ higher utilization of the neurons.

### A. Neuron Group

Separated from A2S and S2A FIFOs, and an asynchronous NoC router within a tile, a neuron group is comprised of 128 integrate-and-fire neurons configurable as either E neurons or I neurons, as shown in Fig. 3. A neuron group receives inputs and spikes that stimulate and inhibit the neurons, respectively.

The inputs received from the input channels are multiplied by input weights to compute stimuli. The input weights are quantized to 3b and cached in an 18kb input weight memory. Spikes are received from neurons within the group and external to the group from the router. The inhibitory weights are looked up based on the spike address. The 2b nonuniform-quantized inhibitory weights are cached in a 9kb inter-group weight memory and a 0.375kb intra-group weight memory and the 2b weights are restored to 3b after being read out.

The 128 neurons, clustered in 4 arrays, accumulate stimuli and subtract inhibitions to update potentials. Spikes are generated when a threshold is crossed. An arbiter and a FIFO serialize the spikes for output.

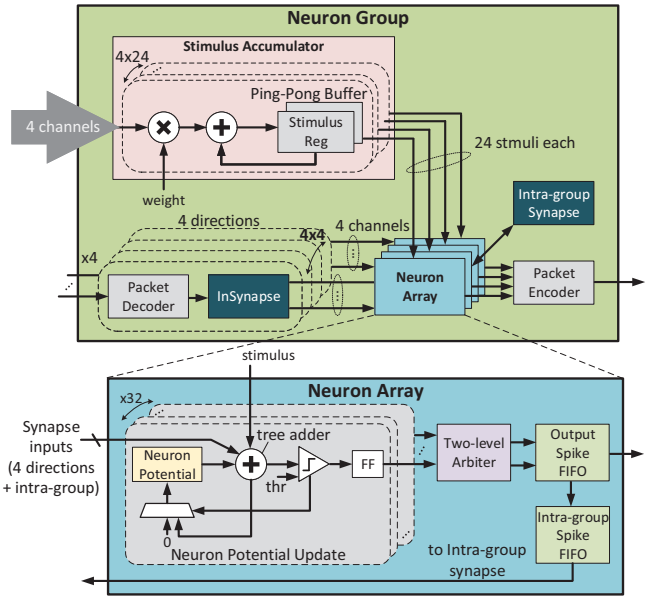


Fig. 3. Neuron group design.

### B. Low-Latency Asynchronous Network-on-Chip

All the neuron groups and input/output buffers in the peripheral are connected through asynchronous NoC routers. The router is designed in dual-rail logic [7], [8], which performs a handshake transaction to transfer output of one gate to the next gate in order to assure valid data flow without a synchronizing clock signal.

An asynchronous NoC router multicasts spike packets from a neuron group to other neuron groups as well as to an output buffer designated for collecting the spike packets from the group for downstream processing, e.g., classification. The router provisions IN and OUT ports to each of the four directions and the neuron group, as shown in Fig. 4.

In a conventional design, to guarantee the correct operation, the input port switch needs to hold a transaction to all the OUT ports until all the OUT ports respond. This mechanism can cause one bottleneck to be propagated to all directions. To alleviate the impact, we employ a branch buffer to enable the independent completion transaction on each branch, freeing an OUT port to handle a new request from other input ports when the OUT port's current transaction is done.

The OUT port performs arbitration before forwarding the data. In a straightforward implementation, the arbiter accepts a request from an input port switch only after all the data bits from the input switch arrive. Then, the forwarding logic needs to wait for the selection signal from the arbiter. The strictly sequential operation results in an increased latency. To shorten the latency, we design the arbiter to look ahead the switch enable signals from IN ports to trigger arbitration using a tree

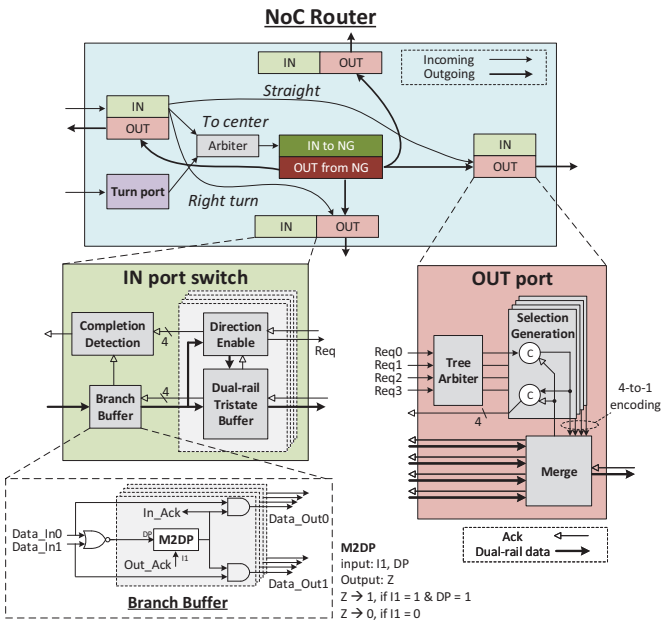


Fig. 4. Asynchronous NoC multicast router design.

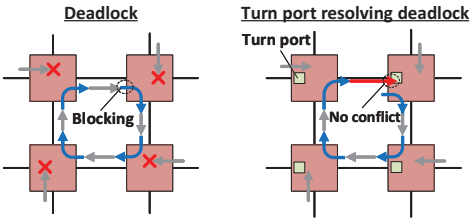


Fig. 5. Deadlock resolution by the addition of a turn port in each router.

arbiter prior to receiving all the bits in the packet, allowing the selection to be done well in advance.

The branch separation and look-ahead arbitration techniques reduce the average per-hop latency to 1.32ns in 40nm CMOS, 3.1× lower than the state-of-the-art [1]. The router is designed by synthesizing dual-rail logic elements [8] as standard cells. To reduce traffic for a more scalable design, spike packets are distance-bounded to 2 hops, beyond which they are converted to output packets or removed from the network. To cut duplicate traffic, routing is limited to straight pass or right turn.

Since the NoC allows all possible right turns, a loop of turns made by unfinished packet transactions can cause a deadlock. In the previous work [1], [2], dimension order routing was adopted to prevent deadlocks by routing with priority on either X- or Y-dimension. Dimension order routing may imbalance the traffic in different ports, resulting in traffic congestion. Instead, we add a turn port in every router to channel the packets that have made a specific turn (*e.g.* packets heading north and then making a turn to east in Fig. 5). The addition of a turn port in each router prevents chain of channel

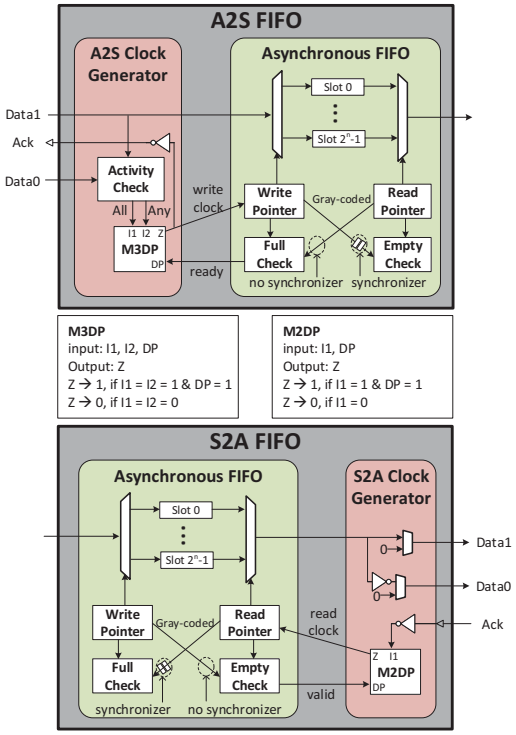


Fig. 6. Asynchronous-to-synchronous (A2S) FIFO and synchronous-to-asynchronous (S2A) FIFO design.

blockage without causing traffic congestion in any direction, as illustrated in Fig. 5. Since our routing rule allows only right turns, each router requires only one turn port and a 2:1 arbiter to direct packets from the turn port.

### *C. Asynchronous-Synchronous Domain Crossings*

The clock domain crossing is handled by A2S and S2A FIFOs. The FIFO designs follow an asynchronous FIFO as shown in Fig. 6. To translate an asynchronous signal into a synchronous signal, A2S FIFO and S2A FIFO require write and read clock generator logic, respectively. Write clock generator produces a positive clock edge for write when all the dual-rail bits arrive, and a negative clock edge when all the dual-rail bits retract. Read clock generator produces a positive clock edge for read when the dual-rail handshake is received, and a negative clock edge when the transaction is acknowledged.

Activity checker in A2S FIFO provides "all" and "any" signals, which notify whether all or any, respectively, of the data bits are active for completion detection. The M3DP and M2DP gates in Fig. 6 are variants of Muller C-element that efficiently provide compound logic functions. Also note that the synchronizer is eliminated on the asynchronous side, as a transaction can be invoked at any time when ready in the asynchronous domain. The dual-rail A2S and S2A FIFOs are designed using standard cells and synthesized with timing constraints.

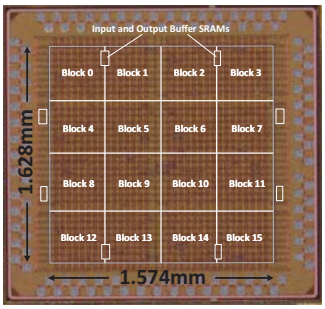


Fig. 7. Chip microphotograph.

| Compression ratio | Avg. E spike rate | NRMSE |
|---|---|---|
| 2.38 | 0.039 | 0.0712 |
| 3.21 | 0.029 | 0.0775 |
| 4.77 | 0.019 | 0.0781 |
| 5.73 | 0.016 | 0.078 |
| 6.84 | 0.014 | 0.0785 |
| 17.48 | 0.005 | 0.0756 |


Fig. 8. Tradeoff between compression ratio and reconstruction error in image compression task. A reconstructed image with the highest compression ratio is compared with the original image.

## IV. Chip Measurement Results

The 2048-neuron, 16-tile GALS SNN accelerator test chip was fabricated in a 40nm CMOS process. The accelerator core occupies 2.56mm$^2$ as shown in Fig. 7. The chip is measured to achieve 7.85GSOP/s at a 0.7V supply voltage in room temperature with neuron groups running at 110MHz, consuming 46.4mW at 5.9pJ/SOP.

We demonstrate an example application of this SNN chip in compressed sensing using learned dictionary elements. In Fig. 8, we show the results of compressing natural images [9]. The experiment was performed by dividing input images into 16×16 patches, and applying the SNN chip to encode the patches using learned dictionary elements. The compression ratio can reach as high as 17.5× with a relatively low normalized root-mean-square error (NRMSE) of 7.6%. The dictionary can be learned from unsupervised training, enabling effective compression of any type of data. To demonstrate the SNN's feature extraction capability, we trained and mapped a one-layer recurrent SNN together with a linear classifier for MNIST, demonstrating a classification accuracy of 91.6%.

Table I compares this work with the latest SNN chip designs. The energy efficiency of this 40nm chip is more competitive than the 14nm Loihi [2] without technology normalization. The energy and area efficiency of this chip are within the ranges reported for the latest 10nm SNN accelerator [3]. Compared to all-digital [4] and mixed-signal SNN ASIC implementations [10], this design supports a larger network, a higher compute density, a higher energy efficiency, and programmablity. The neuron specialization, distance-based pruning and low-latency GALS approaches demonstrated by this work will pave the way for more efficient and scalable SNN accelerator designs.

TABLE I
Measurement results and comparison table

| | [2] | [3] | [4] | [10] | This Work |
|---|---|---|---|---|---|
| **Process** | 14nm | 10nm | 65nm | 40nm | 40nm |
| **Area (mm$^2$)** | 60 | 1.72 | 3.1 | 1.31 | 2.56 |
| **# of Neurons** | 13.1K | 4096 | 256 | 512 | 2048 |
| **# of Synapses** | 126M | 1M | 128K | - | 149K |
| **Synapse Bits** | 1b | 7b | 8b / 13b | - | 3b / 2b |
| **Voltage (V)** | 0.75 | 0.525 | 1.0 | 0.9 | 0.7 |
| **Architecture** | Async | Sync | Sync | Sync | GALS |
| **Freq (MHz)** | - | 105 | 310 | 250 | 110 |
| **Input Bandwidth (Mpixels/s)** | - | - | 1240 | 1778 | 440 |
| **SOP/s** | 440G | 5.2G | 1.24G | 2.0G | 7.85G |
| **SOP/s/mm$^2$** | 7.34G | 3.0G | 400M | 1.53G | 3.07G |
| **pJ/SOP** | 23.6 | 3.8 | 175.8 | 43.5 | 5.9 |
| **Deadlock Handling** | Dimension order routing | - | - | - | Turn port |
| **Latency per Hop** | 4.1 ns / 6.5 ns | - | - | - | 1.32 ns[1] |

[1]Average latency estimated in post-APR simulation

## Acknowledgment

This work was supported in part by DARPA HR0011-13-2-0015, the SONIC Center, and Intel Corporation. The authors would like to thank Pascal Vivet, Chester Liu, and Shuanghong Sun for advice.

## References

[1] P. A. Merolla *et al.*, "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, vol. 345, no. 6197, pp. 668–673, 2014. [Online]. Available: http://science.sciencemag.org/content/345/6197/668

[2] M. Davies *et al.*, "Loihi: A neuromorphic manycore processor with on-chip learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, January 2018.

[3] G. K. Chen *et al.*, "A 4096-neuron 1M-synapse 3.8pJ/SOP spiking neural network with on-chip STDP learning and sparse weights in 10nm FinFET CMOS," in *2018 Symposium on VLSI Circuits (VLSI Circuits)*, June 2018.

[4] P. Knag *et al.*, "A sparse coding neural network ASIC with on-chip learning for feature extraction and encoding," *IEEE Journal of Solid-State Circuits*, vol. 50, no. 4, pp. 1070–1079, April 2015.

[5] P. D. King *et al.*, "Inhibitory interneurons decorrelate excitatory cells to drive sparse code formation in a spiking model of V1," *Journal of Neuroscience*, vol. 33, no. 13, pp. 5475–5485, 2013.

[6] A. Ooyen *et al.*, "Independently outgrowing neurons and geometry-based synapse formation produce networks with realistic synaptic connectivity," *PloS one*, vol. 9, p. e85858, 01 2014.

[7] E. Beigne *et al.*, "Asynchronous circuit designs for the internet of everything: A methodology for ultralow-power circuits with gals architecture," *IEEE Solid-State Circuits Magazine*, vol. 8, no. 4, pp. 39–47, Fall 2016.

[8] P. Vivet *et al.*, "A $4 \times 4 \times 2$ homogeneous scalable 3d network-on-chip circuit with 326 MFlit/s 0.66 pJ/b robust and fault tolerant asynchronous 3D links," *IEEE Journal of Solid-State Circuits*, vol. 52, no. 1, pp. 33–49, Jan 2017.

[9] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, pp. 607–609, 1996.

[10] F. N. Buhler *et al.*, "A 3.43TOPS/W 48.9pJ/pixel 50.1nJ/classification 512 analog neuron sparse coding neural network with on-chip learning and classification in 40nm CMOS," in *2017 Symposium on VLSI Circuits*, June 2017, pp. C30–C31.