2025 Design, Automation & Test in Europe Conference (DATE 2025)

NeuroHexa: A 2D/3D-Scalable Model-Adaptive NoC Architecture for Neuromorphic Computing

Yi Zhong[1,*], Zilin Wang[1,*], Yipeng Gao[1], Xiaoxin Cui[1,3], Xing Zhang[1] and Yuan Wang[1,2,3]
[1]School of Integrated Circuits, Peking University, Beijing 100871, China
[2]Key Laboratory of Microelectronics Devices and Circuits (MoE), MPW Center, Peking University, Beijing 100871, China
[3]Beijing Advanced Innovation Center for Integrated Circuits, Beijing 100871, China
Email: wangyuan@pku.edu.cn

***Abstract*—Neuromorphic computing has endeavored a novel computing paradigm that entails a bio-inspired architecture to reproduce the remarkable functionalities of the human brain, such as massively parallel processing and extremely low-power consumption. However, those promising merits can be greatly canceled by the mismatched communication infrastructure in large-scale hardware implementation, in view of the vast degree of neural connectivity, the unstructured spike dataflow, and the unbalanced model workload assignment. In an effort to tackle those challenges, this work presents NeuroHexa, a network-on-chip (NoC) architecture intended for multi-core neuromorphic design. NeuroHexa adopts a customized intra-chip hexagonal topology, which can be further cascaded in 6 directions by either 2D or 3D chiplet integration. Designed in globally asynchronous, locally synchronous (GALS) methodology, a group of processing nodes can operate in independent work pace to further improve resource utilization. To satisfy the varied requirement of data reuse across the chip, NeuroHexa proposes a flexible multicast routing mechanism to best adapt to the model-defined dataflow. And under a specific congestion scenario, NeuroHexa can switch its routing algorithm between deterministic routing and fully adaptive routing modes. The presented NoC router is evaluated in 28nm CMOS, where we achieve the maximal throughput as 179.2Gbps, and the best energy efficiency as 4.872pJ/packet at the area overhead of 0.0226mm^2.**

Keywords—network-on-chip, neuromorphic computing, 2D/3D scalability, model adaptation, congestion awareness

I. Introduction

The neuromorphic approach of brain-like computing has paved a new way to fulfill artificial intelligence in a more comprehensive way. If compared with the power-intensive supercomputers, the remarkable brain can perform impressive feats with a limited power budget of 20W, in large part because of its vast connectivity, structural and functional organizational hierarchy, and spike time-dependent neuronal and synaptic functionality [1]. Inspired by those features, neuromorphic engineering attempts to mimic the aspects of the brain's architecture and dynamics in the following ways. Firstly, neuromorphic hardware generally entails spiking neurons and their synaptic connections as basic computing unit, and performs neuronal model, such as spiking neural network (SNN), to emulate the group behaviors [2]. Secondly, neuromorphic platform typically processes information in an asynchronous and event-driven manner, characterized by the co-localization of memory and computation [3]. Thirdly, neuromorphic system commonly adopts decentralized multi-core structure to realize massively parallel processing with large-scale spiking communication networks [4]. Taking into account all of those concerns, one of the most challenging parts of designing neuromorphic hardware is how to create a communication infrastructure that best fits the inherent working principles of neuromorphic computing.

Concentrating on the communication infrastructure, the neuromorphic computing systems differ with the conventional von Neuman computers in several significant aspects [5]. Neuromorphic computing utilizes a more biologically realistic approach so as to precisely mimic the nervous system, where a great challenge stems from integrating millions of neurons and billions of synaptic connections on a single chip. Unlike the continuous one-dimensional instruction sequences, the data interaction between neurons is usually performed by discrete spike sequences on a huge scope of spatio-temporal dimensions, which brings great difficulty for unstructured dataflow processing in a communication system. Moreover, the neuromorphic architecture mostly adopts near-memory or in-memory components to eliminate memory wall bottleneck, exploiting scalable but unbalanced parallelism in its operation. This could be a dauntingly hard task for overall workload assignment on a large-scale system.

Confronted with the challenges to build a neuromorphic computer, network-on-chip (NoC) architecture is generally acknowledged as a solution, as it shares many advantages with the human brain, such as resource reusability, good scalability, distributed parallelism, event-driven nature and robustness. In view of this, this work introduces NeuroHexa, a customized NoC design for multi-core/chip integration of neuromorphic system. The key contributions are listed as follows:

1) A novel hexagonal topology with improved throughput, latency and cost is proposed, in which each node can interchange data with its six neighbor nodes. And in accordance with the model requirement, a small chiplet can be further extended in either 2D or 3D directions.
2) By following the globally asynchronous, locally synchronous (GALS) methodology, processing nodes in the NoC can be controlled as group threads in separate synchronization domains, thus operating in their independent paces to improve resource utilization.
3) A flexible unicast and multicast routing mechanism is applied on the proposed NoC architecture, which is highly consistent with the data reuse demand in the specific neuromorphic model-defined dataflow.
4) Configurable routing mode is realized by switching the router between the deterministic XY-X-Y dimensional routing and the fully adaptive congestion-aware routing algorithms to fit in different congestion scenarios.
5) We evaluate the corresponding NoC circuit in the layout implementation of 28nm CMOS technology. We achieve the NoC node's maximal throughput as 179.2 Gbps and energy efficiency as 4.872pJ/packet, at the area overhead of 0.0226mm^2.

II. Background and Related Work

A. Neuromorphic AER Communication Principle

Neuromorphic system utilizes spike sequence to represent data information. As Fig. 1 shows, the computing units are made up of biomimetic neurons that will fire spikes once the

*Yi Zhong and Zilin Wang contributed equally to this work.

*This work was supported by the National Key Research and Development Program of China (Grant No. 2024YFB4405501).

978-3-9826741-0-0/DATE25/© 2025 EDAA

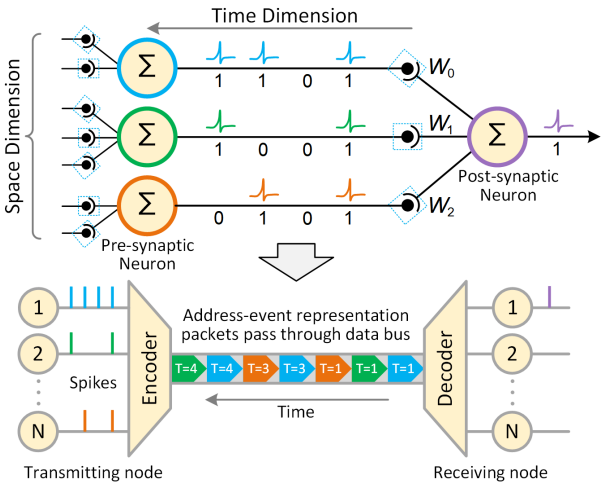


Fig. 1. AER based spiking communication principle.

accumulated input values exceed their thresholds. The fired spikes are generally referred as events, containing both spatial information like source and destination address, and temporal information like spike timing. The communication within neuromorphic system is inherently event-driven and possibly asynchronous, which puts a different strain on the design of hardware structure.

To cope with this, spikes are more commonly encoded into packets through Address-Event Representation (AER) format [7] in neuromorphic engineering. At the transmitting side, an AER packet may carry the spatio-temporal information, and then be multiplexed to the asynchronous digital bus based on its source address. While at the receiving side, the packet is decoded and reconstructed as its target location and sequence based on the destination address, and possible target timing in some designs. As for packet routing between the transmitting and receiving nodes, customized NoC communication is the most preferred solution in current neuromorphic designs.

B. NoC Architectures in Neuromorphic Hardware

In the past decade, plenty of neuromorphic chips have tried to explore the optimal NoC architecture to handle the global spiking communication across many cores. As shown in Fig. 2, with 2D mesh topology, routing nodes are connected in the four cardinal directions, as well as the local core. Previous works, like TrueNorth [8], Tianjic [9], and Darwin3 [10] chips adopt 2D mesh NoC in both multi-core and multi-chip scale. Another popular topology is tree routing structure, especially quadtree in 2D integration. Neuromorphic chips of DYNAPs [11], MorphIC [12], and PAICORE [13] adopt the hierarchical architecture, where a root node will repeatedly branch to connect to the four adjacent nodes in lower levels. They also possess 2D mesh scalability in multi-chip scale. Loihi [14] chip attempts to combine mesh and tree structure into a hybrid form, in which the lowest level is quadtree, while the higher levels remain as 2D mesh. SpiNNaker [15] builds a starlike NoC, where all the cores are directly connected to the central router, and then cascaded in six directions for multi-chip integration. In addition, some more specialized works, such as flatten butterfly [16] and C20 Fullerene [17] topologies, are also included in existing explorations.

Which is the best NoC architecture for neuromorphic computing is still not settled, as all the designs above have their pros and cons. For instance, the mesh structure offers high throughput, but suffers from poor latency as the chip scale increases, while the tree structure is just the opposite.

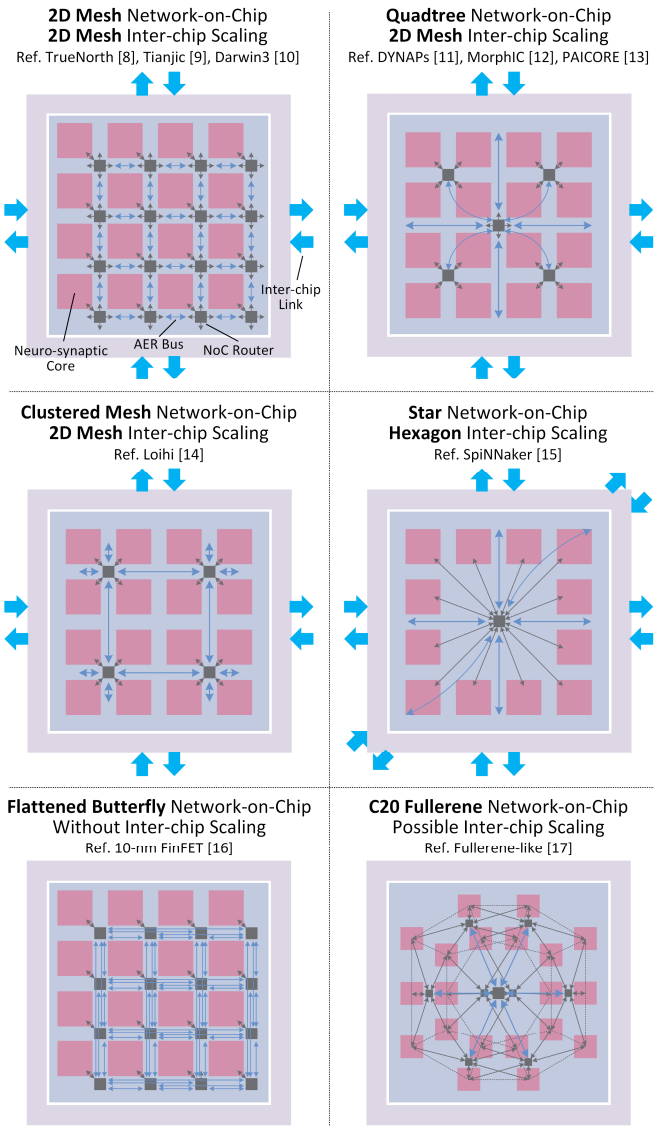


Fig. 2. AER based spiking communication principle.

Thus, a preferred NoC design should be co-optimized with its target computing models, especially in neuromorphic cases.

C. NoC Routing Algorithms in Neuromorphic Hardware

Routing algorithms in neuromorphic chips can be firstly categorized by their AER format as source and destination-based routing. The former labels spiking packet with source node address, and often adopts a costly look-up table to search its target nodes, such as [15] and [17]. Instead, the latter equips the packet with destination address, removes search memory but requests step-by-step parsing at intermediate nodes, like [8]~[10], [13]~[14] and [16]. Routers in [11] and [12] utilize a mixture of the two methods in their different routing phases.

Routing algorithms can also be defined by point-to-point routing and broadcast routing. The former simply routes the packet between two fixed nodes in some early works like [8] and [14], while the latter is commonly used in other recent works, where data reuse and reconfigurability count for a lot.

Moreover, routing algorithms can be classified according to their path planning, i.e., deterministic and adaptive routing. Most of the neuromorphic chips adopt the former to minimize the complexity of NoC, while recent research on congestion-aware design [18] has proved the necessity and effectiveness of introducing adaptive routing into neuromorphic hardware.

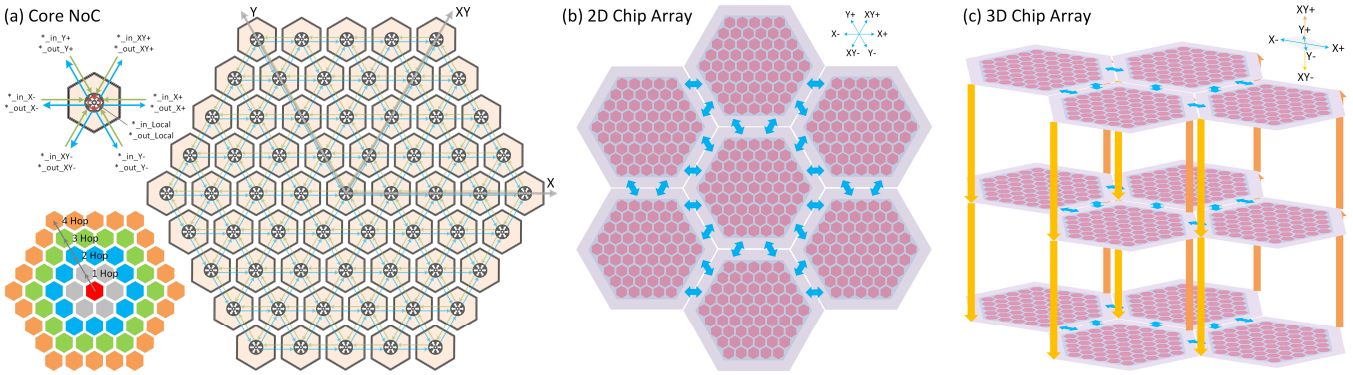


Fig. 3. Multi-core and multi-chip interconnections in NeuroHexa. (a) A hexagon-structured core NoC. (b) Chiplet cascaded to 6 directions within 2D plane. (c) Chiplet extended for 3D integration, with 4 in-plane directions and 2 cross-plane directions.

III. Network-on-Chip Architecture

The presented NoC, NeuroHexa, implements a scalable architecture at the level of multi-core and multi-chip scope. To feature the target neuromorphic hardware with well-matched communication infrastructure, we will focus on the design detail of the proposed NoC architecture hereinafter.

A. Scalable NoC Topology

The selection of an appropriate NoC topology depends heavily on its performance and cost of processing AER spiking packets. As listed in Table I, when applying to similar core number, quadtree shows minimal wire cost but exhibits the worst throughput, and octagon pursues the best throughput and average hops at the expense of too many wires, while 2D mesh performs moderately with the worst maximal hops.

In order to better balance the throughput, latency and cost, NeuroHexa proposes a novel hexagonal topology, which is structured like honeycomb as depicted in Fig. 3(a). A hexagon NoC of radius 5 consists of 61 routing nodes in its computing cores. Besides local side, each node is connected with its six neighbors labeled as XY±, X± and Y± directions. Compared with other 2D topologies, NeuroHexa offers a comprehensive advantage of ~25% improvement at hop latency than 2D mesh and quadtree, and ~25% less wire resources than octagon.

When extending to multi-chip level, NeuroHexa provides the choice of both 2D and 3D chiplet integration as shown in Fig. 3(b) and Fig. 3(c), respectively. It is notable that both hexagon and 3D mesh topologies require 6 adjacent nodes to be connected to single node, which makes the proposed design even suitable for 3D mesh scaling. In line with the 2D model dataflow, one can arrange the chiplet array within a single 2D plane, and utilize all the 6 directions to exchange data. When it comes to 3D dataset model, one can accordingly configure it as 3D chiplet array, in which the XY± directions can be leveraged as cross-plane routing, while X± and Y± directions keep the same as 2D mesh in-plane routing.

B. Routing Node Detail

As illustrated in Fig. 4, the routing node in each core is equipped with an input allocator and an output arbitrator, as well as several asynchronous FIFOs and output buffers in each of the seven directions (synchronous FIFOs in local direction). The input AER packets are synchronized and buffered in the input FIFOs in accordance with their source directions at first, which accomplishes the necessary clock domain crossing operation. And based on their tagged address information, the input allocator will then assign them into a specific output channel. As there may be multiple packets competing for one channel, the output arbitrator will select the prioritized one as winner in turn and send to output buffers. Once the adjacent or local FIFOs feedback an available signal, the output buffers can finally transmit the packets to their desired directions.

TABLE I. Quantitative Comparison of NoC Topologies

Topology	2D Mesh	Quadtree	Octagon	This work 3D Mesh	This work Hexagon
# of cores	64	64	64	64	61
# of wires	224	168	420	288	312
Max. hops	14	6	7	9	8
Avg. hops	5.25	5.34	3.69	3.75	4.05
Throughput	4×	1×	8×	6×	6×

The input allocator is depicted in the left part of Fig. 5. Whenever a FIFO non-empty signal is enabled, the allocator will read the FIFO and parse the data according to its tagged data type and configured routing algorithm. The AER packet can be defined as either single data frame or continuous data frame, where the latter can only carry the necessary spiking data to improve information density, and follow the same direction of previous packets. Next, by analyzing its target address, the packet can be performed as a unicast or other multicast routing. With the specific routing algorithm of either deterministic XY-X-Y dimensional routing or fully adaptive routing, the output channel for each packet can be determined.

The output arbitrator is depicted in the right part of Fig. 5. It adopts a Round-Robin-like priority queue mechanism, but always grants the local input the first to avoid the interruption of local core. Except for that, the routing request from XY±,

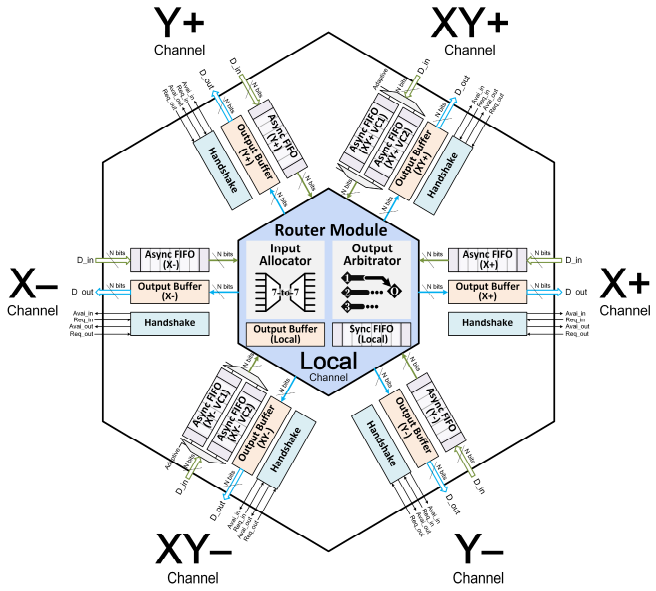


Fig. 4. Illustration of the structure details inside a routing node.

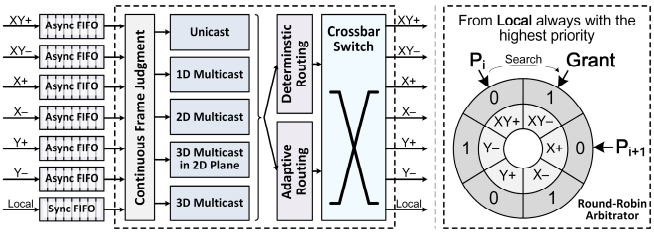


Fig. 5. Input allocator and output arbitrator design.

X± and Y± directions will undergo a compulsory prioritizing process, to mitigate the routing starvation. It is notable that the input allocator and the output arbitrator of all the directions can operate in parallel, which can adequately make use of the throughput in all the directions.

C. GALS Group Thread Control

NeuroHexa also follows a globally asynchronous locally synchronous (GALS) style, in which each node is located in a separate clock domain, hence possessing inherent asynchrony. Considering that there are common cases in which a group of cores will request for interaction with data dependency in neuromorphic computing, a concise and effective mechanism of local synchronization called group thread control is also incorporated in the NoC architecture.

As shown in Fig. 6(a), the group thread control mechanism assigns several control signals to monitor and instruct the computing cores within the NoC. These signals can be flexibly configured as whether connected or not with their neighbors. Within the scope of the connected wires, they can be relayed to and affect a certain scale of cores, so-called a group thread. In each group thread, the local synchronization is performed by the *sync_all* signal, which ensures the precise calculation order of data streams; the initial operation is performed by the *initial_all* signal; and the working status of the processing core and the NoC routing is reflected by the *done* and *busy* signals.

The illustration of multiple group threads is shown in Fig. 6(b), which may span the boundary of several chips. On this occasion, different threads are controlled by different pairs of global signals with independent time steps, hence enabling locally synchronous intra-group and globally asynchronous

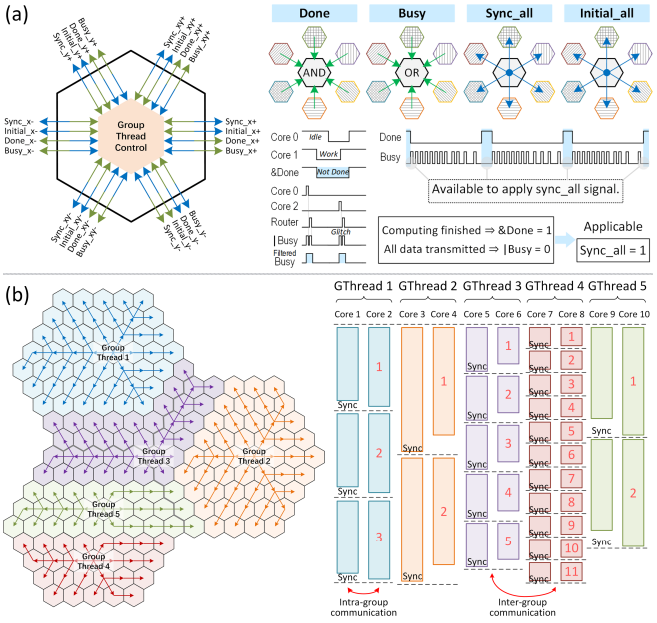


Fig. 6. GALS group thread control mechanism. (a) Group control signals. (b) Multiple group threads running on cascaded chiplets.

inter-group communication. Through the group thread control mechanism, NeuroHexa provides simultaneous multitasking solutions for neuromorphic chips, hopefully helping balance the overall workload on a large-scale system.

IV. Routing Algorithm Design

To better adapt to the characteristics of the unstructured dataflow in neuromorphic computing, NeuroHexa proposes a novel routing algorithm design. It realizes both unicast and multicast routing to meet the different requirements of data reuse, and configurable deterministic and adaptive routing to fit in the diverse congestion-aware scenarios.

A. Unicast and Multicast Routing

The neuromorphic datasets, commonly made up of spatio-temporal spiking event stream, may differ in their data forms. As Fig. 7 shows, neuromorphic cores may be dedicated to processing 1D audio data, 2D visual data, and even 3D point cloud data. However, the unstructured dimensional dataflows pose a critical challenge on the NoC routing. As a distributed multi-core parallel system, neuromorphic processor requires a fair degree of data reuse between any of the neural networks. For instance, the basic convolutional operations may request for sharing spiking AER packets among A, A×B and A×B×C cores in 1D, 2D and 3D dataflows, respectively. Therefore, it is crucial for a preferred NoC to support broadcast function.

In view of this issue, NeuroHexa proposes a novel unicast and multicast routing mechanism to flexibly adapt to 1D, 2D and 3D broadcast cases. As Fig. 8(a) lists, the spiking AER packets in NeuroHexa shall carry their destination addresses, which contains the relative distance $\triangle XY_n$, $\triangle X_n$ and $\triangle Y_n$, as well as a pair of special addresses $\#XY_n$, $\#X_n$ and $\#Y_n$ for packet

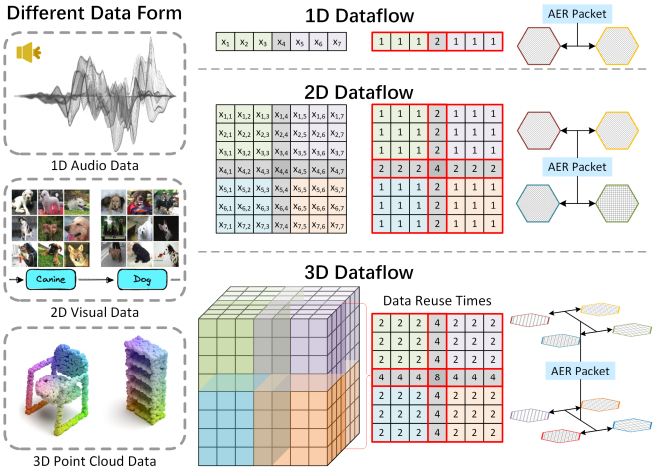


Fig. 7. Different dimensional dataflows in neuromorphic computing.

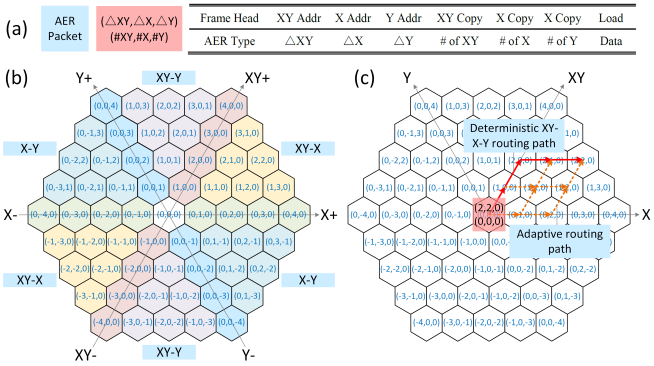


Fig. 8. Basic concept of AER routing in NeuroHexa. (a) AER packet format. (b) Relative distance coordinates. (c) Unicast routing example.

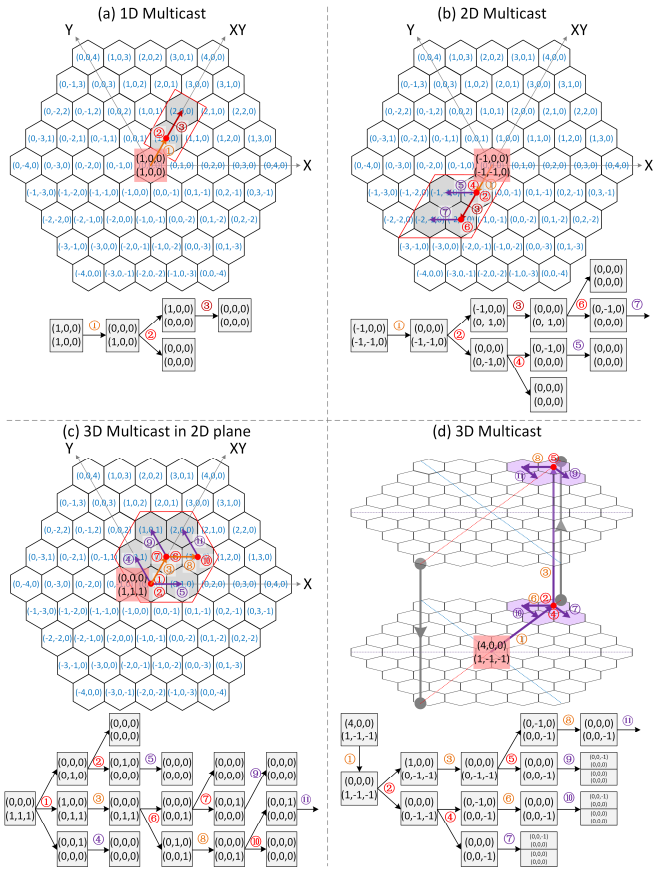


Fig. 9. Multicast routing implementation in NeuroHexa. (a) 1D multicast. (b) 2D multicast. (c) 3D multicast in 2D plane. (d) 3D multicast.

duplication in the 6 directions. Fig. 8(b) pictures the relative distance of all the nodes on a basis of the central node. Once an AER packet is created, the routing node will transmit it to the adjacent cores according to its tagged address information. Taking a unicast packet as example in Fig. 8(c), the address of (2, 2, 0) will require two XY+ hops and two X+ hops to finish routing, where its specific routing path can be decided by the deterministic or adaptive routing algorithms discussed below.

As for multicast scenarios, NeuroHexa leverages the copy addresses to represent how many packets should be duplicated in each direction. As illustrated in Fig. 9, when there are one, two or three of the copy addresses equal non-zero, we can refer them as 1D multicast, 2D multicast or 3D multicast cases, respectively. In each of the dimensions, the routing nodes will firstly conduct unicast routing until the relative address equals to zero. Then the packet will be duplicated as two copies each time, with one sent to the local core and the other sent to the desired multicast direction. By this means, through multiple duplications at the intermediate nodes, one can realize flexible multicast for any number of cores with the form like A, A×B and A×B×C, which corresponds to the proposed 1D, 2D and 3D multicast, respectively. Note that in Fig. 9(c), 3D multicast can even be conducted in 2D plane by splitting it as three 2D multicasts. And the analyzing sequence of the XY, X and Y dimensions can be alterable when applying to adaptive routing.

B. *Deterministic and Adaptive Routing*

NeuroHexa implements a configurable deterministic and adaptive routing algorithm for different congestion occasions, as it is not always the case that the latter performs better than the former and vice versa. As Algorithm 1 expresses, the XY-X-Y dimensional routing with multicast function is adopted as deterministic algorithm, in which the router always parses the

Algorithm 1: Deterministic Routing (XY-X-Y Dimensional Routing)

Input: n_{th} AER packet address ($\triangle XY_n$, $\triangle X_n$, $\triangle Y_n$) and (#XY_n, #X_n, #Y_n)

Output: n_{th} AER packet routing direction dir_n

for AER packet n∈1~N:
  if $\triangle XY_n > 0$
    $dir_n \leftarrow$ {0,1,0,0,0,0,0}, $\triangle XY_n \leftarrow \triangle XY_n$–1 (to XY+ direction)
  else if $\triangle XY_n < 0$
    $dir_n \leftarrow$ {0,0,1,0,0,0,0}, $\triangle XY_n \leftarrow \triangle XY_n$+1 (to XY– direction)
  else if #$XY_n > 0$
    $dir_n \leftarrow$ {1,1,0,0,0,0,0}, #$XY_n \leftarrow$ #XY_n–1 (to local and XY+ directions)
  else if #$XY_n < 0$
    $dir_n \leftarrow$ {1,0,1,0,0,0,0}, #$XY_n \leftarrow$ #XY_n+1 (to local and XY– directions)
  else
    if $\triangle X_n > 0$
      $dir_n \leftarrow$ {0,0,0,1,0,0,0}, $\triangle X_n \leftarrow \triangle X_n$–1 (to X+ direction)
    else if $\triangle X_n < 0$
      $dir_n \leftarrow$ {0,0,0,0,1,0,0}, $\triangle X_n \leftarrow \triangle X_n$+1 (to X– direction)
    else if #$X_n > 0$
      $dir_n \leftarrow$ {1,0,0,1,0,0,0}, #$X_n \leftarrow$ #X_n–1 (to local and X+ directions)
    else if #$X_n < 0$
      $dir_n \leftarrow$ {1,0,0,0,1,0,0}, #$X_n \leftarrow$ #X_n+1 (to local and X– directions)
    else
      if $\triangle Y_n > 0$
        $dir_n \leftarrow$ {0,0,0,0,0,1,0}, $\triangle Y_n \leftarrow \triangle Y_n$–1 (to Y+ direction)
      else if $\triangle Y_n < 0$
        $dir_n \leftarrow$ {0,0,0,0,0,0,1}, $\triangle Y_n \leftarrow \triangle Y_n$+1 (to Y– direction)
      else if #$Y_n > 0$
        $dir_n \leftarrow$ {1,0,0,0,0,1,0}, #$Y_n \leftarrow$ #Y_n–1 (to local and Y+ directions)
      else if #$Y_n < 0$
        $dir_n \leftarrow$ {1,0,0,0,0,0,1}, #$Y_n \leftarrow$ #Y_n+1 (to local and Y– directions)
      else
        $dir_n \leftarrow$ {1,0,0,0,0,0,0} (to local direction)

Algorithm 2: Adaptive Routing (Congestion-Aware Routing)

Input 1: n_{th} AER packet address ($\triangle XY_n$, $\triangle X_n$, $\triangle Y_n$) and (#XY_n, #X_n, #Y_n)

Input 2: FIFO credit *crd* from XY±, X± and Y± directions

Output: n_{th} AER packet routing direction dir_n

for AER packet n∈1~N:
  if ($\triangle XY_n$!= 0 or #XY_n != 0) and ($\triangle X_n$!= 0 or #X_n != 0) and ($\triangle Y_n$!= 0 or #Y_n != 0)
    $dir_n \leftarrow$ max(crd_{XY+} or crd_{XY-}, crd_{X+} or crd_{X-}, crd_{Y+} or crd_{Y-})
  else if ($\triangle XY_n$!= 0 or #XY_n != 0) and ($\triangle X_n$!= 0 or #X_n != 0) and ($\triangle Y_n$ == 0 and #Y_n == 0)
    $dir_n \leftarrow$ max(crd_{XY+} or crd_{XY-}, crd_{X+} or crd_{X-})
  else if ($\triangle XY_n$!= 0 or #XY_n != 0) and ($\triangle X_n$ == 0 and #X_n == 0) and ($\triangle Y_n$!= 0 or #Y_n != 0)
    $dir_n \leftarrow$ max(crd_{XY+} or crd_{XY-}, crd_{Y+} or crd_{Y-})
  else if ($\triangle XY_n$ == 0 and #XY_n == 0) and ($\triangle X_n$!= 0 or #X_n != 0) and ($\triangle Y_n$!= 0 or #Y_n != 0)
    $dir_n \leftarrow$ max(crd_{X+} or crd_{X-}, crd_{Y+} or crd_{Y-})
  else $dir_n \leftarrow$ the same as deterministic routing

XY-dimension firstly, following by X-dimension, and finally with the Y-dimension. When all the addresses return to zero, the packet is absorbed at its destination.

The self-adaptive routing is also based on the shortest path routing, except for some changes. As shown in Algorithm 2, It utilizes a pair of credit counters to indicate the busy levels of the adjacent routing nodes. When there are multiple choices of output directions, i.e., two or three of the ($\triangle XY_n$, $\triangle X_n$, $\triangle Y_n$) or (#XY_n, #X_n, #Y_n) addresses equal to non-zero, the router can transmit the packet to the neighbor with the lowest busy level. Besides, as the fully adaptive routing may come into deadlock without any turning restriction, the router also equips the XY± directions with two virtual channels (VC) to keep deadlock-free, where the packets to X+ and Y– directions must be sent to the first VC, while the packets to X– and Y+ directions must be sent to the second VC, just as Fig. 10 displays.

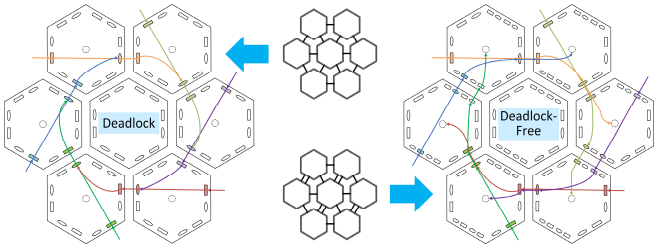


Fig. 10. Deadlock-free fully adaptive routing with 2 virtual channels at XY±.

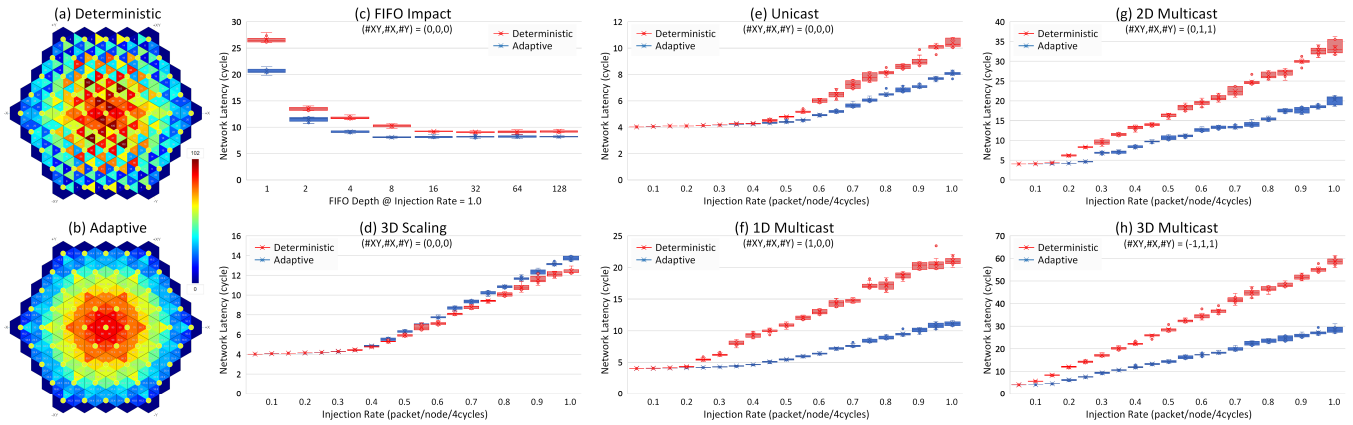


Fig. 11. Routing load and latency evaluation of NeuroHexa. (a) Load distribution of deterministic unicast routing in each direction. (b) Load distribution of adaptive unicast routing in each direction. (c) Network latency in random unicast case versus input FIFO depth. (d) Network latency in random unicast case versus injection rate for 3D scaling. (e) Network latency in random unicast case versus injection rate. (f) Network latency in random 1D multicast case versus injection rate. (g) Network latency in random 2D multicast case versus injection rate. (e) Network latency in random 3D multicast case versus injection rate.

V. Evaluation

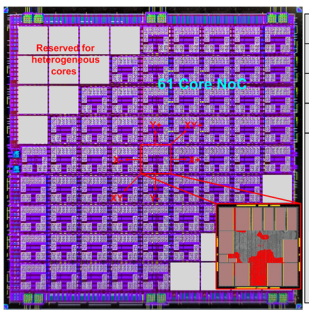
A. Quantitative Analysis

The presented NoC, NeuroHexa, is evaluated in detail for its routing load and latency as demonstrated in Fig. 11. Above all, Fig. 11(a) and Fig. 11(b) show the routing load distribution in XY±, X± and Y± directions for all the routing nodes by deterministic and adaptive routing algorithms, respectively. The adaptive routing exhibits isotropic distribution and also achieves a smaller load maximum, whereas the deterministic routing shows anisotropic distribution but still reserves some paths with less congestion. Precisely for this reason, in some special case such as 3D chiplet array scaling in Fig. 11(d), it is notable that the average network latency of the deterministic routing can even be better than that of the adaptive routing.

Generally, as Fig. 11(e)~(h) illustrates, the deterministic routing performs almost the same well as the adaptive routing with small injection rates. In this case, NeuroHexa is more suitable to be configured as the XY-X-Y dimensional routing for power-saving purpose. Nevertheless, when the injection rates of AER packets reach a certain degree, i.e., >0.4 in unicast, >0.2 in 1D multicast, >0.15 in 2D multicast, or >0.1 in 3D multicast routing, the deterministic routing may suffer from a steep latency rising due to severe routing congestion. Under the circumstances, NeuroHexa is more suitable to be configured as the congestion-aware routing for optimized routing latency. The configurable routing mode makes it possible for the NoC to better fit in different routing scenarios.

B. Hardware Implementation

The network latency of different FIFO depths is evaluated in Fig. 11(c), where one can find a router with a FIFO depth of 8 is sufficient for our NoC architecture. We implement the router in 28nm CMOS with FIFO depth and width as 8 words and 64 bits. Fig. 12 displays the chip layout of a neuromorphic core with the presented router highlighted in red color. The implementation results show that a single router node can achieve maximal throughput as 179.2Gbps at clock frequency of 800MHz within a limited area of 0.0226mm^2. And at the supply voltage of 0.81V, the router circuit can achieve the best energy efficiency as 4.872, 8.638, 12.520, 19.345, and 16.900 pJ/packet in unicast, 1D multicast, 2D multicast, 3D multicast in 2D plane, and 3D multicast cases, respectively.



		Specification
Technology		28nm CMOS
Router Area		0.0226mm^2
Router Throughput		179.2Gbps @ 800MHz
Router Energy Efficiency	Unicast	4.872pJ/packet @ 0.81V
	1D Multicast	8.638pJ/packet @ 0.81V
	2D Multicast	12.520pJ/packet @ 0.81V
	3D Multicast in 2D Plane	19.345pJ/packet @ 0.81V
	3D Multicast	16.900pJ/packet @ 0.81V

Fig. 12. Layout and specifications of the proposed NoC chip and router.

TABLE II. Comparison With Other Neuromorphic NoC

Design	[8]	[11]	[14]	[15]	**This work**
Technology	28nm	180nm	14nm	130nm	28nm
# of cores	4096	4	128	18	61
Topology	2D mesh	Quadtree	Clustered	Star	Hexagon
Chip Scaling	2D	2D	2D	2D	2D/3D
Multicast Capability	Unicast	Unicast/ multicast	Unicast	Unicast/ multicast	Unicast/ multicast
Routing Algorithm	Determ.	Determ.	Determ.	Determ.	Determ./ Adaptive
Throughput	5.44Gbps	51M events/s	3.44G spikes/s	5.3Gbps	179.2Gbps
Energy Cost	2.3pJ	17pJ	N.A.	1.1nJ	4.872pJ

A comparison with other neuromorphic NoC counterparts is listed in Table. II. NeuroHexa can achieve more functions, including 2D/3D scaling, multicast and adaptive routing, with much higher throughput and relatively lower energy cost.

VI. Conclusion

This work presents NeuroHexa, a novel NoC multi-core architecture for neuromorphic computing. NeuroHexa adopts a hexagonal topology with six pairs of bidirectional links, and is also equipped with the capability of both 2D and 3D chiplet integration. Following GALS methodology, the NoC realizes group thread control mechanism in order to improve resource utilization. In accordance with the unstructured spike dataflow, it proposes a flexible unicast and multicast routing mechanism, which meets the requirements of the diverse dimensional data reuse cases. Moreover, to better fit in the complex congestion scenarios, it can be configured as either deterministic XY-X-Y dimensional routing or adaptive congestion-aware routing mode for power-first or latency-first purpose, respectively. The implementation results in 28nm CMOS indicate that the router can achieve the maximal throughput of 179.2Gbps, as well as the best energy efficiency of 4.872pJ/packet in unicast routing, within the area overhead of 0.0226mm^2.

REFERENCES

[1] K. Roy, A. Jaiswal, and P. Panda, "Towards spike-based machine intelligence with neuromorphic computing," *Nature*, vol. 575, pp. 607–617, November 2019.

[2] J. Yang, R. Wang, Y. Ren, J. Mao, Z. Wang, Y. Zhou, and S. Han, "Neuromorphic engineering: from biological to spike-based hardware nervous systems," *Advanced Materials*, vol. 32, no. 52, pp. 1–32, December 2020.

[3] G. Indiveri, F. Corradi, and N. Qiao, "Neuromorphic architectures for spiking deep neural networks," in *IEEE International Electron Devices Meeting (IEDM)*, pp. 4.2.1–4.2.4, 2015.

[4] A. R. Young, M. E. Dean, J. S. Plank, and G. S. Rose, "A Review of Spiking Neuromorphic Hardware Communication Systems," *IEEE Access*, vol. 7, pp. 135606–135620, September 2019.

[5] D. V. Christensen, R. Dittmann, B. Linares-Barranco, A. Sebastian, M. L. Gallo, A. Redaelli, S. Slesazeck et al., "2022 roadmap on neuromorphic computing and engineering," *Neuromorphic Computing and Engineering*, vol. 2, pp. 1–112, May 2022.

[6] A. Basu, C. Frenkel, L. Deng, and X. Zhang, "Spiking neural network integrated circuits: a review of trends and future directions," in *IEEE Custom Integrated Circuits Conference (CICC)*, pp. 1–8, 2022.

[7] D. B. Fasnacht, A. M. Whatley, and G. Indiveri, "A serial communication infrastructure for multi-chip address event systems," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 648–651, 2008.

[8] F. Akopyan, J. Sawada, A. Cassidy, R. Alvarez-Icaza, J. Arthur, P. Merolla et al., "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, October 2015.

[9] L. Deng, G. Wang, G. Li, S. Li, L. Liang, M. Zhu et al., "Tianjic: a unified and scalable chip bridging spike-based and continuous neural computation," *IEEE Journal of Solid-State Circuits*, vol. 55, no. 8, pp. 2228–2246, August 2020.

[10] D. Ma, X. Jin, S. Sun, Y. Li, X. Wu, Y. Hu et al., "Darwin3: a large-scale neuromorphic chip with a novel ISA and on-chip learning," *National Science Review*, vol. 11, pp. 1–17, March 2024.

[11] S. Moradi, N. Qiao, F. Stefanini, and G. Indiveri, "A scalable multicore architecture with heterogeneous memory structures for dynamic neuromorphic asynchronous processors (DYNAPs)," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 12, no. 1, pp. 106–122, February 2018.

[12] C. Frenkel, J. Legat, and D. Bol, "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*, vol. 13, no. 5, pp. 999–1010, October 2019.

[13] Y. Zhong, Y. Kuang, K. Liu, Z. Wang, S. Feng, G. Chen et al., "PAICORE: A 1.9-million-neuron 5.181-TSOPS/W digital neuromorphic processor with unified SNN-ANN and on-chip learning paradigm," *IEEE Journal of Solid-State Circuits*, Early Access, pp. 1–21, 2024.

[14] M. Davies, N. Srinivasa, T. Lin, G. Chinya, Y. Cao, S. H. Choday et al., "Loihi: a neuromorphic manycore processor with on-chip learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, January/February 2018.

[15] E. Painkras, L. A. Plana, J. Garside, S. Temple, F. Galluppi, C. Patterson et al., "SpiNNaker: a 1-W 18-core system-on-chip for massively-parallel neural network simulation," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 8, pp. 1943–1953, August 2013.

[16] G. K. Chen, R. Kumar, H. E. Sumbul, P. C. Knag, and R. K. Krishnamurthy, "A 4096-neuron 1M-synapse 3.8-pJ/SOP spiking neural network with on-chip STDP learning and sparse weights in 10-nm FinFET CMOS," *IEEE Journal of Solid-State Circuits*, vol. 54, no. 4, pp. 992–1002, April 2019.

[17] P. J. Zhou, Q. Yu, M. Chen, Y. C. Wang, L. W. Meng, Y. Zuo et al., "A 0.96pJ/SOP, 30.23K-neuron/mm^2 heterogeneous neuromorphic chip with fullerene-like interconnection topology for edge-AI computing," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5, 2024.

[18] J. Pu, W. L. Goh, V. P. Nambiar, and A. T. Do, "A low power and low area router with congestion-aware routing algorithm for spiking neural network hardware implementations," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 68, no. 1, pp. 471–475, January 2021.