IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 10 June 2024, accepted 3 July 2024, date of publication 10 July 2024, date of current version 17 September 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3425900

RESEARCH ARTICLE

BTSAM: Balanced Thermal-State-Aware Mapping Algorithms and Architecture for 3D-NoC-Based Neuromorphic Systems

MOHAMED MAATAR, (Student Member, IEEE), ZHISHANG WANG, (Member, IEEE), KHANH N. DANG, (Member, IEEE), AND ABDERAZEK BEN ABDALLAH, (Senior Member, IEEE)
Graduate School of Computer Science and Engineering, Neuroengineering Laboratory, The University of Aizu, Aizuwakamatsu 965-0006, Japan

Corresponding authors: Mohamed Maatar (d8232110@u-aizu.ac.jp) and Abderazek Ben Abdallah (benab@u-aizu.ac.jp)

This work was supported in part by The University of Aizu, Competitive Research Funding (CRF) under Grant UoA-P7-2023; and in part by the VLSI Design and Education Center, The University of Tokyo, Japan, in Collaboration with Synopsys, Inc., and Cadence Design Systems, Inc.

ABSTRACT Neuromorphic computing systems are biologically inspired approaches created from many highly connected neurons to model neuroscience theories and solve machine learning problems. They promise to drastically improve the efficiency of critical computational tasks such as decision-making and perception. Combining neuromorphic computing systems and 3D interconnect (3D-NoC) technology leads to an advanced architecture that inherits the benefits of both computing and interconnect paradigms. However, designing large-scale neuromorphic systems based on 3D-NoC faces several challenges, including thermal power, power distribution, increased power density at the heat sink interface, and fabrication requirements. This work tackles the thermal issues in designing large-scale neuromorphic systems by proposing a Balanced Thermal-State-Aware Mapping (BTSAM) for 3D-NoC-based neuromorphic systems. This includes a Periodic Activity Scoring (PAS), a Seesaw Neuron Clustering (SNC) method, and a thermal-aware genetic algorithm to eliminate hotspots, balance the thermal state, and lower the temperature while keeping the system's accuracy acceptable. Evaluation results on various system configurations demonstrate a notable up to 12.4 K and 5.2 K temperature reduction compared to linear methods and HeterGenMap, respectively, and a 4× increase in Mean-Time-to-Failure (MTTF), with an acceptable power and area overheads and little degradation of the communication cost.

INDEX TERMS Neuromorphic systems, mapping, thermal-state-aware, 3D-NoC, genetic algorithm.

I. INTRODUCTION

Neuromorphic systems implement biologically inspired computing models such as Spiking Neural Networks (SNNs) that mimic the parallelism and efficiency of the human brain [1], [2]. SNNs are third-generation neural networks in which information is encoded and transmitted via synapses as spikes [3]. SNNs consume lower power through spike-driven processing than other artificial neural networks (ANNs) [4]. Moreover, with the increased interest in event-based data and sensors, SNNs are promising in processing such data inputs. Even though ANNs still have better accuracy results, research is being done to implement new learning techniques and create databases suited for SNNs [4], [5]. Efforts to integrate SNNs on a single chip have led to notable implementations such as Loihi and TrueNorth [6], [7], [8], [9], [10]. While software simulations provide flexibility, performance acceleration is paramount for high-efficiency and real-time processing applications. Hardware implementation on dedicated neuromorphic hardware improves energy efficiency and execution speed, making these systems ideal for power-sensitive applications such as mobile and IoT devices [11], [12]. The increasing demand for large-scale neural networks in emerging edge applications requires

The associate editor coordinating the review of this manuscript and approving it for publication was Muhammad Sharif.

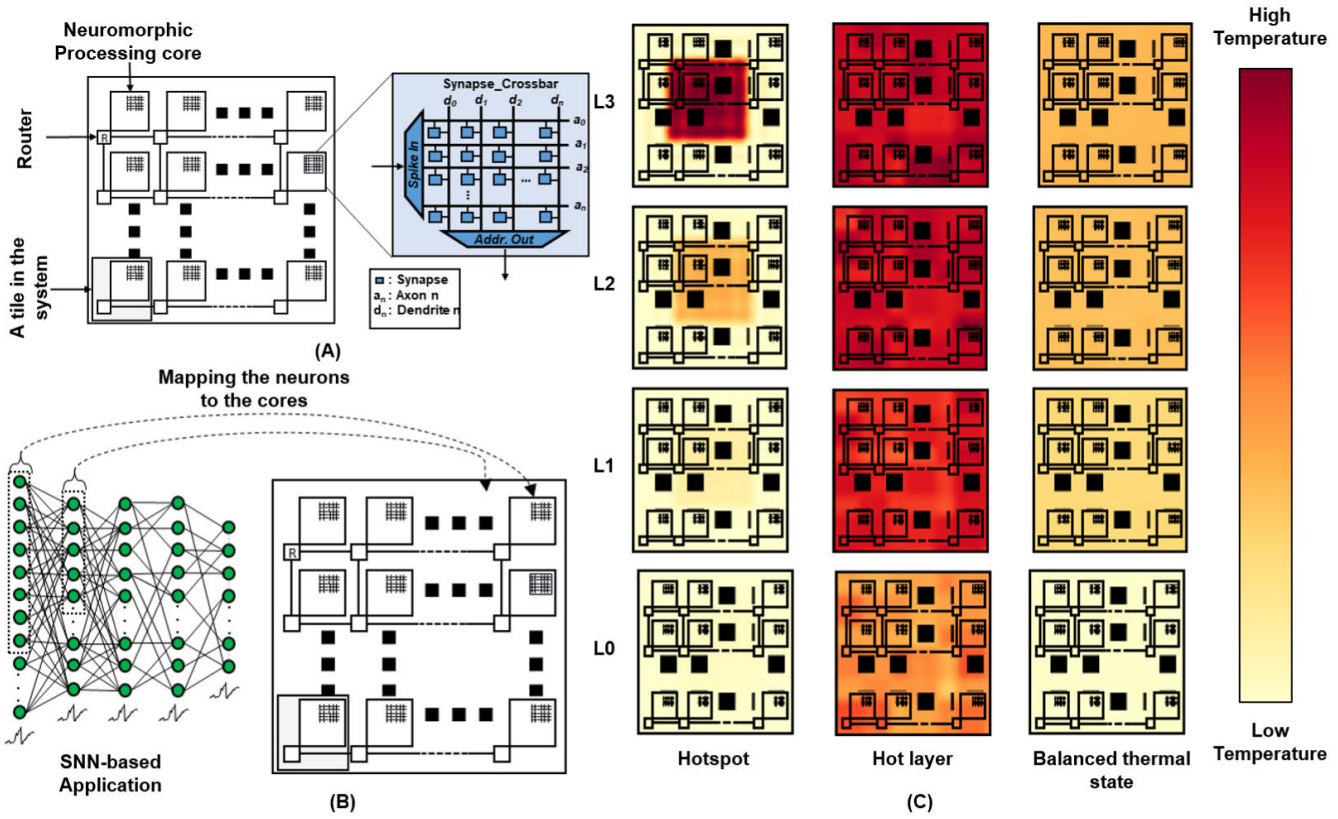

FIGURE 1. Mapping effect on the system's temperature. (A) Components of a NoC-based neuromorphic system. (B) Mapping SNN applications to neuromorphic processing cores. (C) Different thermal state scenarios are caused by different mapping strategies: Example 1 (Hotspot), Example 2 (Hot layer), and Example 3 (Balanced thermal state).

scalable solutions. To address this issue, designers have adopted network-on-chip (NoC) interconnects, particularly 3D-NoC, which provide multiple traversal paths for packet communication and efficient routing. Despite advances in hardware and circuitry speed, the design challenge of neuromorphic systems is to map applications to optimize functionality and maintain peak performance effectively. The pursuit of scalability through 3D interconnects, achieved by stacking 2D dies connected vertically with Through-Silicon-Vias (TSVs), brings new challenges, especially in terms of thermal considerations [13].

A. MOTIVATION

3D-ICs exhibit higher power density, exacerbating the thermal challenges compared to 2D structures [13]. The heterogeneous heat transfer in 3D-ICs due to the different lengths of the heat conduction paths increases the risk of thermal thresholds being exceeded [14], [15], [16]. This leads to an unbalanced thermal state in the 3D-IC. This contributes to accelerated material degradation and increased system faults, affecting overall robustness and reliability [17], [18]. Overcoming these thermal issues is critical to the longevity and reliability of systems. However, executing applications on a neuromorphic system requires mapping them to it but presents an NP-hard problem [19]. Furthermore, the scalability of neuromorphic systems, achieved through NoC interconnect adoption, complicates the mapping task. Fig. 1 shows three examples of thermal states in 3D neuromorphic systems. Fig. 1(A) shows the main components of a NoC-based neuromorphic system, where each system tile comprises a router and a neuro-processing core. In each processing core, there is a synapse crossbar wherein the synaptic operations take place. Fig. 1(B) illustrates an example of mapping an SNN-based application to a neuromorphic system, in which neurons from each layer are selected and mapped to different neuro-processing cores. Fig. 1(C) presents three different cases for the temperature of a 3D neuromorphic chip. The first example results from mapping a large amount of activity in a small area, leading to a hotspot in the system. The second example arises when an extensive application is mapped in the furthest layers from the heat sink, resulting in a hot layer that affects the other layers. Finally, an example of a system in which the application is strategically mapped to balance the energy consumption and achieve a balanced thermal state in the 3D neuromorphic system.

Two mapping methods have been widely used in recent decades due to their simplicity: *Linear mapping* from SpiNNaker [20]. It consists of placing the neurons in linear

order in the crossbars. And *linear layer-to-layer mapping*, which maps each layer of the SNN linearly to a layer of the chip [21]. Despite their organizational benefits, these methods pose considerable problems during system implementation, namely:

1) The crossbars in the system run the risk of having an *unbalanced energy consumption* due to the different neuronal activity from one layer to another or from a core to another. Consequently, this leads to hotspots in the system, as shown in the first example in Fig. 1(C).
2) In the 3D-IC, the top layer of the chip (furthest from the heat sink) tends to exceed the thermal thresholds and reach a peak temperature, which also affects nearby layers, as shown in the second example of Fig. 1(C).

To address the issues above, this work proposes a Balanced Thermal-State-Aware Mapping (BTSAM) method to eliminate hotspots, balance the system's temperature, and lower the maximum temperature. A thermal model is also proposed to estimate the system's energy generation and heat transfer. The contributions of this work are summarised as follows:

1) Balanced Thermal-State-Aware Mapping (BTSAM) method.
2) Periodic Activity Scoring (PAS) assigns a score to each neuron based on its activity over time.
3) Seesaw Neuron Clustering (SNC) method, which moderates the activity of the clusters within the system.
4) A thermal-aware genetic algorithm method for mapping clusters to crossbars.
5) A thermal analytical model to evaluate the system's temperature based on energy consumption and heat transfer between tiles.

The paper is organized as follows: Section II reviews related works. Section III provides an overview of the baseline 3D-NoC neuromorphic system. The proposed method is detailed in Section IV, followed by a thermal analytical model in Section V. The evaluation is provided in Section VI. The conclusion is presented in Section VII.

II. RELATED WORKS

This section presents strategies for SNN mapping to neuromorphic hardware, followed by an overview of thermal-aware mapping techniques in 3D IC-based systems.

A. SNN MAPPING TO NEUROMORPHIC HARDWARE

Mapping SNN neuromorphic hardware is an NP-hard problem [22] and can be represented as a Quadratic Assignment Problem (QAP) [23]. Therefore, the use of meta-heuristic methods such as genetic algorithm (GA) [24] and Particle Swarm Optimization (PSO) [25] is mandatory. The authors in [19] and [26] used PSO and Kernighan-Lin for SNN partitioning, aiming to minimize global edge usage and maximize crossbar utilization. While reducing the communication cost, these methods increase the activity of the crossbars and, thus, the power density. With millions of neurons in an application, the high-dimensional search space often traps optimization methods in local minima.

A common practice proposed by Balaji et al. [11] when mapping SNN to neuromorphic systems is to split the process into clustering and then placement. The authors proposed SpiNeCluster to improve performance by reducing global synapse spikes, while SpiNePlacer optimally places synapses using meta-heuristics to minimize routing energy consumption and system latency. However, maximizing activity in crossbars can lead to hotspots in the system.

The works in [27] and [28] proposed a genetic algorithm to minimize the communication cost within a 3D-IC system, providing offline and online remapping methods to improve reliability and fault tolerance. Nevertheless, the technique overlooked the power consumption within the crossbars and their thermal state.

B. THERMAL-AWARE MAPPING FOR 3D SYSTEMS

High temperature affects the functioning and the reliability of systems [17]. Therefore, addressing this problem is imminent. In conventional architectures, task allocation and scheduling are critical components of Dynamic Thermal Management (DTM) for 3D-ICs due to the similar temperatures of vertically adjacent tiles [29].

Hamedani et al. [30] proposed three thermal-aware mappings for 3D-NoC using branch and bound method to solve the integer linear programming (ILP) problem and a partitioning-based mapping in which they partition the processing elements (PE). However, these methods assume the application can be divided into tasks with known communication requirements. Li et al. [13] also tackled the issue by dividing the NoC into 3D-cuboids close to each other to enable faster communication and are closest to the heat sink and further from other applications. However, this work focused on mapping different applications to the system instead of mapping a large task number into different cores. Both methods referenced above are employed in conventional architectures, yet large-size 3D neuromorphic systems face distinct challenges and require different system designs. For instance, mapping large SNN applications to neuromorphic systems presents a significantly greater complexity than task mapping in conventional architectures due to the dense neuron-to-core ratio and the high core count [22]. Due to the high neuron count, task migration becomes expensive in large neuromorphic systems. Furthermore, the throttling of cores in such systems can degrade accuracy. Consequently, the mapping process in neuromorphic systems is a highly complex and crucial phase. Beigi and Memik [31] investigated the effect of temperature on ReRAM-based neuromorphic systems and introduced a temperature-aware strategy by avoiding mapping significant weights to thermally stressed cells. Zhang et al. [32] proposed reordering the crossbar's columns based on the weight and input distribution to reduce peak temperature. However, neither strategy addresses the thermal challenges that can arise in 3D architectures or the adaptability of these methods to large-scale application mapping. Our prior work in [33] aimed to reduce the variance

between the system tiles by clustering the neurons based on their activity and then randomly placing them evenly in the system. However, the randomness of the mapping process should be optimized.

III. NEUROMORPHIC SYSTEM (NASH) OVERVIEW

This section overviews the baseline neuromorphic system and briefly discusses the main components. In addition, the sequence of operations performed by the Network Interface (NI) that supports the mapping method is discussed. As shown in Fig. 2, the baseline neuromorphic system is 3D NoC-based with adaptation capabilities based on the system in [21] and [27]. The system consists of interconnected neural tiles in a mesh architecture. The neurons' initial configurations and weights are fed to the system via an external host CPU. The neural tiles comprise a Spiking Neurons Processing Core (SNPC) and a 3D multicast router. In the SNPC, the NI supports the mapping method.

A. SPIKING NEURON PROCESSING CORE (SNPC)

The SNPC, as shown in Fig. 2(C), consists of several components, namely a Leaky-Integrate-and-Fire (LIF) array, a synapse memory, a synapse crossbar, an NI, a control unit and a Spike Time-Dependent Plasticity (STDP) learning module. In the LIF array, an output spike is generated when the membrane voltage reaches a particular threshold value, initiating the refractory period. The synapse crossbar performs parallel neuron updates, allowing neurons to update in a single cycle. Moreover, the synapse crossbar stores all synapses associated with a single neuron. When a spike is output, the NI provides the memory address for the associated synapses. The synapse crossbar then fetches the synapse values stored at these addresses based on a neuron mask and forwards them to the corresponding post-synaptic neurons for updating. The STDP learning module implements a trace-based learning rule [34] for each synapse of the synapse crossbar. In the learning rule, the weights are updated eight steps before and after a spike event. The control unit is based on a finite state machine (FSM) with seven distinct states. The initial state is idle, in which the SNPC remains inactive and waits for input spikes. Once a pre-synaptic spike array is received, the control unit transitions to the second state, allowing the NI to process the spike array. In the third state, the crossbar is activated, facilitating the identification and update of post-synaptic neurons. The fourth state enables the decay of membrane voltage values—subsequently, the fifth state checks for the occurrence of output spikes. If spikes are detected, the control unit enters the sixth state, transmitting the spikes to the network interface. Finally, the control unit enters the seventh state, where the learning process is initiated, and returns to the idle state after completion.

B. 3D ROUTER

The 3D router architecture has seven input and output ports. These ports include one dedicated to the local SNPC, four for connecting to neighboring routers in the north, south, east, and west directions, and two for vertical connections to neighboring layers. The packet routing process within the router consists of four stages: buffer writing, routing calculation, port allocation via the switch-allocator, and crossbar traversal for transmitting the packet to the designated output port. For spike routing, the router supports two routing algorithms, namely, the K-means-based multicast routing algorithm (KMCR) and the shortest path K-means-based multicast routing algorithm (SP-KMCR) [35]. The KMCR algorithm adopts a clustering approach in which the nodes are grouped into clusters. The cluster centroid is determined by selecting the node with the lowest mean distance from all other nodes within the cluster. When a spike packet is destined for a node within a cluster, it is first routed to the centroid node and then forwarded to the intended destination node. The SP-KMCR algorithm, on the other hand, selects the cluster centroid based on the node with the shortest distance to the source node of the spike packet.

C. MAPPING SCHEMES IN THE NETWORK INTERFACE

As shown in Fig. 2(D), the NI has an encoder and a decoder. The NI categorizes incoming spikes into input spikes or memory access commands. In the case of input spikes, the decoder identifies the corresponding address in the weight SRAM and sends it to the weight memory. The encoder translates spike outputs to packets and then forwards them to other neural tiles via the interconnect. The look-up table (LUT) organizes and stores the configuration of the neurons, specifying their memory addresses and connections. When mapping neurons to the neuromorphic system, the host CPU sends the synapses' weights to the NI. The decoder uses the LUT to identify the address in the weight memory and writes/reads the value to/from the memory.

IV. BTSAM METHODOLOGY

This section presents the proposed Balanced Thermal-State-Aware Mapping (BTSAM) method, which comprises three phases to map an SNN to the neuromorphic system.

- A Periodic Activity Scoring (PAS) assigns a score to each neuron based on its windowed activity over time to capture its internal activity.
- A Seesaw Neuron Clustering (SNC) clusters the neurons based on their score to balance the workload and moderate the energy consumed across the system.
- A genetic algorithm-based mapping for clusters to tiles aims to reduce the maximum and average temperature and the system's thermal state variance.

Comprehensive details on these three phases are provided in the following subsections.

A. MAPPING PROCESS TO THE NOC-BASED NEUROMORPHIC SYSTEM

Mapping SNNs to the NoC-based neuromorphic system begins with implementing the SNN model, encompassing neuron models, thresholds, and learning rules. The model's weights and inputs are then adjusted to ensure hardware

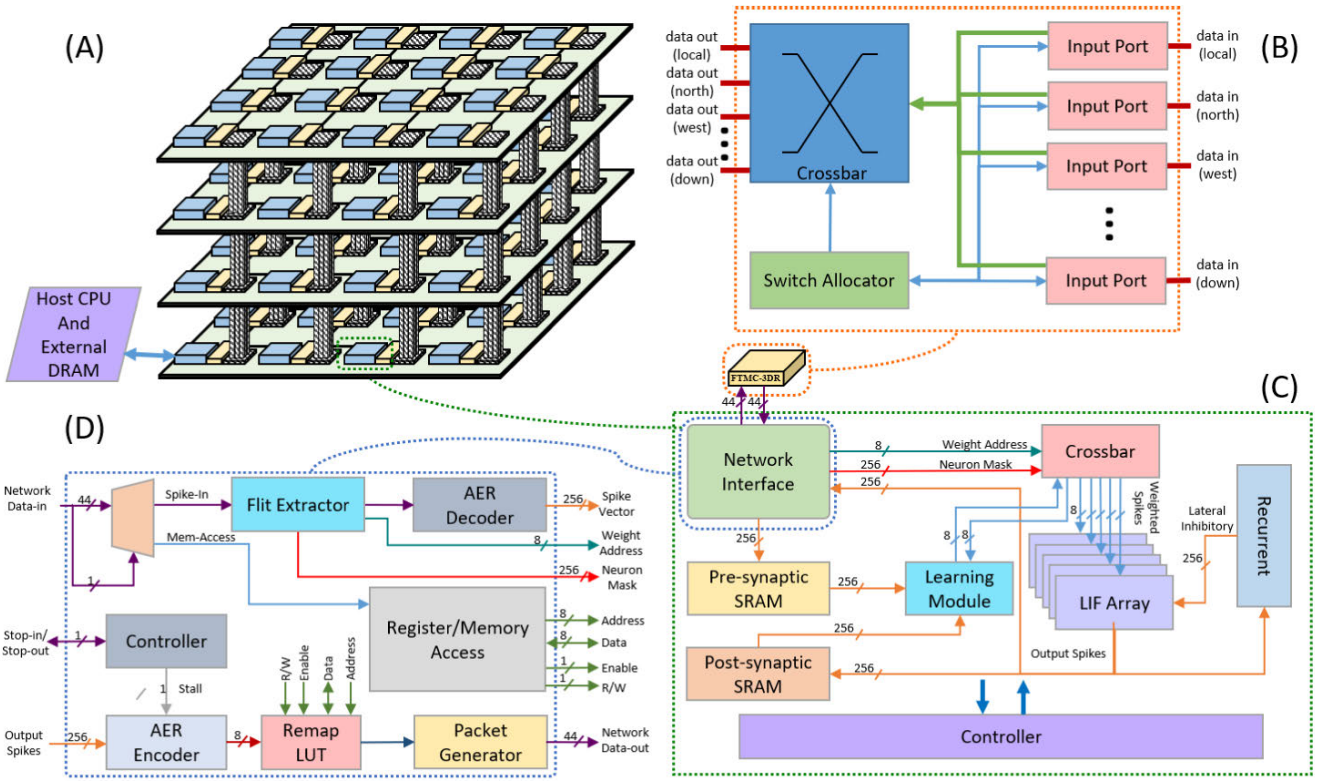

FIGURE 2. High-level view of 3D NoC-based neuromorphic system (NASH) architecture: (A) 4 × 4 × 4 system configuration, (B) 3D fault-tolerant multicast router, (C) Spiking neurons processing core, (D) Network interface [21], [27].

compatibility. The neurons are then systematically clustered based on predefined rules and strategically positioned within the crossbars. Once the model weights have been adjusted, the weights and neuron addresses are uploaded to the memory cells. At the same time, the packets are encoded and decoded via the network interface. This systematic approach ensures the seamless and efficient operation of the neuromorphic system.

B. PERIODIC ACTIVITY SCORING (PAS) ALGORITHM

The Periodic Activity Scoring (PAS) algorithm computes a score vector of neurons based on their activities within designated time windows of a given spike train. This method systematically identifies the most active neurons across different periods. Algorithm 1 formalizes the PAS. Inputs are spike trains (ST), representing the neural activity of an SNN with N_a neurons over time and the desired size of the time window t for evaluating this activity. First, PAS counts the spikes in each time window and creates a matrix representing the windowed spike count wsc. Then, the neurons are ranked according to their activity levels within these windows to generate the matrix $rank\_wsc$. Subsequently, the cumulative score for each neuron is calculated by summing its ranks across all time windows. The output is a score vector S that represents the activity level of each neuron over the observation period.

Algorithm 1 Periodic Activity Scoring (PAS)

```
/* spike trains                    */
Input  : ST
/* Time Window size                */
Input  : t
/* Neurons score                   */
Output: S
wsc ← count spikes in each t;
rank_wsc ← rank neurons in each time window in
  wsc based on spike count;
foreach n ∈ N_a do
    S[n] ← Σ_{i=0}^{t} rank_wsc[n, i]
return S
```

Fig. 3 shows how the PAS algorithm attributes a score to each neuron. The illustration provides a case study featuring spike trains generated by an SNN comprising 16 neurons ($N1$ to $N16$). As shown in Fig. 3(A), the original spike trains are presented over time, with the vertical axis denoting the neurons and the horizontal axis indicating the time windows (tw_1 to tw_4). Fig. 3(B) shows the spike counts aggregated within the specified time windows. The spike trains are divided into sections based on the time window size t. Within these sections, the spike counts are aggregated for each neuron. Fig. 3(C) displays the rank of each neuron within

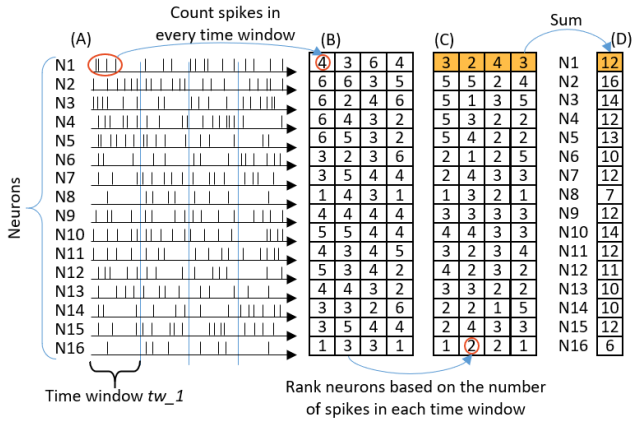

FIGURE 3. Example of the Periodic Activity Scoring: (A) represents the spike trains of each neuron. (B) represents the spike counts in each time window. (C) is each neuron's rank in each time window based on its spike count. (D) is the final score of the neurons, which is the sum of the ranks of the neurons in all time windows.

each time window based on its spike count. Each neuron is assigned a rank, where 1 represents the lowest spike count within the window. Fig. 3(D) illustrates the final score of the neurons. The score for each neuron is the sum of its ranks across all time windows. For example, $N1$ has a spike count of 4 in the first time window, ranking it in third place compared to the other neurons within the same window. In the following time windows, its ranks are based on the spike counts being 2, 4, and 3, respectively. Adding these ranks yields a final score of 12 for $N1$, as shown in Fig. 3(D). This score reflects the neuron's relative activity compared to other neurons over the entire period the algorithm considers.

The time complexity of PAS is $\mathcal{O}(PAS) = \mathcal{O}(N \times M \times t)$, where N, M, and t represent the total number of neurons, the number of time windows, and the size of each time window, respectively.

C. SEESAW NEURON CLUSTERING (SNC) ALGORITHM

Algorithm 2 formalizes the SNC, where neurons are assigned to clusters based on their scores. The number of crossbars in the system determines the number of clusters. SNC operates with three primary inputs: scores vector (S), the total number of neurons (N), and the number of crossbars ($Xbars$), which defines the number of clusters. The output is a cluster vector (C), wherein each element denotes a neuron's cluster. Initially, the algorithm sets C and a counter (x) to zero. It then applies the *sort_arguments* method to S, which returns the ordered indexes based on their respective values, ranking neurons by their scores and returning their sorted indices ($sorted\_S$). The algorithm iteratively assigns each neuron index n in $sorted\_S$ to a cluster, incrementing x subsequently. The variable x represents the current cluster cycles through the available clusters, controlled by a Boolean flag *reverse*. If x exceeds the value $Xbars$, it is set to $Xbars$ and *reverse* to *True*; If x becomes negative, it is set to 0 and *reverse* to *False*.

Algorithm 2 Seesaw Neuron Clustering - SNC

```
/* score, number of neurons and
   crossbars                                  */
Inputs : S, N, Xbars
/* Clustering of neurons                      */
Output: C
Init: C_N ← [0]_N
Init: x ← 0
Init: reverse ← False
sorted_S ← sort_arguments(S);
foreach n ∈ sorted_S do
    C[n] ← x;
    if reverse then
        x − −;
    else
        x + +;
    if x > len(Xbars) then
        reverse ← True;
        x ← Xbars
    else if x < 0 then
        reverse ← False
        x ← 0
return C
```

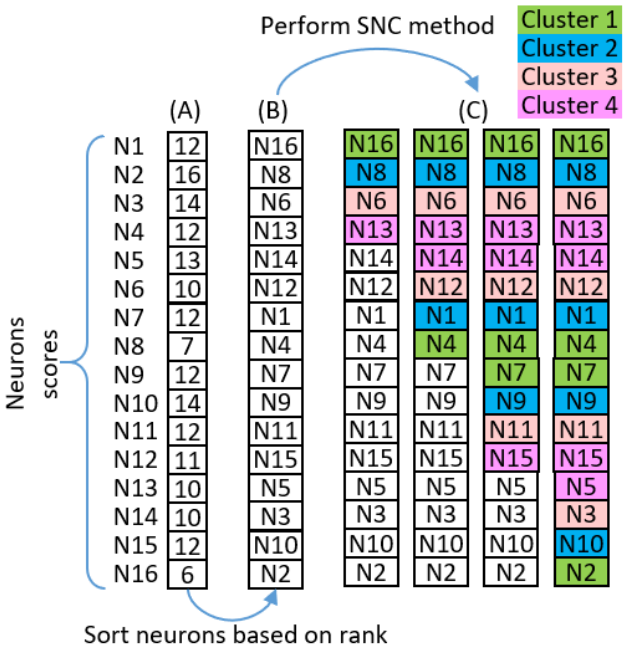

FIGURE 4. Example of the SNC: (A) represents the neuron's score from PAS. (B) represents the ordered neurons based on their respective scores. (C) is the clustering result after we change the clusters' order each time.

This mechanism ensures a balanced distribution of neural activity in the defined clusters.

The example shown in Fig. 4 continues the case study introduced in Section IV-B. In this scenario, it is assumed that

the system consists of four crossbars. Fig. 4(A) represents the scores of each neuron derived from the PAS method in the earlier example as shown in Fig. 3. Subsequently, Fig. 4(B) shows the neurons arranged in ascending order of their scores. Fig. 4(C) illustrates the clustering operation. The neuron N16, with the lowest score, is initially assigned to cluster 1. The subsequent neurons, N8, N6, and N13, are assigned to clusters 2, 3, and 4, respectively. Then the *reverse* Boolean value changes, x begins to decrease, and the neurons N14 and N12 are assigned to clusters 4 and 3 until all neurons are distributed to the clusters.

The time complexity of SNC is $\mathcal{O}(SNC) = \mathcal{O}(N \log N)$ with N representing the total number of neurons.

D. THERMAL-AWARE CLUSTER MAPPING

To perform the thermal-aware cluster mapping after the generation of clusters, we represent the mapping as a matrix $P_{ij} = \{0, 1\}^{Xbars \times Xbars}$, where each row and column of the matrix represents a crossbar and a cluster, respectively. The matrix P_{ij} is defined as follows:

$$P_{ij} = \begin{cases} 1 & \text{if cluster } j \text{ is mapped to crossbar } i \\ 0 & \text{otherwise} \end{cases}$$

The genetic algorithm respects the following two constraints:

- A crossbar can contain only one cluster:

$$\sum_{i} P_{ij} = 1 \tag{1}$$

- A cluster can only be mapped into one crossbar:

$$\sum_{j} P_{ij} = 1 \tag{2}$$

Fig. 5 illustrates the flowchart of the GA for the cluster to crossbar mapping. The inputs include the number of generations, population size, and mutation rate. The GA initiates by generating an initial population and evaluating each individual's fitness. The objective is to minimize the fitness function, which is defined as follows:

$$Min(F) = Max(T) + \frac{Avg(T)}{2} + Var(T) \tag{3}$$

where T is the set of temperatures of the crossbars in the system. The GA method uses this fitness function to minimize the maximum temperature, followed by the average temperature and, finally, the variance. The new population is created from the crossover and mutation. To ensure adherence to constraints, we employ the two-point order crossover (OX) technique [36]. OX is well suited for combinatorial problems as it produces offspring that preserve the uniqueness of the elements. We use the inverse mutation [37], which reverses the order of the clusters between two randomly selected indices in the individual. We set the population size, mutation rate, and selection mechanism to 200, 0.01, and tournament selection, respectively. Two stopping criteria are employed: The algorithm halts if there are no improvements after a certain number of repetitions or when the maximum number of generations is reached. Upon completion of the evolution loop, the best individual is selected from all generations. Where T is the set of temperatures of the crossbars in the system, the GA method uses this fitness function to minimize the maximum temperature, followed by the average temperature and, finally, the variance. The new population is created from the crossover and mutation. To ensure adherence to constraints, we employ the two-point order crossover (OX) technique [36]. OX is well suited for combinatorial problems as it produces offspring that preserve the uniqueness of the elements. We use the inverse mutation [37], which reverses the order of the clusters between two randomly selected indices in the individual. We set the population size, mutation rate, and selection mechanism to 200, 0.01, and tournament selection, respectively. Two stopping criteria are employed: The algorithm halts if there are no improvements after a certain number of repetitions or when the maximum number of generations is reached. Upon completion of the evolution loop, the best individual is selected from all generations. Fig. 6(A) shows the clusters obtained by applying the PAS and SNC methods to the 16-neuron SNN, as presented in Section IV-B. Each gene in the chromosome representation corresponds to a tile in the system. We place each cluster

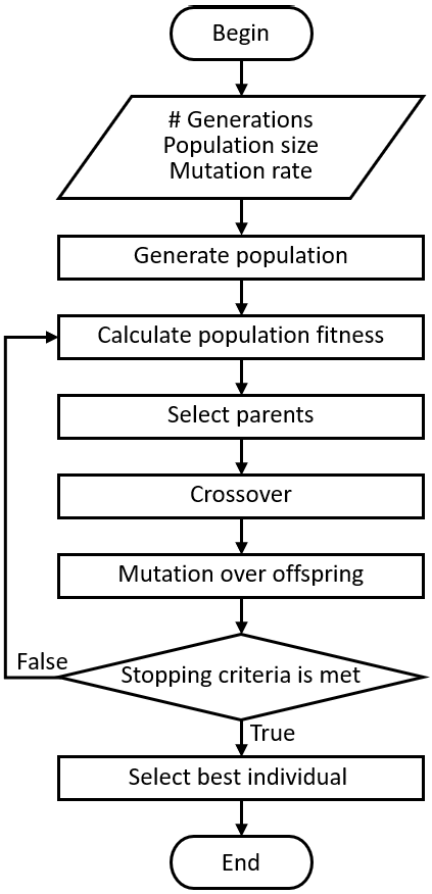

FIGURE 5. Used genetic algorithm flowchart for mapping the clusters into the crossbars.

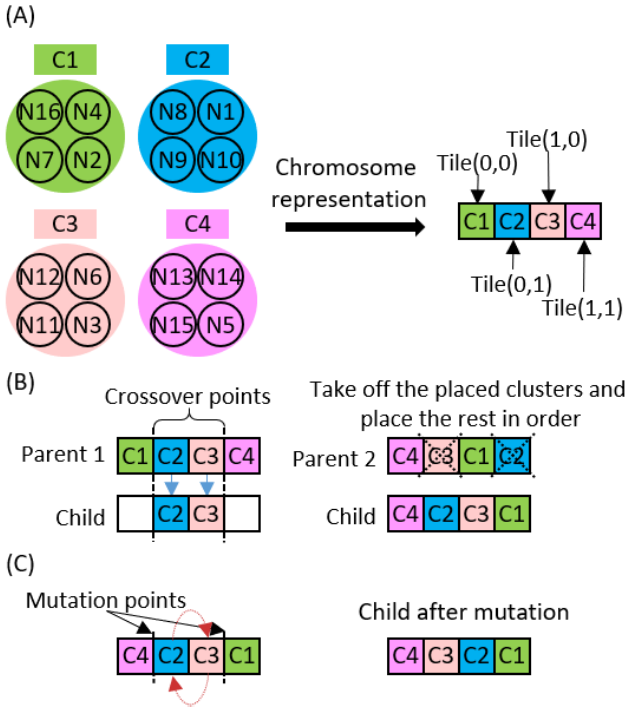

FIGURE 6. Example of chromosome representation, crossover, and mutation operations in the proposed GA method. (A) Chromosome representation from the clusters. (B) Ordered crossover. (C) Inverse mutation.

in one gene of the chromosome; For instance, cluster C1 is placed in tile(0,0), C2 in tile(0,1), C3 in tile(1,0), and C4 in tile(1,1). Fig. 6(B) illustrates the *OX* operation for the parents [C1, C2, C3, C4] and [C4, C3, C1, C2], where the crossover points are at indices 1 and 2. This process runs as follows: Initially, the clusters [C2, C3] from the first parent are selected and positioned in the corresponding slots of the offspring. Subsequently, the remaining slots in the offspring are filled with genes from the second parent, maintaining their order and preventing duplication. Consequently, C4 occupies the first vacant position and C1 the last, producing the offspring [C4, C2, C3, C1]. This method adeptly merges segments from both parents, ensuring a unique set of genes in the offspring. Fig. 6(C) illustrates an instance of the inverse mutation applied to the offspring. In this example, the mutation points are at indices 1 and 2, reversing the sub-sequence [C2, C3] to [C3, C2], which is subsequently reinserted into the offspring. The result is the mutated chromosome [C4, C3, C2, C1].

The time complexity of the thermal-aware mapping genetic algorithm is $\mathcal{O}(\text{Thermal-Aware Mapping GA}) = \mathcal{O}(G \times P \times \mathcal{O}(Fitness))$, where G and P represent the number of generation and the population size, respectively. The $\mathcal{O}(Fitness)$ can be expressed as $\mathcal{O}(Power\_trace) + \mathcal{O}(Thermal\_simulation)$, where the time complexity of the thermal simulation is proportional to the system architecture and the total number of cores.

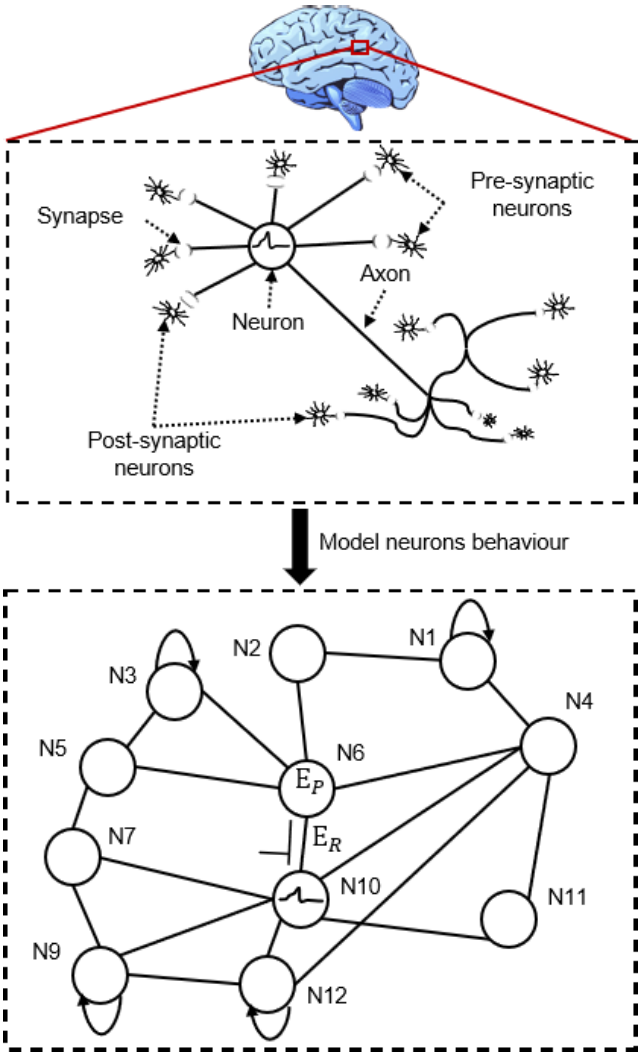

FIGURE 7. Modeling biological spiking neurons behavior into hardware to analyze the energy and temperature generated from each synaptic operation.

V. THERMAL ANALYTICAL MODEL

This section presents the thermal state assessment method by providing an analytical model to analyze the energy consumed and the temperature generated in the 3D neuromorphic system. As shown in Fig. 7, after modeling the SNN, the energy consumed by each synaptic operation can be represented as the energy for communicating the spike and the energy used for processing it. The average energy per synaptic operation ($\overline{E}_{SOP}$) is given by the Eq. (4):

$$\overline{E}_{SOP} = \overline{E}_{Routing} + \overline{E}_{SNPC} \tag{4}$$

where $\overline{E}_{Routing}$ and $\overline{E}_{SNPC}$ are the average energy consumed by the routing of the spike and the processing of the spike, respectively. The $\overline{E}_{Routing}$ is calculated using the Eq. (5):

$$\overline{E}_{Routing} = \overline{H}_{Hops} \times \overline{E}_R \tag{5}$$

With $\overline{H}_{Hops}$ and $\overline{E}_R$ representing the average number of hops in the system and the average energy consumed by a

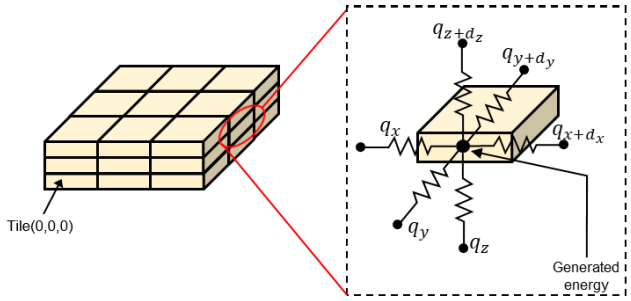

FIGURE 8. The representation of the 3D neuromorphic system tiles and the heat exchange of each tile in the different directions.

router. The analytical model focuses on the temperature of the neuro-processing unit and the heat transfer between the different SNPCs; Therefore, the energy consumed by routing is neglected. Each synaptic operation (SOP) is assumed to generate the system's energy tile-wise, meaning that the power is supposed to be consumed by the given tile's volume. As shown in Fig. 8, let the 3D system be represented as tiles in a 3D grid, where each tile contains a router, a network interface, and an SNPC. A tile's thermal conduction k is assumed to be constant. The governing equation for the heat transfer in a tile is given by Eq. (6):

$$E_{in} - E_{out} + E_{gen} = E_{st} \tag{6}$$

where E_{in}, E_{out}, E_{gen}, and E_{st} are the rates at which thermal energy enters, leaves, is generated, and is stored in the tile, respectively. By applying Fourier's law to Eq. (6) and after simplifications, the heat transfer by conduction in the 3D system can be represented as follows:

$$\frac{\partial T_i(t)}{\partial t} = (\nabla^2 T_i(t) + \frac{q_i(t)}{k}) \times \alpha \tag{7}$$

Where T_i is a tile of the system with coordinates (x, y, z), $\frac{q_i(t)}{k}$ is the power generated per unit volume (T_i) at time step t, α is the thermal diffusivity of the tile, and $\nabla^2 T_i(t)$ is the Laplacian operator of the temperature presented by:

$$\nabla^2 T_i(t) = \frac{\partial^2 T_i(t)}{\partial x^2} + \frac{\partial^2 T_i(t)}{\partial y^2} + \frac{\partial^2 T_i(t)}{\partial z^2} \tag{8}$$

Since we assume that k is a constant, then α is a constant with $\alpha = \frac{k}{\rho \cdot c_p}$, where ρ is the density of the tile and c_p is the specific heat capacity of the tile. From Eq. (7), the system's temperature using different metrics can be calculated. The average temperature of the system is considered to be:

$$\overline{T_s(t)} = \frac{\sum_{i=0}^{X} T_i(t)}{X} \tag{9}$$

where X is the total number of tiles in the system. This analytical model allows a thermal-state evaluation during the design based on the energy consumption of the components and the size of the 3D neuromorphic system. The following section shows the evaluation results of the temperature variations from applying the presented model to the evaluated SNN applications.

TABLE 1. System configuration and used applications for evaluating BTSAM.

Category	NoC Configuration	Applications ID	Total Neurons
Synthetic	6x6x6	S_6_N_{tile}	$216 \times N_{tile}$ [a] $\times 0.9$
	8x8x8	S_8_N_{tile}	$512 \times N_{tile} \times 0.9$
	10x10x10	S_10_N_{tile}	$1000 \times N_{tile} \times 0.9$
Realistic	3x3x3	N_MNIST3	6152 [b]
	4x4x4	N_MNIST4	10250 [c]

[a] N_{tile} is the number of neurons per crossbar. We use 64, 128, 256, 512 for synthetic applications and 256 for real applications
[b] *Feed-forward* topology with three hidden layers, each having 2048 neurons, input 784, and output 10
[c] *Feed-forward* topology with five hidden layers, each having 2048 neurons, input 784, and output 10

TABLE 2. Thermal simulation configuration.

Layer	Height (μm)	Thermal Conductivity (W/μmK)
BEOL	10	2.25×10^{-6}
PCB	10	2.25×10^{-6}
Silicon	50	1.30×10^{-4}
Source	2	1.30×10^{-4}
Parameter		Value
Ambient temperature		300.15 (K)
Heat transfer coefficient of the heat sink		1.3×10^{-9} (W/$\mu m^2 K$)

VI. EVALUATION AND RESULTS

This section presents the proposed mapping method's evaluation methodology and results. The evaluation includes synthetic and real applications in Table 1. The proposed method is evaluated based on temperature and MTTF for both categories. The communication cost and accuracy are assessed for real applications only. The evaluation further includes insights into the effectiveness of the proposed mapping method through analysis and discussion.

A. EVALUATION METHODOLOGY

The proposed mapping method is evaluated on both synthetic and real applications. Table 1 presents a detailed description of these applications. For synthetic applications, neurons were mapped to $6 \times 6 \times 6$, $8 \times 8 \times 8$, and $10 \times 10 \times 10$ NoC-based neuromorphic systems having 64, 128, 256, and 512 neurons per neural tile. Each system configuration has 90% of the total neurons mapped to them. For N-MNIST [38] applications, the neuron activity is extracted from an SNN trained using SnnTorch [39]. Subsequently, these neurons are mapped to $3 \times 3 \times 3$ and $4 \times 4 \times 4$ NoC-based neuromorphic systems having 256 neurons per crossbar.

First, the system's temperature is evaluated using 3D-ICE simulator [40], [41], [42]. The configuration of the simulations is detailed in Table 2. The floorplan of the system is generated based on the thermal model presented in Section V, where each tile's dimensions are 1.151×1.151 mm and each SOP consumes 11.3 pJ [21].

Then, the Mean-Time-To-Failure (MTTF), which is defined by the following equation [17], is then evaluated:

$$MTTF = \frac{1}{A \times J^2} \times e^{(\phi/k \times T)} \tag{10}$$

where A, J, k, ϕ, and T represent constants, the current density, the Boltzmann constant, the activation energy, and the temperature in Kelvin, respectively. Since the experiments are assumed to be conducted on the same hardware design, the constants A and J and ϕ are set to 1 and k is set to 8.617×10^{-5} (eV/K). The MTTF of the different mapping methods is calculated for all the tiles in the system. Then, normalized to the lowest MTTF obtained. In the end, the lowest normalized MTTF for each layer is presented. The accuracy and communication costs are then evaluated. The communication cost is defined by Eq. (11):

$$Communication\_cost = \sum_{n_i, n_j}^{N} distance(n_i, n_j), \forall i \neq j \quad (11)$$

where N is the number of mapped neurons, and *distance* is a function that calculates the required number of hops for each spike to reach from neuron n_i to neuron n_j.

Finally, the hardware complexity of the system with NANGATE 45nm and FreePDK45 is presented, and the area and power are elaborated on.

B. THERMAL-STATE EVALUATION

Fig. 9 and 10 present the results of temperature distribution comparison for linear mappings (XYZ and ZYX), HeterGenMap, and BTSAM of synthetic applications described across various NoC-based neuromorphic system configurations detailed in Table 1. Notably, BTSAM consistently maintains lower temperatures across all layers in nearly all configurations. One exception is observed with a configuration of 10 crossbars, where the linear XYZ mapping results in lower peak temperatures in the last two layers. This anomaly is due to the lack of neuron assignment in the previous layer, which directly affects the reduced temperature of these layers.HeterGenMap achieves slightly lower maximum temperatures than BTSAM in the fourth layer for the S_6_64 application and the sixth and seventh layers for the S_8_64 application. This is because highly active neurons are attributed to other layers, resulting in higher temperatures in these layers than in other mapping methods. Compared to the linear methods, BTSAM achieves a reduction in peak temperature of up to 2.5 K, 5.6 K, 9 K, and 10.7 K for configurations with 64, 128, 256, and 512 neurons per crossbar, respectively, across all layer configurations. The temperature difference, compared to HeterGenMap, is higher, with BTSAM showing lower peak temperatures of up to 12.3 K, 23 K, 45 K, and 58.7 K across all layer configurations with 64, 128, 256, and 512 neurons per crossbar, respectively. The thermal results obtained from the HeterGenMap method are due to the focus on minimizing the communication cost, disregarding power distribution and activity load within the core, resulting in high peak temperatures and an unbalanced thermal state. As the neuron density per crossbar increases, the proposed method exhibits a more pronounced thermal advantage over linear mappings. However, all mapping methods demonstrate a gradual increase in temperature

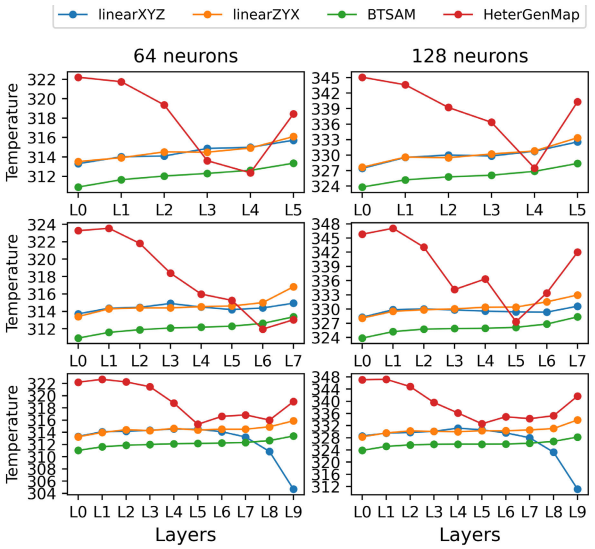

FIGURE 9. Maximum temperature (K) per layer for the linear mapping using both XYZ and ZYX, HeterGenMap, and BTSAM for 64 and 128 neurons per crossbar and different numbers of crossbars per dimension.

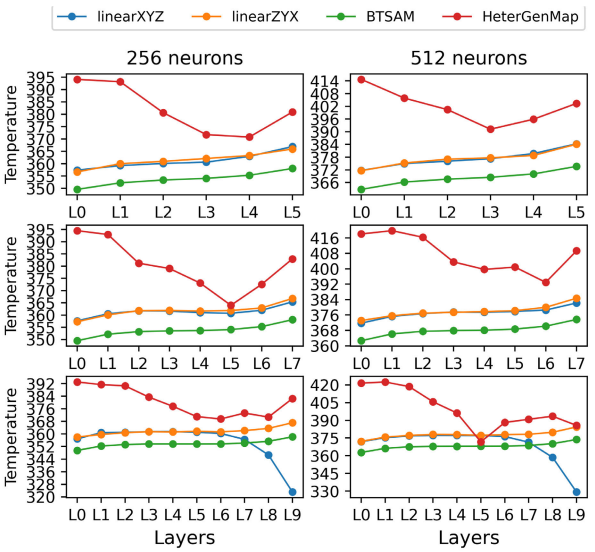

FIGURE 10. Maximum temperature (K) per layer for the linear mapping using both XYZ and ZYX, HeterGenMap, and BTSAM for 256 and 512 neurons per crossbar and different numbers of crossbars per dimension.

escalation from one layer to the next, a phenomenon due to the heterogeneous thermal conductivity between layers. Fig. 11 compares the temperature (maximum, average, and minimum) for $3 \times 3 \times 3$ and $4 \times 4 \times 4$ neuromorphic systems. The proposed method reduces the maximum temperature by 2.6 K, 12.4 K, and 5.2 K compared to linear XYZ, linear ZYX, and HeterGenMap mappings, respectively, in $3 \times 3 \times 3$ neuromorphic system. The reduction is even more significant in the $4 \times 4 \times 4$ neuromorphic system configuration, with the maximum temperature disparities

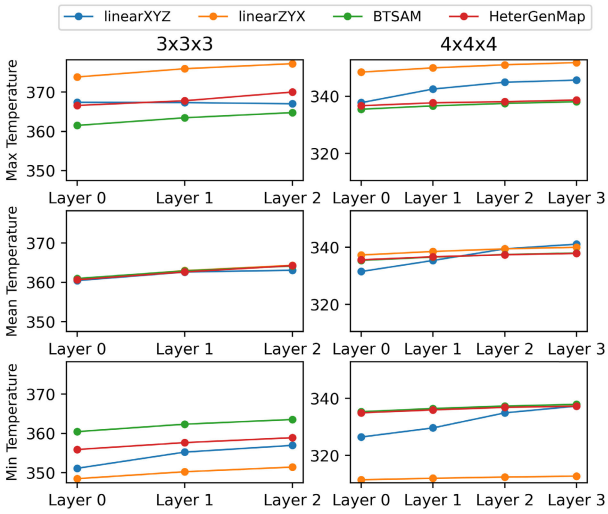

FIGURE 11. Maximum, average, and minimum temperature per layer for different mapping methods for the 3 × 3 × 3 and the 4 × 4 × 4 system configuration.

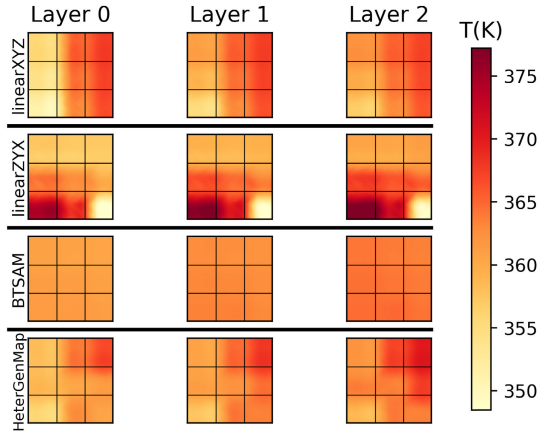

FIGURE 12. Heatmaps obtained from the *N_MNIST* 3 application. Black lines separate the mapping methods. Each heatmap represents a layer.

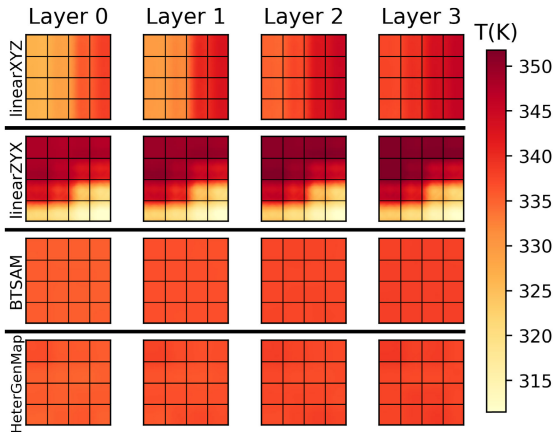

FIGURE 13. Heatmaps obtained from the *N_MNIST* 4 application. Black lines separate the mapping methods. Each heatmap represents a layer.

reaching 7.6 K, 13.7 K, and 0.6 K for linear XYZ, linear ZYX, and HeterGenMap mappings, respectively.

Fig. 12 and 13 show the resulting heatmap from the applications *N_MNIST* 3 and *N_MNIST* 4, respectively, of the mapping methods linear (XYZ and ZYX), HeterGenMap and BTSAM. Fig. 12 and 13 show a high-temperature variance in the system nodes and within each layer for both linear methods. In contrast, HeterGenMap shows a low-temperature variance in the *N_MNIST* 3 application, with a hotspot in the system. For the *N_MNIST* 4 application, HeterGenMap shows a balanced temperature throughout the system. Meanwhile, BTSAM provides a low-temperature variance between nodes within each system layer with a negligible increase in temperature difference between layers for both applications. These results emphasize the effectiveness of the proposed method in lowering the maximum temperature, thereby eliminating hotspots in the system and moderating its thermal state. The following section presents the MTTF results, showing how eliminating hotspots reduces the probability of faults occurring in the system.

C. MEAN-TIME-TO-FAILURE EVALUATION

Fig. 14 and 15 illustrate the Mean-Time-To-Failure (MTTF) across each layer for both linear mappings (XYZ and ZYX), HeterGenMap, and BTSAM across various NoC-based neuromorphic system configurations. BTSAM outperforms the other mapping methods in nearly all neuromorphic system configurations, except in a 10 × 10 × 10, where the linear XYZ mapping has no neuron mapped in the last layer. Notably, BTSAM improves the MTTF by up to 3.2×, 6.9×, 25.9×, and 32.4× for configurations of 64, 128, 256, and 512 neurons per crossbar, respectively, across different layer configurations. Fig. 16 shows the MTTF for each layer in 3 × 3 × 3 and 4 × 4 × 4 NoC-based neuromorphic systems.

In a 3 × 3 × 3 neuromorphic system, the proposed methods significantly improve the life expectancy by 4.5×, 3.7×, and 3.3× in layers 0, 1, and 2, respectively. This improvement is evident in the 4 × 4 × 4 neuromorphic system where the proposed method yields an MTTF 6.1×, 5.3×, 4.8× and 4.5× higher than the lowest durability node across all the system. This improvement underscores BTSAM's efficiency, especially as the neuron density per crossbar increases. It provides a significant thermal advantage and reliability improvement over HeterGenMap and conventional linear mappings.

D. ACCURACY AND COMMUNICATION COST EVALUATION

We estimate the inference accuracy by quantizing the weights and comparing the floating-point precision with an 8-bit fixed-point representation of the weights in neural network configurations with 3 and 5 hidden layers. The transition to a fixed-point representation resulted in a marginal decrease in accuracy, with the 3-layer configuration dropping from 97.66 The accuracy estimation results from quantizing the weights may slightly differ from the hardware simulation

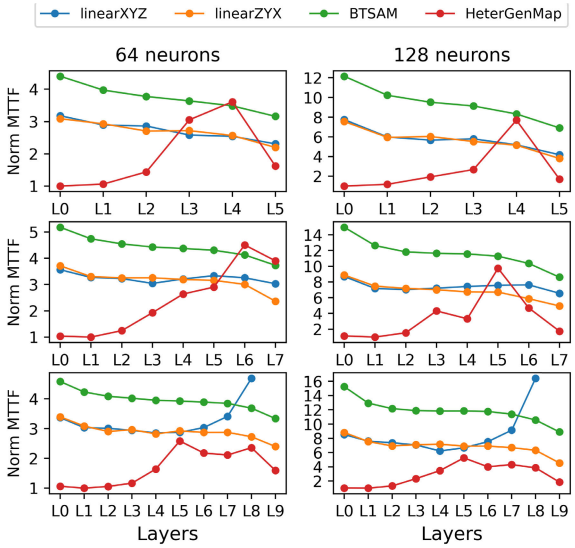

FIGURE 14. Normalized MTTF per layer for linear mappings (XYZ and ZYX), HeterGenMap, and BTSAM, across configurations with 64 and 128 neurons per crossbar and different numbers of crossbars per dimension.

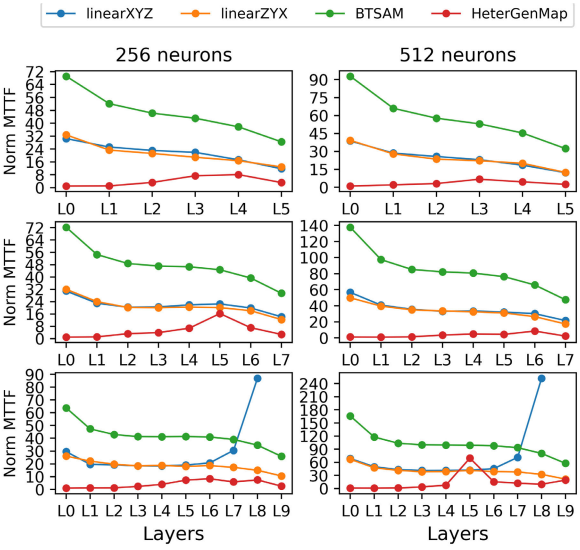

FIGURE 15. Normalized MTTF per layer for linear mappings (XYZ and ZYX), HeterGenMap, and BTSAM, across configurations with 256 and 512 neurons per crossbar and different numbers of crossbars per dimension.

results. The BTSAM communication cost increases by a factor of 3.8 in the 3 × 3 × 3 configuration and by a factor of 7.7 in the 4 × 4 × 4 configuration. This is a direct consequence of the proposed approach, which focuses on an even energy distribution throughout the system, disregarding communication cost minimization.

E. HARDWARE COMPLEXITY EVALUATION

Table 3 presents the hardware complexity for one layer of a 3D-NoC-based NASH neuromorphic system, including the area distribution and power consumption of each module

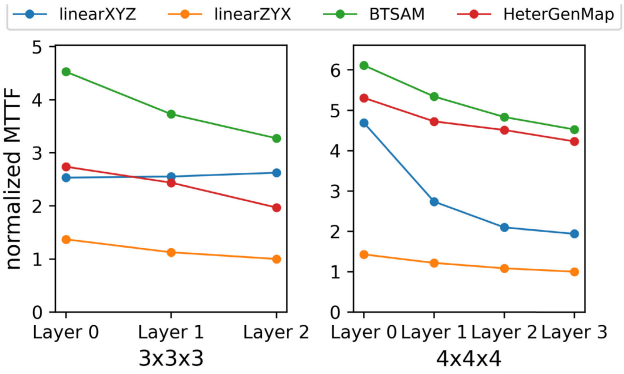

FIGURE 16. Normalized MTTF per layer for different mapping method for the 3 × 3 × 3 and 4 × 4 × 4 NoC-based neuromorphic systems.

TABLE 3. Hardware complexity for one layer (5 × 5) of a 3D-NoC-based neuromorphic system (NASH), including the tile's main components.

Architecture/Module	Power (W)	Area (mm^2)
5x5 (layer)	1.953	23.46
Neurons Cluster	0.059	0.839
Network Interface	0.007	0.066
3D-Router	0.009	0.028

in a tile. The neuron cluster, the NI, and the 3D-Router consist of 63.56%, 5%, and 2.12%, respectively, of the area on a single tile. The power consumption distribution of the tile component is 78.46%, 10.15%, and 11.37% for the neuron cluster, NI, and 3D-Router, respectively. The neuron cluster configuration includes 256 spike inputs in address-event representation (AER) format, 8-bit synapse weights, and 32 physical neurons per cluster. Each crossbar uses a 256-bank 8-bit dual-port SRAM with OpenRAM. Due to Place & Route time constraints, only 32 neurons are integrated per node. The implementation of thermal-aware mapping into the hardware to facilitate future analysis and advanced phases of hardware evaluation has yet to be investigated. However, we expect minimal impact on the area and power consumption overhead.

VII. CONCLUSION AND FUTURE WORKS

This work presents a novel balanced thermal-state-aware mapping (BTSAM) method and a thermal analytical model that emphasizes the uniformity of thermal distribution in 3D-NoC-based neuromorphic systems. The BTSAM consists of three main stages: activity-based scoring, balanced clustering, and thermal-aware mapping leveraging genetic algorithm. This method significantly lowers the maximum temperatures in the system, as shown by a reduction of up to 13.7 K and 10.7 K compared to linear methods and up to 5.2 K and 58.7 K compared to HeterGenMap in real and synthetic applications, respectively. This temperature reduction and increased reliability are achieved with an acceptable trade-off regarding communication cost and overall system accuracy. The proposed approach addresses the inherent thermal issues in 3D-NoC-based neuromorphic systems characterized

by high power density and heterogeneous heat transfer. By strategically mapping spiking neurons to crossbars, the method ensures a more balanced temperature distribution across the system, enhancing reliability and reducing the likelihood of thermal hotspots. The results of our study demonstrate a practical solution for thermal management in reliable 3D neuromorphic systems and contribute to the broader field of hardware design for brain-inspired computing systems.

Future work will focus on a fault-tolerant and thermal-state-aware mapping method to avoid creating new hotspots and thus improve the system's reliability and robustness. Moreover, the hardware implementation with a detailed power and area consumption analysis will also be investigated.

REFERENCES

[1] A. Katsiamis and E. Drakakis, "Analogue CMOS cochlea systems: A historic retrospective," *Biomimetic Based Appl.*, vol. 26, pp. 429–472, Apr. 2011.

[2] C. Mead, "Neuromorphic electronic systems," *Proc. IEEE*, vol. 78, no. 10, pp. 1629–1636, Oct. 1990.

[3] W. Maass, "Networks of spiking neurons: The third generation of neural network models," *Neural Netw.*, vol. 10, no. 9, pp. 1659–1671, Dec. 1997.

[4] L. Deng, Y. Wu, X. Hu, L. Liang, Y. Ding, G. Li, G. Zhao, P. Li, and Y. Xie, "Rethinking the performance comparison between SNNS and ANNS," *Neural Netw.*, vol. 121, pp. 294–307, Jan. 2020, doi: 10.1016/j.neunet.2019.09.005.

[5] P. Falez, "Improving spiking neural networks trained with spike timing dependent plasticity for image recognition," M.S. theses, Univ. de Lille, Lille, France, Oct. 2019. [Online]. Available: https://hal.science/tel-02429539

[6] M. Davies, N. Srinivasa, T.-H. Lin, G. Chinya, Y. Cao, S. H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain, Y. Liao, C.-K. Lin, A. Lines, R. Liu, D. Mathaikutty, S. McCoy, A. Paul, J. Tse, G. Venkataramanan, Y.-H. Weng, A. Wild, Y. Yang, and H. Wang, "Loihi: A neuromorphic manycore processor with on-chip learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, Jan. 2018.

[7] G. Orchard, E. P. Frady, D. B. D. Rubin, S. Sanborn, S. B. Shrestha, F. T. Sommer, and M. Davies, "Efficient neuromorphic signal processing with Loihi 2," 2021, *arXiv:2111.03746*.

[8] F. Akopyan, J. Sawada, A. Cassidy, R. Alvarez-Icaza, J. Arthur, P. Merolla, N. Imam, Y. Nakamura, P. Datta, G.-J. Nam, B. Taba, M. Beakes, B. Brezzo, J. B. Kuang, R. Manohar, W. P. Risk, B. Jackson, and D. S. Modha, "TrueNorth: Design and tool flow of a 65 mW 1 million neuron programmable neurosynaptic chip," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 34, no. 10, pp. 1537–1557, Oct. 2015.

[9] C. Pehle, S. Billaudelle, B. Cramer, J. Kaiser, K. Schreiber, Y. Stradmann, J. Weis, A. Leibfried, E. Müller, and J. Schemmel, "The BrainScaleS-2 accelerated neuromorphic system with hybrid plasticity," *Frontiers Neurosci.*, vol. 16, Feb. 2022, Art. no. 795876.

[10] S. Moradi, N. Qiao, F. Stefanini, and G. Indiveri, "A scalable multicore architecture with heterogeneous memory structures for dynamic neuromorphic asynchronous processors (DYNAPs)," *IEEE Trans. Biomed. Circuits Syst.*, vol. 12, no. 1, pp. 106–122, Feb. 2018.

[11] A. Balaji, A. Das, Y. Wu, K. Huynh, F. G. Dell'Anna, G. Indiveri, J. L. Krichmar, N. D. Dutt, S. Schaafsma, and F. Catthoor, "Mapping spiking neural networks to neuromorphic hardware," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 28, no. 1, pp. 76–86, Jan. 2020.

[12] T. H. Vu, O. M. Ikechukwu, and A. Ben Abdallah, "Fault-tolerant spike routing algorithm and architecture for three dimensional NoC-based neuromorphic systems," *IEEE Access*, vol. 7, pp. 90436–90452, 2019.

[13] B. Li, X. Wang, A. K. Singh, and T. Mak, "On runtime communication- and thermal-aware application mapping in 3D NoC," in *Proc. 11th IEEE/ACM Int. Symp. Networks-Chip (NOCS)*. New York, NY, USA: Association for Computing Machinery, Oct. 2017, pp. 1–8, doi: 10.1145/3130218.3130228.

[14] B. Black, D. Pantuso, P. Reed, J. Rupley, S. Shankar, J. Shen, C. Webb, M. Annavaram, N. Brekelbaum, J. DeVale, L. Jiang, G. H. Loh, D. McCaule, P. Morrow, and D. W. Nelson, "Die stacking (3D) microarchitecture," in *Proc. 39th Annu. IEEE/ACM Int. Symp. Microarchitecture (MICRO)*, Dec. 2006, pp. 469–479.

[15] K. J. Chen, C.-H. Chao, and A. A. Wu, "Thermal-aware 3D network-on-chip (3D NoC) designs: Routing algorithms and thermal managements," *IEEE Circuits Syst. Mag.*, vol. 15, no. 4, pp. 45–69, 4th Quart., 2015.

[16] C.-H. Chao, K.-C. Chen, T.-C. Yin, S.-Y. Lin, and A.-Y. Wu, "Transport-layer-assisted routing for runtime thermal management of 3D NoC systems," *ACM Trans. Embedded Comput. Syst.*, vol. 13, no. 1, pp. 1–22, Aug. 2013.

[17] J. R. Black, "Mass transport of aluminum by momentum exchange with conducting electrons," in *Proc. 6th Annu. Rel. Phys. Symp. (IEEE)*, Nov. 1967, pp. 148–159.

[18] K. N. Dang, A. B. Ahmed, A. B. Abdallah, and X.-T. Tran, "Hotcluster: A thermal-aware defect recovery method for through-silicon-vias toward reliable 3-D ICs systems," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 41, no. 4, pp. 799–812, Apr. 2022.

[19] A. Das, Y. Wu, K. Huynh, F. Dell'Anna, F. Catthoor, and S. Schaafsma, "Mapping of local and global synapses on spiking neuromorphic hardware," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Mar. 2018, pp. 1217–1222.

[20] X. Jin, "Parallel simulation of neural networks on spinnaker universal neuromorphic hardware," Ph.D. dissertation, Univ. Manchester, Manchester, U.K., 2010.

[21] O. M. Ikechukwu, K. N. Dang, and A. B. Abdallah, "On the design of a fault-tolerant scalable three dimensional NoC-based digital neuromorphic system with on-chip learning," *IEEE Access*, vol. 9, pp. 64331–64345, 2021.

[22] K. N. Dang, N. A. V. Doan, N.-D. Nguyen, and A. B. Abdallah, "HeterGen-Map: An evolutionary mapping framework for heterogeneous NoC-based neuromorphic systems," *IEEE Access*, vol. 11, pp. 144095–144112, 2023.

[23] T. C. Koopmans and M. Beckmann, "Assignment problems and the location of economic activities," *Econometrica*, vol. 25, no. 1, pp. 53–76, Jan. 1957. [Online]. Available: http://www.jstor.org/stable/1907742

[24] J. H. Holland, *Adaptation in Natural and Artificial Systems*, 1975, pp. 390–401.

[25] J. Kennedy and R. Eberhart, "Particle swarm optimization (PSO)," in *Proc. IEEE Int. Conf. Neural Netw.*, Dec. 1995, vol. 4, no. 1, pp. 1942–1948.

[26] Y. Ji, Y. Zhang, S. Li, P. Chi, C. Jiang, P. Qu, Y. Xie, and W. Chen, "NEUTRAMS: Neural network transformation and co-design under neuromorphic hardware constraints," in *Proc. 49th Annu. IEEE/ACM Int. Symp. Microarchitecture (MICRO)*, Oct. 2016, pp. 1–13.

[27] A. Ben Abdallah and K. N. Dang, "Toward robust cognitive 3D brain-inspired cross-paradigm system," *Frontiers Neurosci.*, vol. 15, Jun. 2021, Art. no. 690208, doi: 10.3389/fnins.2021.690208.

[28] K. N. Dang, N. A. V. Doan, and A. B. Abdallah, "MigSpike: A migration based algorithms and architecture for scalable robust neuromorphic systems," *IEEE Trans. Emerg. Topics Comput.*, vol. 10, no. 2, pp. 602–617, Apr. 2022.

[29] F. K. Tiangsheng Zhang and A. K. Coskun, "Thermal modeling and management for 3D stacked systems," in *Physical Design for 3D Integrated Circuits*. Boca Raton, FL, USA: CRC Press, 2017, ch. 10, pp. 229–244.

[30] P. K. Hamedani, S. Hessabi, H. Sarbazi-Azad, and N. E. Jerger, "Exploration of temperature constraints for thermal aware mapping of 3D networks on chip," in *Proc. 20th Euromicro Int. Conf. Parallel, Distrib. Network-Based Process.*, Feb. 2012, pp. 499–506.

[31] M. V. Beigi and G. Memik, "Thermal-aware optimizations of ReRAM-based neuromorphic computing systems," in *Proc. 55th ACM/ESDA/IEEE Design Autom. Conf. (DAC)*, Jun. 2018, pp. 1–6.

[32] C. Zhang, Y. Ma, and P. Zhou, "Thermal-aware layout optimization and mapping methods for resistive neuromorphic engines," in *Proc. 27th Asia South Pacific Design Autom. Conf. (ASP-DAC)*, Jan. 2022, pp. 50–55.

[33] M. Maatar, K. N. Dang, and A. B. Abdallah, "Thermal-aware task-mapping method for 3D-NoC-based neuromorphic systems," in *Proc. 6th Int. Conf. Electron. Technol. (ICET)*, May 2023, pp. 1067–1076.

[34] M. R. Azghadi, N. Iannella, S. F. Al-Sarawi, G. Indiveri, and D. Abbott, "Spike-based synaptic plasticity in silicon: Design, implementation, application, and challenges," *Proc. IEEE*, vol. 102, no. 5, pp. 717–737, May 2014.

[35] T. H. Vu, Y. Okuyama, and A. B. Abdallah, "Comprehensive analytic performance assessment and K-means based multicast routing algorithm and architecture for 3D-NoC of spiking neurons," *ACM J. Emerg. Technol. Comput. Syst.*, vol. 15, no. 4, pp. 1–28, Oct. 2019, doi: 10.1145/3340963.
[36] L. Davis, "Applying adaptive algorithms to epistatic domains," in *Proc. IJCAI*, vol. 85, 1985, pp. 162–164.
[37] D. B. Fogel, "A parallel processing approach to a multiple travelling salesman problem using evolutionary programming," in *Proc. 4th Annu. Symp. Parallel Process.*, Fullerton, CA, USA, 1990, pp. 318–326.
[38] G. Orchard, A. Jayawant, G. K. Cohen, and N. Thakor, "Converting static image datasets to spiking neuromorphic datasets using saccades," *Frontiers Neurosci.*, vol. 9, p. 437, Nov. 2015.
[39] J. K. Eshraghian, M. Ward, E. Neftci, X. Wang, G. Lenz, G. Dwivedi, M. Bennamoun, D. S. Jeong, and W. D. Lu, "Training spiking neural networks using lessons from deep learning," *Proc. IEEE*, 2023.
[40] A. Sridhar, A. Vincenzi, D. Atienza, and T. Brunschwiler, "3D-ICE: A compact thermal model for early-stage design of liquid-cooled ICs," *IEEE Trans. Comput.*, vol. 63, no. 10, pp. 2576–2589, Oct. 2014.
[41] A. Sridhar, A. Vincenzi, M. Ruggiero, T. Brunschwiler, and D. Atienza, "3D-ICE: Fast compact transient thermal modeling for 3D ICs with inter-tier liquid cooling," in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design (ICCAD)*, Nov. 2010, pp. 463–470.
[42] F. Terraneo, A. Leva, W. Fornaciari, M. Zapater, and D. Atienza, "3D-ICE 3.0: Efficient nonlinear MPSoC thermal simulation with pluggable heat sink models," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 41, no. 4, pp. 1062–1075, Apr. 2022.

MOHAMED MAATAR (Student Member, IEEE) received the Engineering degree in computer science from the ESPRIT School of Engineering, Tunisia, in 2018, and the M.Sc. degree in data science and big data from CY-TECH, France, in 2018. He is currently pursuing the Ph.D. degree in thermal awareness in neuromorphic systems with The University of Aizu. His research interests include operational research, machine learning, spiking neural networks, and neuromorphic systems.

ZHISHANG WANG (Member, IEEE) received the B.S. degree in computer science from Wuhan University, China, in 2014, the M.S. degree in computer science from the University of Freiburg, Germany, in 2019, and the Ph.D. degree in computer science from The University of Aizu, Japan, in 2023. He is currently a Postdoctoral Researcher with the Division of Computer Engineering, Department of Computer Science and Engineering, The University of Aizu. His current research interests include machine learning systems, collaborative learning, blockchain, and trustworthy AI. He is also interested in event-driven neuromorphic systems targeted for a new generation of brain-inspired computing technologies and adaptive edge computing systems.

KHANH N. DANG (Member, IEEE) received the M.Sc. degree from The University of Aizu, in 2014, and the Ph.D. degree from Paris-Sud University, France, in 2017. He is currently an Associate Professor with the Division of Computer Engineering, Department of Computer Science and Engineering, The University of Aizu. Before joining The University of Aizu as a Faculty Member, and an Assistant Professor with Vietnam National University Hanoi, Vietnam. His research interests include VLSI design, thermal awareness, fault tolerance, and neuromorphic computing.

ABDERAZEK BEN ABDALLAH (Senior Member, IEEE) received the Ph.D. degree in computer engineering from The University of Electro-Communications, Tokyo, in 2002. From April 2014 to March 2022, he was the Head of the Computer Engineering Division, School of Computer Science and Engineering, The University of Aizu, Japan. Since April 2022, he has been the Dean of the School of Computer Science and Engineering, The University of Aizu. He is currently a Full Professor with The University of Aizu. He is the author of four books, 14 patents, and more than 150 publications in peer-reviewed journal articles and conference papers. His research interests include adaptive/self-organizing systems, brain-inspired computing, interconnection networks, and AI-powered cyber-physical systems. He is a Senior Member of ACM.