Circuits and Systems
Mekelweg 4,
2628 CD Delft
The Netherlands
http://ens.ewi.tudelft.nl/

CAS-2022-03

M.Sc. Thesis

Modeling of router structure for SNN-applicable NoC definitions

Yongkang Zhou B.Sc.

Abstract

Spiking neural networks (SNN), as the third-generation artificial neural network, has a similar potential pulse triggering mechanism to the biological neuron. This mechanism enables the spiking neural network to increase computing power compared to the traditional artificial neural network to process complex information. However, a large number of interconnection resources is required. This requirement is highly consistent with the characteristics of the network on chip (NoC). This thesis is aimed at developing a scalable cycle-accurate simulator based on Noxim, which provides a configurable NoC that can simulate neuron-to-neuron communication for delivering spiking traffic. This simulator achieves several configurable metrics including topology and routing schemes, network size, the number of channels, and neuron mapping methods. This thesis then evaluates the effects of these metrics on performance for two kinds of traffic patterns. To take power consumption and area into account, this thesis also provides an approximate estimate of area and power consumption for trade-offs in the early-design stage.

Faculty of Electrical Engineering, Mathematics and Computer Science

Delft University of Technology

Modeling of router structure for SNN-applicable NoC definitions

Thesis

submitted in partial fulfillment of the
requirements for the degree of

Master of Science

in

Electrical Engineering

by

Yongkang Zhou B.Sc.
born in Xiangyang, China

This work was performed in:

Circuits and Systems Group
Department of Microelectronics & Computer Engineering
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology

Delft University of Technology

Delft University of Technology
Department of
Microelectronics & Computer Engineering

The undersigned hereby certify that they have read and recommend to the Faculty of Electrical Engineering, Mathematics and Computer Science for acceptance a thesis entitled **"Modeling of router structure for SNN-applicable NoC definitions "** by **Yongkang Zhou B.Sc.** in partial fulfillment of the requirements for the degree of **Master of Science**.

Dated: 08.03.2022

Chairman: ________________________
Prof.dr.ir. Rene van Leuken

Advisor: ________________________
Prof.dr.ir. Rene van Leuken

Committee Members: ________________________
Dr.ir. Amir Zjajo

____
Dr.ir. Arjan van Genderen

Abstract

Spiking neural networks (SNN), as the third-generation artificial neural network, has a similar potential pulse triggering mechanism to the biological neuron. This mechanism enables the spiking neural network to increase computing power compared to the traditional artificial neural network to process complex information. However, a large number of interconnection resources is required. This requirement is highly consistent with the characteristics of the network on chip (NoC). This thesis is aimed at developing a scalable cycle-accurate simulator based on Noxim, which provides a configurable NoC that can simulate neuron-to-neuron communication for delivering spiking traffic. This simulator achieves several configurable metrics including topology and routing schemes, network size, the number of channels, and neuron mapping methods. This thesis then evaluates the effects of these metrics on performance for two kinds of traffic patterns. To take power consumption and area into account, this thesis also provides an approximate estimate of area and power consumption for trade-offs in the early-design stage.

Acknowledgments

First, I would like to thank my supervisor, Prof.dr.ir. Rene van Leuken and Dr.ir. Amir Zjajo provide this opportunity to work on this thesis project, give me a lot of useful advice on my project. Their rigorous attitude towards science and knowledge will affect my future life.

I would like to thank my company supervisors, Charles Lo, Shashanka Marigi Rajanarayana and Alexander de Graaf. Without their help and weekly meeting, I cannot complete this project. I appreciate their kindness and patience.

I would like to thank all my colleagues at Innatera. Though we have to meet online at most time due to COVID-19, they create a pleasant work environment and makes me feel like work in a big family.

I would further like to thank my girlfriend Ziwei Xia and my family. Their support gives me confidence to study for the master degree at TU Delft. Their encouragement gives me great strength when I meet troubles.

Yongkang Zhou B.Sc.
Delft, The Netherlands
08.03.2022

Contents

List of Figures

List of Tables

Introduction 1

Neural network is considered the main driving force of the current development of artificial intelligence, and it has been a popular field for a long time. In 1958, Rosenblatt[7] proposed ”perceptron”, a machine that can simulate human perception, and completed the simulation of the perceptron on the IBM704. In 1960 he realized a neural computer capable of recognizing some English letters based on the perceptron[8]. In the mid-1980s, multi-layer artificial neural networks(ANN) emerged and developed deep convolutional networks[9]. However, since ANN lacks the internal dynamic mechanism of nerves, it is biologically inaccurate and cannot mimic the mechanism of biological brain neurons accurately. Recently, the spiking neural network(SNN) is known as a new generation of neural networks and has received more and more attention.

1.1 Motivation

In the recent years, Spiking Neural Networks(SNNs), which originate from computational neuroscience, have been researched deeply in the field of neuromorphic engineering and brain-like inspired computing due to their rich spatiotemporal dynamic characteristics, diverse coding mechanisms, and hardware-compatible event-driven characteristics.

Compared with traditional neural networks, spiking neural networks are more like the results of bionics research than those based on mathematical research. From the neuron model to the network structure, it is based on the biological neural network, and it has a lot of advantages that are suitable for hardware implementation, such as the working mode of threshold used in the spiking neuron model. Research on how to use hardware to achieve efficient computation of spiking neural networks has a long history, resulting in multiple classic designs. Early researches include designing and implementing several basic units with neuron working functions in FPGA, and then realizing the calculation of spiking neural network through the interconnection between units[10]. However, in order to simulate the complete function of neurons, the hardware resource overhead of a unit neuron is too large, coupled with the resource overhead of implementing large-scale inter-neuron interconnections on-chip, this design cannot implement a large-scale spiking neural network in a single chip. Later research on the use of multi-chip interconnects to achieve larger-scale network designs[11] offered additional solutions.

In a multi-core SNN architecture, spikes are generated from one core (usually an analog array and encoder/decoder, etc.) to another. In general, the network should be as transparent as possible in that a spike produced by a neuron should immediately appear at any target synapse. However, this is impractical at scale. Instead, spikes must be encoded and transmitted across a chip. The SNN-based NoC is required to provide high traffic capacity because each neuron in the network has a large fan-out, and neurons are usually divided into groups when implementing them into hardware. Thus, the synapse among neurons can be divided into local synapses, which connect neurons in the same group, and global synapses which connect different neuron groups. In this project, we mainly focus on global communication and explore different metrics of networks that can affect the performance and can be implemented to hardware.

1.2 Objective and approach

The objective of this work is to implement a network evaluation framework. This framework is used to determine what sort of interconnection network would work well for input spiking traffic and can easily quantify and compare different spike routing architectures in terms of latency, energy consumption, and area. The approach to achieve the goal is as follows:

- **Implement an interconnect network for SNN.**

  The SNN-based chip has a multi-core feature. This feature required an interconnect network instead of traditional bus architecture in the chip design. In this case, the evaluation framework is required to integrate interconnect networks with different topologies.

- **Mapping neurons to network.**

  In spiking neural networks, neurons are connected (usually in feed-forward connections) and spikes are transmitted from input to output. A significant problem is to design a proper unit in hardware to place neurons. Also, different mapping methods bring about different performances.

- **Implement latency performance evaluation framework**

  In this project, the evaluation framework should have the ability to measure the cycle-accurate latency of every single spike. The values of global latency including average latency and maximum latency are required to be obtained for analysis.

- **Power and Area estimation**

  The design of NoC for SNNs is required to consider trade-offs among latency, energy consumption, and area. Besides latency calculation in the network, a feasible method to estimate power consumption and area is needed to make the framework more comprehensive.

1.3 Contributions

The mainly contributions of this project follows:

- Three types of topologies, including ring, multi-path ring, and mesh for SNN application as well as corresponding routing schemes are implemented.

- Four neuron-to-cluster mapping methods are developed.
- The configurable evaluation framework for SNN networks is implemented. This framework works in synchronous and can inject spikes based on the external spike raster.
- The development of an approximate framework to estimate energy consumption and area in a network. The power and area parameter is unknown since no hardware implementation is completed. These parameters can still be estimated and be approximate to the real result.

1.4 Outline

This thesis is organized as follows: Chapter 2 presents some previous research that relates to this work, basic concepts, and state-of-the-art research. The architecture of the core of SNN used to accommodate neurons and send/receive spikes and interconnections are described in Chapter 3. Chapter 4 focuses on the detailed design method, such as the architecture of the network, the microarchitecture of the components of NoC, and the neuron mapping algorithms. Chapter 5 provides the simulation results (latency) of different topologies with different variable settings. Chapter 6 gives the conclusion of the thesis and lists some interesting directions that can be explored in future work.

Part I

Background

Neuromorphic computing background and state of the art 2

2.1 The neuron

The neuron, or we can call it a nerve cell, is organized by a soma (the cell body), dendrites, and a single axon. The neuron is an electrically excitable cell that transmits information with other same cells by partly electrical and partly chemical signals. The interconnect between these two cells are called synapses and they are unidirectional. Generally, each neuron has 1,000 to 10,000 synaptic connections to other neurons[12]. In the communication process, the neuron is electrically excitable and its membranes need to maintain the voltage gradient. If the voltage changes sufficiently in a short period of time, the neuron will produce an all-or-nothing electrochemical pulse called an action potential. The action potential, usually seen as an all-or-nothing electrochemical pulse, will be produced by neurons. This potential will be transmitted along the axon and activates the destinate synapses when reaching the destination.

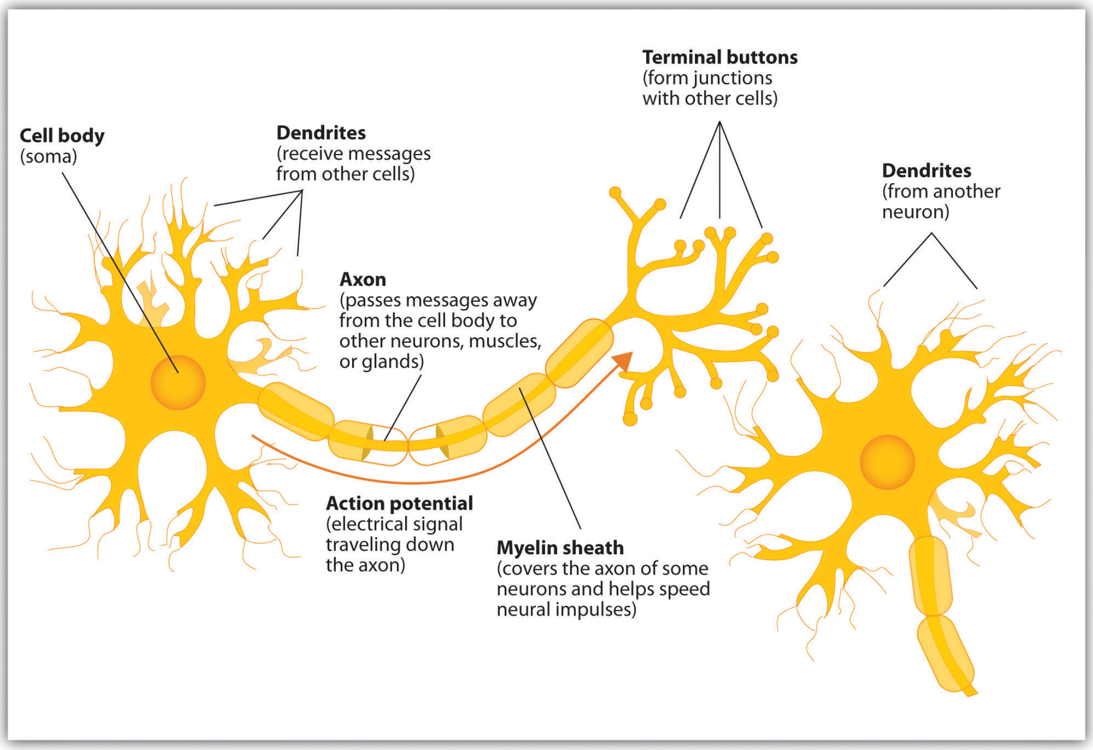


Figure 2.1: Main Components and structures of a biological neuron[1]

2.2 Spiking neuron models

In view of the dynamic characteristics of the potential of neurons at work, neurophysiologists have implemented a lot of models. They are the basic units of the spiking neural network and determine the basic dynamic characteristics of the network. Influential models include Hodgkin-Huxley Model, LIF (leaky integrate-and-fire) model, and ANN model, which are introduced below.

2.2.1 Hodgkin-Huxley model

In 1952, Hodgkin and Huxley[13] measured the action potential of giant squid axons, and proposed a theoretical mathematical model (HH model) of the electrical activity mechanism of neurons and corresponding circuit simulations, laying the foundation for the generation and propagation mechanism of action potentials.

$$I = C_M \frac{dV}{dt} + \overline{g}_K n^4 (V - V_K) + \overline{g}_1 n^4 (V - V_l) + \overline{g}_{N_a} m^3 h (V - V_N a), \tag{2.1}$$

$$\frac{dm}{dt} = \alpha_n(V)(1-n) - \beta_n(V)n, \tag{2.2}$$

$$\frac{dm}{dt} = \alpha_m(V)(1-m) - \beta_m(V)m, \tag{2.3}$$

$$\frac{dh}{dt} = \alpha_h(1-h) - \beta_h(V)h; \tag{2.4}$$

where I is membrane current density; C_M is the membrane capacitance per unit area; V represents membrane potential; $g_K = \overline{g}_K n^4$, $g_{N_a} = \overline{g}_{N_a} m^3 h$, and $\overline{g}_l$ are the conductivity densities of potassium ions, sodium ions and other ions, respectively; V_K , V_{N_a}, and V_l represent the reversal potentials of several ion channels; n, m, and h are assumed to be certain particle concentrations related to ion transport, and their corresponding α and β symbolize the movement of the particles into or outside the membrane. Rate. In the HH model, the generation mechanism between the Na^+, K^+ plasma channels, and the action potential is clearly modeled, and the potential change curve recorded by the biological nerve tissue experiment is accurately approximated.

2.2.2 LIF model

As early as 1907, Lapicque proposed the Integrate-and-fire (I&F) model[14]. The process of action potential is: "If the membrane potential is higher than the threshold V_{th}, the neuron will excite the pulse, then the membrane potential will fall back to the reset potential V_{reset}". The model describes the change law of the sub-threshold potential, of which the simplest and most common is the LIF model[15]:

$$\tau_m \frac{dV}{dt} = V_{reset} - V + R_m I \tag{2.5}$$

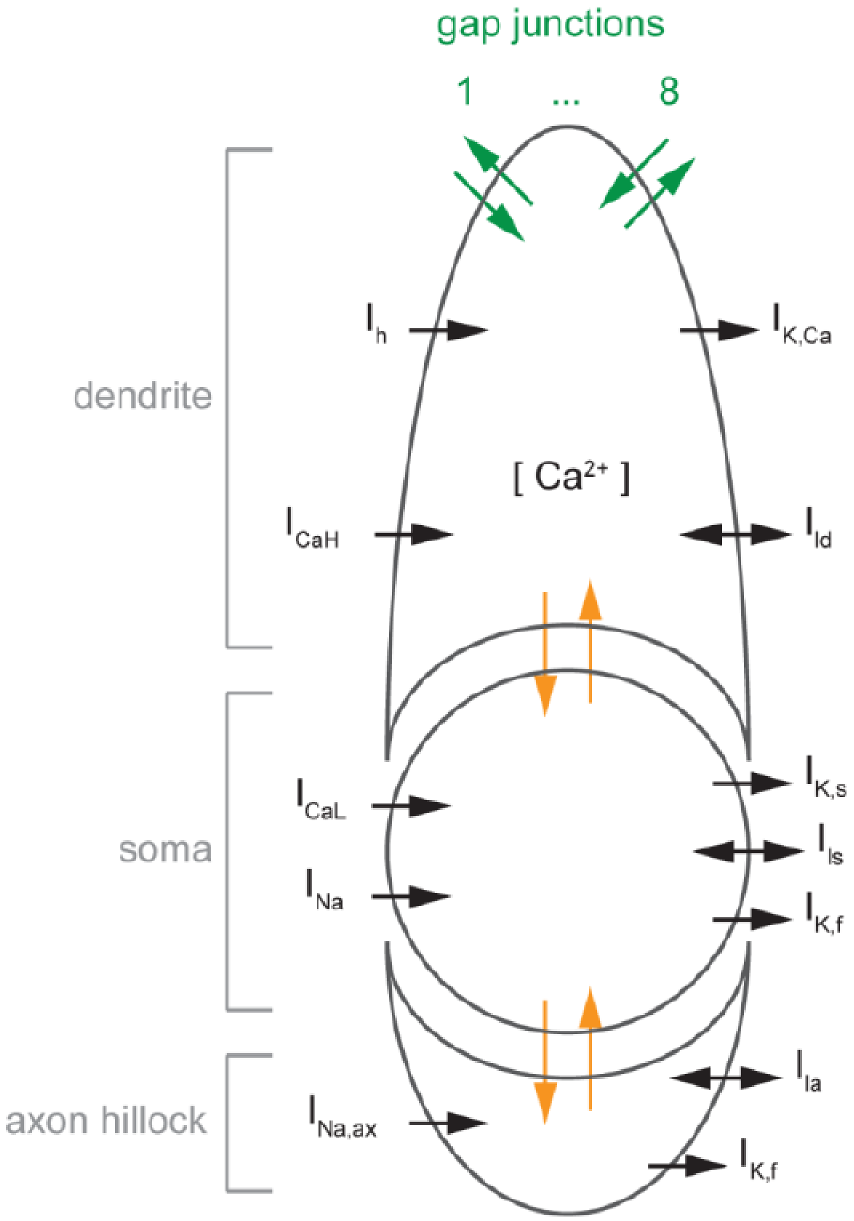


Figure 2.2: Schematic representation of Hodgkin Huxley neuron model[2]

where τ_m represents the membrane time constant, V_{rest} represents the resting potential, and R_m and I represent the impedance and input current of the cell membrane, respectively.

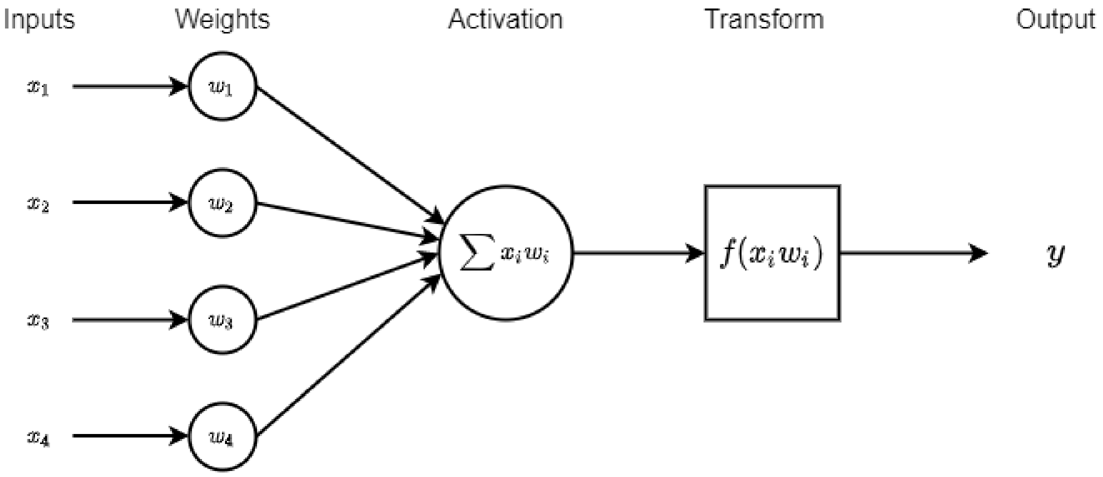


Figure 2.3: ANN neuron model

2.2.3 ANN model

ANN neuron model[16], we also briefly discuss the basic neuron structure in artificial neural networks for comparison. The ANN neuron model retains the information processing function of biological neurons with multiple inputs and single outputs, but its threshold characteristics and the action potential machine makes a further abstract simplification, and its modeling is as follows:

$$y_i^l = \phi(\sum_{j=0}^{n^{l-1}} w_{ij}^{l-1} x_j^{l-1}) \tag{2.6}$$

Among them, the output value y_i^l of the i-th neuron located in the l layer is calculated by the nonlinear activation function $\phi(\cdot)$ after the weighted summation of the output x_j^{l-1} of the n^{l-1} neurons in the previous layer. Compared with SNN, ANN Neurons use high-precision continuous activation function values instead of discrete pulse sequences for communication, abandoning operations in the time domain and only retaining the spatial domain structure of layer-by-layer calculation. Although the expression accuracy of SNN is lower, it retains more abundant Neuron dynamics, in addition to receiving input in the spatial domain, the current state is also naturally affected by past historical moments. Therefore, SNN may have stronger spatiotemporal data processing potential. In addition, due to the existence of threshold characteristics, SNN's impulse signal is usually sparse, and the calculation is event-driven (executed only when the pulse arrives). Combining the 0/1 pulse signal expression form can avoid the high cost of multiplication calculation in ANN, showing the characteristics of ultra-low power consumption, which also contributes to the birth of many neuromorphic hardware.

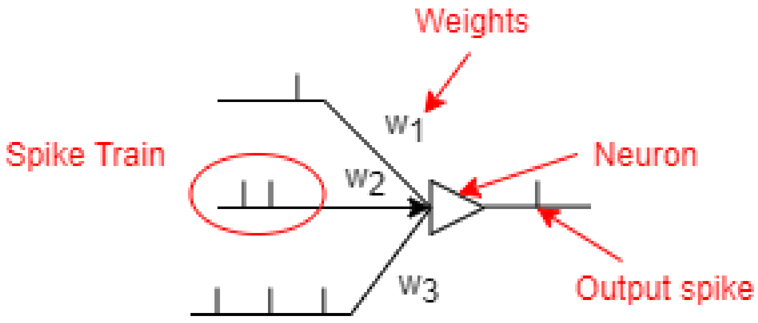


Figure 2.4: Spiking neuron model

2.3 Noxim

Several open-source simulator, such as Noxim, Booksim and Atlas exists. The base structure of **Noxim**[17] is used and a novel simulator in this project is built upon Noxim. Noxim is an open-source software simulator used for network simulation. This simulator can provide cycle-accurate simulation and allows users to collect different evaluation results, such as the number of packets, traffic in the network, latency and power consumption.

2.4 Neural communication schemes - Feed Forward Network

The neural communication schemes represents how synapses work as interconnections among such a large number of neurons. *Feed forward* neural network[18], as shown in Figure 2.5, is most common neural networks. In general, the feed-forward network contains three parts. The first and last layers are called input and output layer, respectively, and rest layers in the middle are hidden layers. All signals are sent from input to output layer through hidden layers in one direction.

2.5 Address Event Representation

The frequency of events in biological neurons is about 10 kHz - 20 kHz. This frequency is tolerable compared to modern digital processing systems. However, each neuron in the system works independently and has the characteristics such as asynchronous and self-timed. If the events generated by the neuron are processed separately, the system is prone to event conflicts and cause information loss, and the structure of the information transmission network between neurons will also become large and complex, hence difficult to implement it in a circuit. Therefore, it is necessary to design a neuron encoding circuit to record the time and address of each event. By encoding the event and multiplexing it on a bus, it is output in order according to the time sequence of the event, without the need for separate processing of the event. Thus, it can solve the communication transmission and conflict problems when the neurons in the system generate events. This module that multiplexes events onto the bus for processing is called AER encoding circuit[19][20][21]. The operating mechanism of AER in the neural

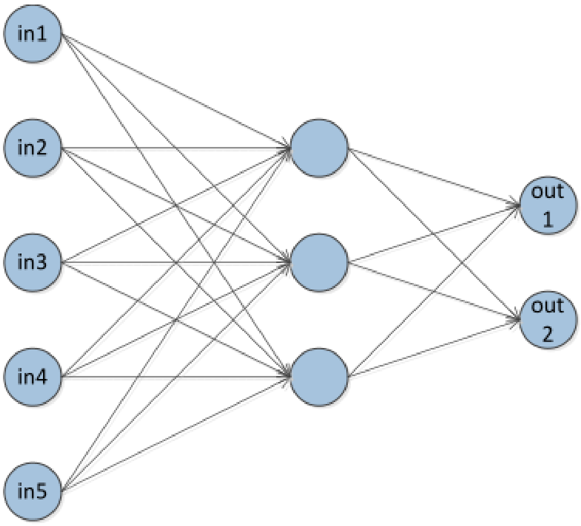


Figure 2.5: Feed forward communication pattern

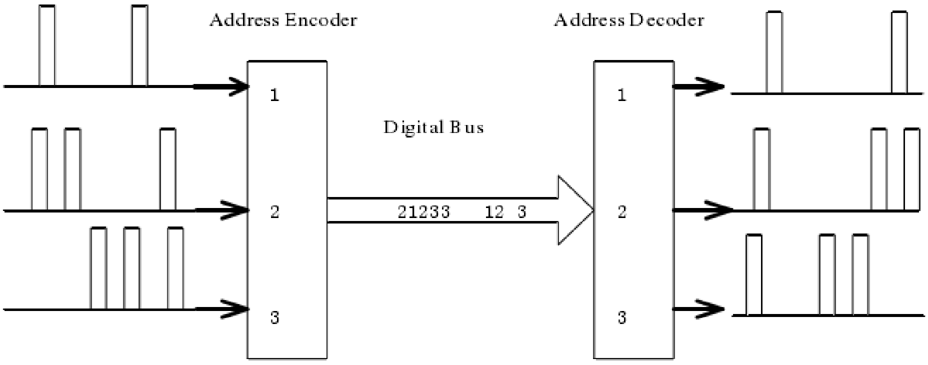


Figure 2.6: One-dimensional AER encoding

network is: when the impulse neuron in the network has an event output, it only needs to output the neuron's address and other attributes in the way of impulsive information transmission, and then restore the event through the processing unit. Figure 2.6 shows a schematic diagram of AER encoding. Each row represents the pulse event in the network. The row address encoding information of the event is output through the encoder, and then the address and encoding time of the event are restored by the decoding circuit.

2.6 Neuromorphic Projects

Spiking neuron networks and neuromorphic hardware have been researched by many research institutions and corporations. In the past 10 years, the SpiNNaker chip of the

University of Manchester, the Neurogrid of Stanford University, the IBM TrueNorth, Intel Loihi are some representatives. Some neuromorphic chip projects are introduced briefly as following:

- **SpiNNaker**[22][23] is a digital neuromorphic computing platform. A single chip contains 18 ARM processor cores and 128 MB of off-chip DRAM memory, and each ARM core can simulate nearly 1,000 Neurons, DRAM memory is used to store synaptic parameters. SpiNNaker can support a variety of neuron dynamics models, including LIF, Izhikevich and Hodgkin-Huxley models, and even support synaptic learning functions. Based on the chip architecture, the SpiNNaker team has further developed a PCB board containing 48 chips. It is estimated that about 1,200 pieces of this circuit board will be needed to achieve one percent of the scale of the human brain. The neurons are interconnected using a hexagonal topology to form a huge neuromorphic network.

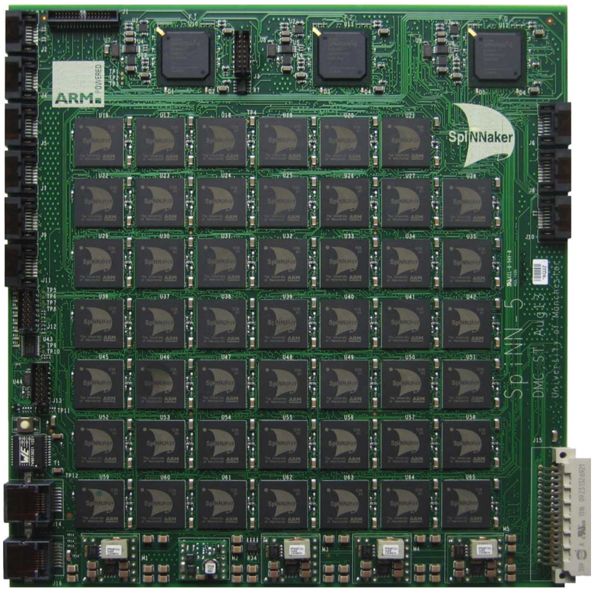


Figure 2.7: SpiNNaker

- **Neurogrid**[3] uses the sub-threshold simulation electrical characteristics of silicon transistors to simulate the dynamic behavior of neuron ion channels in real time, including axon circuits, cell body circuits, refractory periods and calcium potassium ion circuits, synaptic circuits, etc. This method to simulate circuits can support complex neuron models such as Hodgkin-Huxley model. By using the dynamic system method, various neuron models can be mapped to analog circuits. Figure 2.8 shows the Neurogrid system, each computing core contains 256×256 neurons, the 16 computing cores of each circuit board are connected by a tree-topology routing network, which can simulate a million-level neuron network, and the power consumption is only 3.1W. The entire system consists of a motherboard and a daughter board Composition, the motherboard uses analog circuits to achieve neuron dynamics, while the daughter board uses digital circuits to achieve routing communication between neurons.

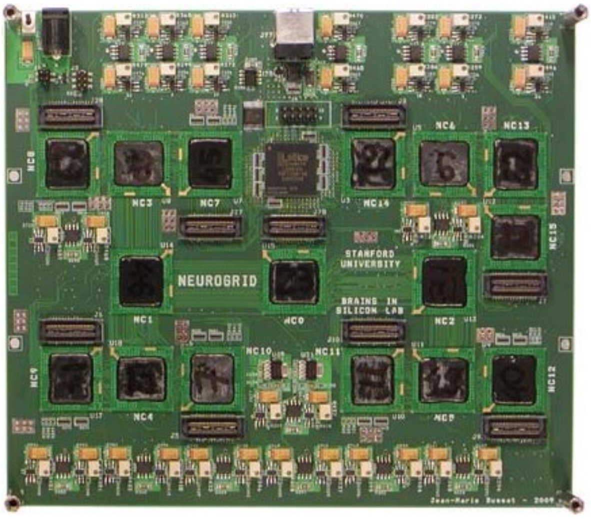


Figure 2.8: Neurogrid[3]

- **DYNAPs**[24] also use transistors to simulate electrical characteristics to achieve neuronal dynamics. It uses a heterogeneous routing topology with higher storage efficiency. 2D grid routing topology is used between chips, and The computing cores in the chip adopt a tree structure routing topology (each parent node has 4 child nodes), and the computing core uses multicast and tag matching for routing packet transmission, and the matching process is content-addressed memory (CAM) is completed. This routing structure can combine the advantages of the low latency of the tree topology and the low bandwidth requirements of the grid topology. The actual manufacturing system configuration is that a single board contains 9 chips, and a single chip contains 4 computing cores, a single core contains 256 neurons, and a single neuron has 64 fan-in and about 4,000 fan-out capabilities.

- **TrueNorth**[12] single chip contains 4,096 computing cores (shown in Figure 2.9). The computing cores are connected by 2D grid routing, which can be expanded by multiple slices. Each computing core includes 256 neurons and a 256×256 size synaptic array. It supports the basic LIF neuron model and its many variants. Each neuron shares 4 strengths of synaptic weights. Due to the use of an event-driven asynchronous synchronous circuit hybrid design, the calculation is only started when there is a pulse event input. The power consumption is only tens or hundreds of milliwatts when simulating a spiking neural network with a scale of one million neurons.

- **BrainScaleS**[25] uses sub-threshold analog electrical characteristics for neuron dynamics simulation. It is produced in 180nm CMOS technology. A single wafer contains 352 chips, a total of 352 × 512 neurons. The digital board is equipped with FPGA for routing communication.

- **Rolls**[26] neuromorphic processor is a mix-signal neuromorphic learning circuit. It was fabricated with a 180nm CMOS process. It uses sub-threshold analog cir-

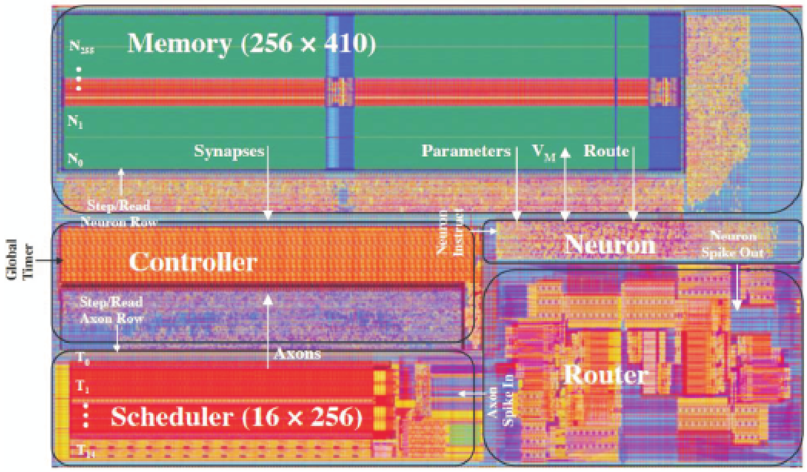


Figure 2.9: TrueNorth[4]

cuits to realize neuron and synaptic dynamics, without pursuing network scale (single chip only has 256 neurons, 256×256 short-term plasticity synapses and 256×256 long-term plasticity synapses). The area is only $51.4mm^2$ and it accomadates approximately 12.2 million transistors.

Part II

Methodology

3 Network and Interconnection implementation

Due to the improvement of integrated circuit manufacturing technology, the design density of chips has gradually increased, but the performance of the bus-based global on-chip data communication method is improved slowly. As the number of cores integrated into a single chip gradually increases, network-on-chip (NoC), routing-based on-chip communication method is proposed.

The on-chip network has the following advantages:

1. Multiple nodes in the network can be interconnected using different physical links. As the number of on-chip cores and links increases, the network bandwidth of the NoC using multi-node interconnection is much greater than that based on the shared bus.

2. NoC can use the same router circuit to expand to different topologies, such as mesh network, butterfly network, etc.

3. As the number of cores in the chip increases, the power consumption caused by data communication also gradually increases. The power consumption in NoC is related to the distance among cores. The shorter the distance is, the lower the power consumption is.

4. In NoC, only adjacent cores are interconnected, which greatly reduces the length of the interconnect links and reduces the difficulty of back-end design and signal integrity design.

5. Multi-core systems based on NoC can use globally asynchronous locally synchronous design to reduce clock power consumption.

For general network on chip, there are three main blocks [27]. First, links are the most important because it physically connects the nodes and acts as the key role to achieve the communication among cores. The second unit is called router. Basically, the router can receive packets (spikes) from the link and transmit the data to the core linked to it or adjacent output link based on the address carried by the packet. The last block is the network interface, which makes the logic connection between the network and cores.

The basic unit of the NoC in this project is named as **Tile**. Tiles can be viewed as the nodes in the network. Each of them connects their neighbors and is required to achieve computing and receiving/transmitting functions. A tile mainly includes **Processing element(PE)** and **Router** which is shown in Figure 3.1.

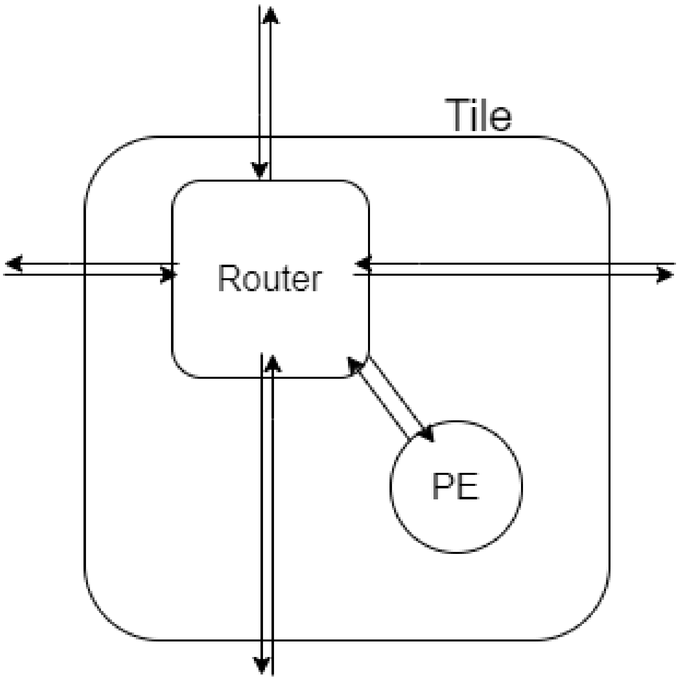


Figure 3.1: The Tile for mesh topology

3.1 Processing Element

3.1.1 Architecture

The Processing Element is the computing unit of the system. Neurons are laid out in neurosynaptic arrays inside. In other words, a Processing Element can be viewed as a collection of neurons as well as AER encoders and decoders.

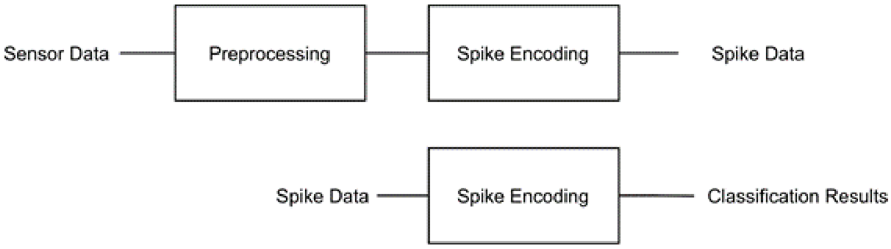


Figure 3.2: Encoder and Decoder Elements

Aside from the neurosynpatic cores themselves, there can be other blocks that produce or consume spikes. In particular, spike encoders take raw data and produce spikes that need to be sent to particular neuron inputs. Encoders basically implement the input layer of the SNN and only produce spikes. Decoders are the opposite in that they receive spikes and perform some analysis on them but do not directly produce new spikes. Finally, there may be off-chip interfaces that produce/consume spikes.

These components, shown in Figure 3.2 can be thought of as implementing the neurons contained at the input layer (encoders) or output layer (decoders) of an SNN.

3.1.2 Broadcast

When mapping spiking neural networks to hardware, a significant number of neurons can be assigned to a single PE. This number can reach 256 or more. Considering that the data channels of PE in all directions are limited, in some cases, certain input ports will have to receive peak traffic, which will cause severe congestion. However, in most cases, the spikes sent to different neurons are exactly the same, because these spikes usually come from the same neuron in the previous layer, which is called fan-in neuron or upstream neuron. The fan-in neuron can only generate and send a spike to the core, and **broadcast** to its connected downstream neurons, which are called fan-out neurons. A simplified neurosynaptic array that can support broadcasting is shown in Figure 3.3. The spike signal is sent from the input port to the internal PE. Each block in the figure can be seen as a weighted synapse for each input that contributes to a certain neuron. The horizontal line connects the external fan-in neuron and all fan-out neurons in this PE. This is the realization of broadcasting.

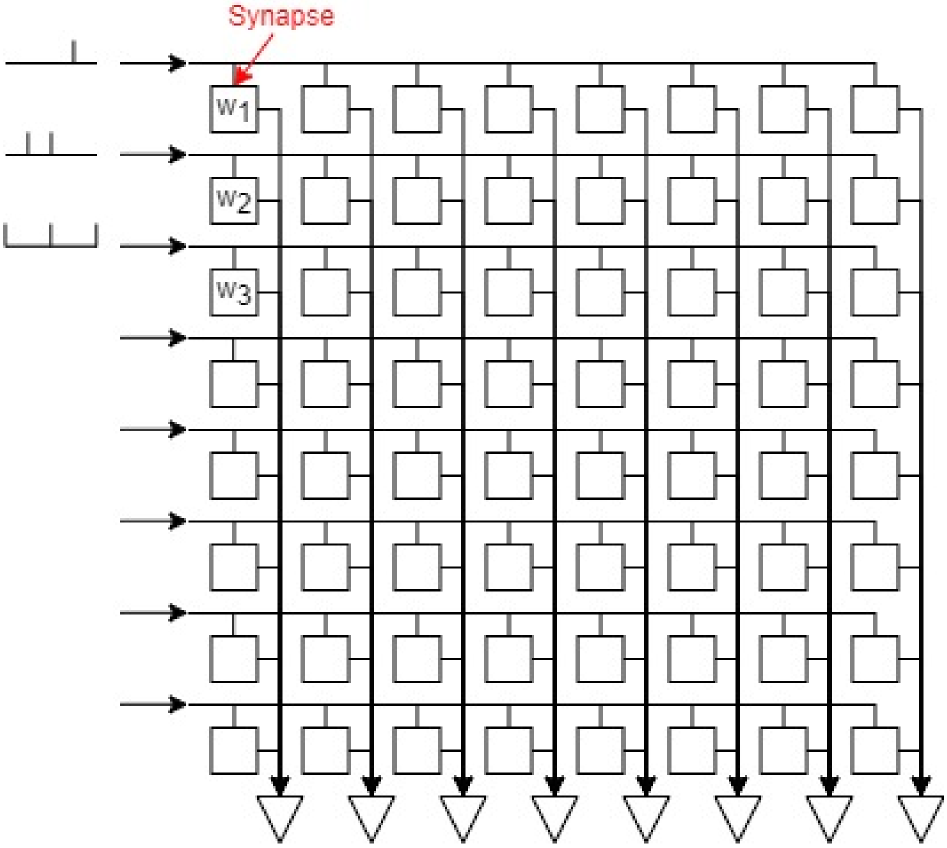


Figure 3.3: Simplified Neurosynaptic Array

3.2 Router

Router is the unit that receive and transmit spikes in the NoC. A router for NoC is required to compose of a number of input ports, a number of output ports that are connected to adjacent shared links, a crossbar switch that connects all the input ports and output ports and determine how spikes flow, and local input and output port to communicate the local PE to the network.

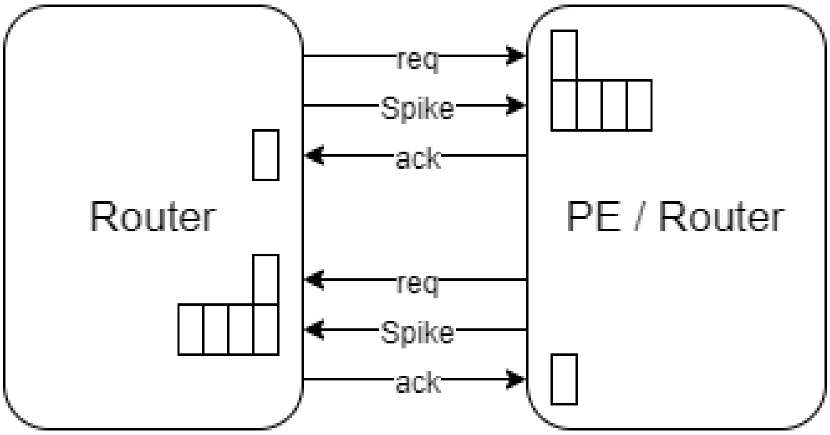


Figure 3.4: Communication schemes between Router-Router or PE-Router

The microarchitecture of a router determines the delay of the critical path which determines delay of each hop and impact the maximum system frequency. It affects the usage rate of links and buffers in the whole network. It also influences the power consumption of the system, including both dynamic energy and leakage energy. In addition, the microarchitecture and underlying circuits have an impact on the area of the network.

When a router receives a spike, it will forward the packet according to the address. A typical data packet will carry its source address and destination address. In this project, the packet destination address is generated based on the destination neuron address. This neuron address is related to the PE address depending on the applying mapping method (described in Chapter 4 in detail). The router will transmit the packet to the destination in reference to the address and routing schemes of topology.

3.2.1 Handshaking protocol

The communications between router and router/PE use traditional handshaking protocol with request/acknowledge signals. The upstream router/PE keeps sending a spike along with a ”request” signal from the beginning of the simulation, which will continue until the downstream router/PE returns the ”acknowledge” signal. Each spike transmission of one hop takes two cycles in the system.

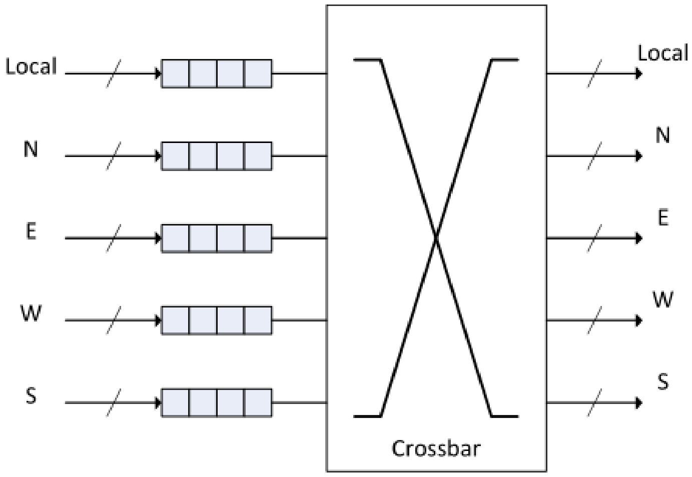


Figure 3.5: A typical mesh router

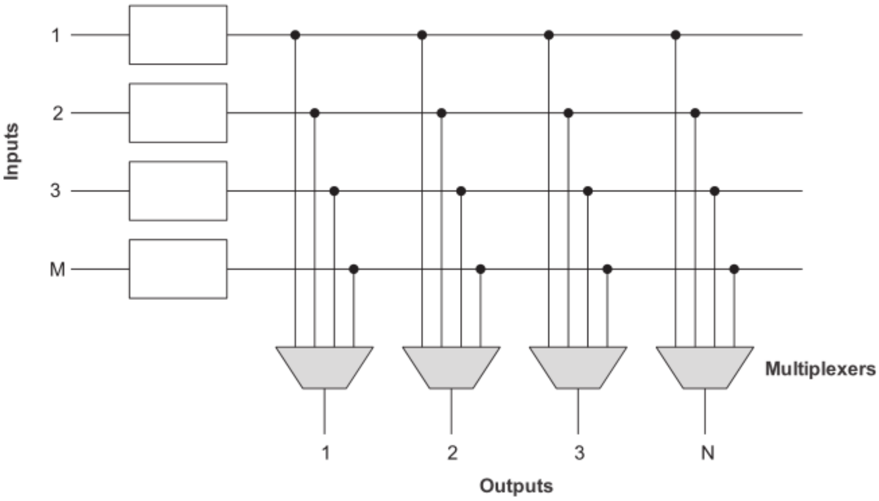


Figure 3.6: Multiplexer-based $M \times N$ crossbar[5]

3.2.2 Crossbar

The crossbar is the core unit of a router. The crossbar is usually a $M \times N$ switch (M input and N output), which is corresponding to the input and output ports of the router. Crossbar can be used to design systems with very high bandwidths. It is usually composed of many multiplexers. Multiplexers are set at all output ports to select data sent from all the input ports. Figure 3.6 shows a typical structure of crossbar.

Evaluation framework architecture 4

In this chapter, the evaluation framework for neuromorphic network hardware is described with emphasis on the design idea, the topology implementation, mapping method, and multiple channels.

4.1 Network Topologies

Network topologies can be viewed as the arrangement of links and nodes of networks. The network topology affects the physical layout of chips and also connections among nodes. Topology and its size can determine the number of hops that a spike is sent through and the wire lengths between hops as well. Both these two factors will affect transmitting latency significantly. Several common network topologies, such as ring, mesh, and torus[28] exists. The network topology For different SNN applications with different traffic patterns, the most efficient network topology can be different. In this evaluation framework, three mainstream topologies (ring, multipath ring and mesh) are implemented.

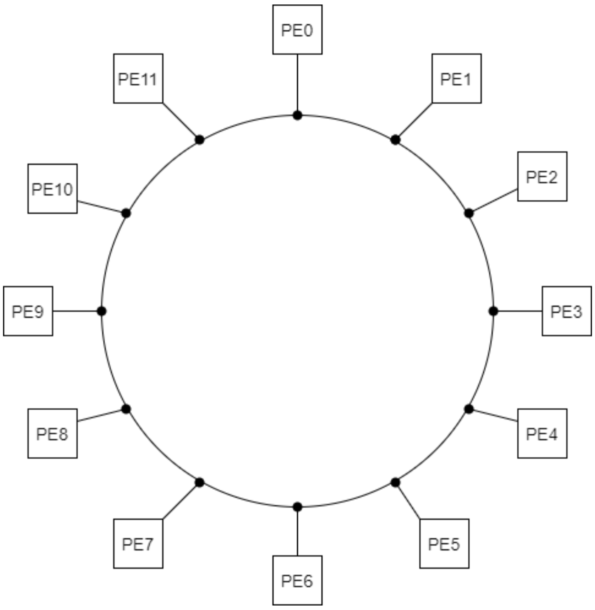


Figure 4.1: A 12-node ring topology

4.1.1 Bidirectional Ring

4.1.1.1 Characteristics

The ring is relatively simple topology where its nodes have only 3 connections. 2 of these ports are connected to the network, while the other one is connected to the neuron array. However, the ring topology leads to a large diameter for large number of nodes in the network, and consequently the hop distance for spikes may be quite large.

4.1.1.2 Routing schemes

The ring topology has only 2 ports for each router to connect the network. The choices for routing spikes is limited compared with MPR and mesh. For this reason, the routing scheme is based on the router index as following:

1. Calculate the distance between the routers of the destination PE and source PE
2. If the distance is larger than half of the ring size, send the spike clockwise.
3. If the distance is equal to or smaller than half of the ring size, send the spike counterclockwise.

4.1.2 Multipath Ring(MPR)

4.1.2.1 Characteristics

Figure 4.2 shows a specific Multi-path Ring, which is similar to the ring. It forms a ring and all node degrees are 5 (four connect to the network and one port links to local PE). MPR has a high clustering coefficient[29]. However, MPR has the same weak point with ring topology: large hop distances when applied to large networks. The diameter of MPR network is:

$$D = \begin{cases} \frac{N}{2} & ,\text{if } N = 2k \\ \frac{N+1}{2} & ,\text{if } N = 2k+1 \end{cases} (k = 1, 2, 3...). \tag{4.1}$$

where N is the number of nodes.

4.1.2.2 Routing schemes

The routing algorithm for MPR topology is presented in Algorithm 1. Since 5 connections for each router can be divided into 3 parts: local, close and far connections, we named 5 directions as Local, Far_cw, Close_cw. Far_ccw, Close_ccw. *cw* and *ccw* represent clockwise and counterclockwise, which are the directions of data flow in the ring.

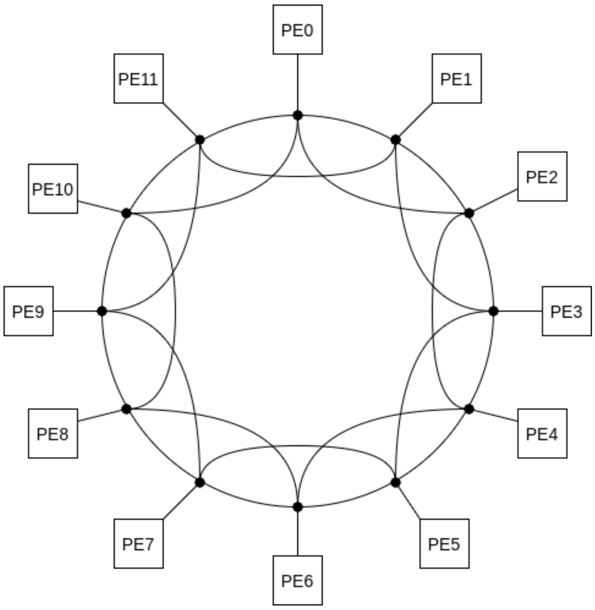


Figure 4.2: A 12-node multipath ring topology

Algorithm 1: Routing algorithm for Multipath ring network

```
Data: Ring size ($N_{ring}$), Index of current node ($I_{current}$) and destination node ($I_{dest}$)
Result: Select output directions
Initialization;
foreach spike in network do
    $I_{offset} = |I_{dest} - I_{current}|$ ;
    if $I_{offset} < \lfloor N_{ring}/2 \rfloor$ then
        if $I_{offset} = 1$ then
            Direction = Close_cw;
        else
            Direction = Far_cw;
    else if $I_{offset} > \lfloor N_{ring}/2 \rfloor$ then
        if $I_{offset} = 1$ then
            Direction = Close_ccw;
        else
            Direction = Far_ccw;
    else if $I_{offset} = 0$ then
        Direction = Local;
end
```

4.1.3 Mesh

4.1.3.1 Characteristics

The two-dimensional mesh topology, which is frequently used in NoC architectures, is built by making all tiles into a rectangular lattice. Each tile connects adjacent tiles in four directions (north, south, west, and east). Figure 4.3 shows a regular mesh topology with 4 × 4 nodes. Each router in a mesh topology also has 5 ports, which is the same with MPR.

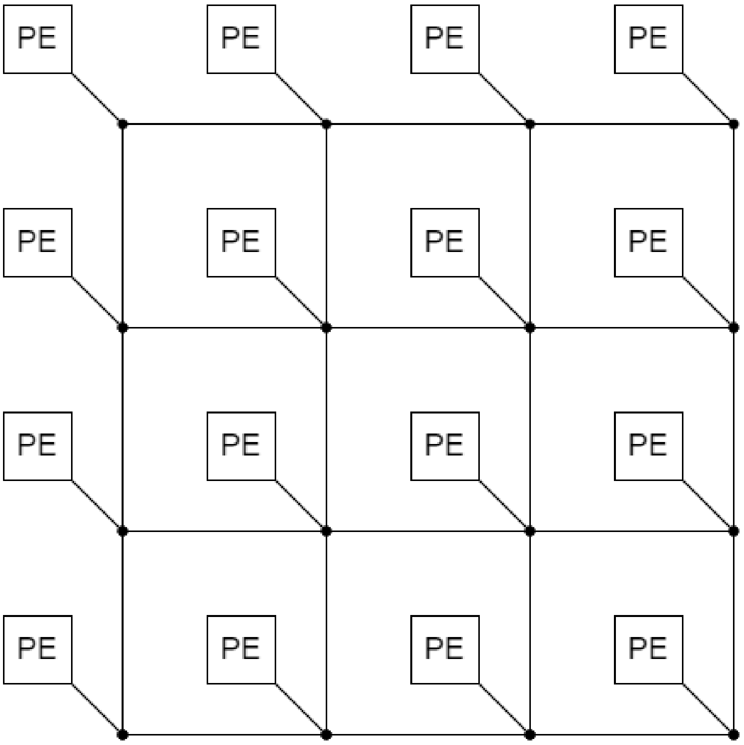


Figure 4.3: A 4 × 4 mesh topology

The mesh network is flexible compared to ring-shape group topologies because it has two-dimensional routing links. In the entire network, the nodes are fully connected, which has the advantages of security and reliability. After a node fails, it will only affect the two nearby nodes. However, since spikes need to be transmitted via other tiles in the network, a large number of overlaps between communication links will happen. This could lead to partial congestion. Thus, resulting in large waiting queues and high latency.

4.1.3.2 Routing schemes

The routing algorithm adopts the XY routing algorithm. It is a fixed target routing algorithm with a simple implementation method and no deadlock. It is suitable for data routing of low-load multi-core chips such as impulse neural network chips. The XY routing algorithm is as follows in Algorithm 2.

4.1.3.3 Conclusion

Table 4.1 shows some typical metrics of different kinds of topologies, where N is the number of nodes in the network, k is the number of buffers in a direction of a single

Algorithm 2: XY Routing for 2D mesh network

```
Data: Coordinates of current tile ($X_{current}$,$Y_{current}$) and destination tile
      ($X_{destination}$,$Y_{destination}$)
Result: Select output directions
Initialization;
foreach spike in network do
    $X_{offset} = X_{destination} - X_{current}$ ;
    $Y_{offset} = Y_{destination} - Y_{current}$ ;
    if $X_{offset} < 0$ then
        Direction = West ;
    end
    if $X_{offset} > 0$ then
        Direction = East ;
    end
    if $X_{offset} = 0$ and $Y_{offset} < 0$ then
        Direction = North ;
    end
    if $X_{offset} = 0$ and $Y_{offset} > 0$ then
        Direction = South ;
    end
    if $X_{offset} = 0$ and $Y_{offset} = 0$ then
        Direction = Local ;
    end
end
```

router. These metrics will be important for the energy and area estimation in Chapter 5.

Table 4.1: Comparison of different topologies

Topology	Degree of tile	Links	Routers	Buffers
Ring	3	2N	N	$k \cdot 3N$
Multipath Ring	5	3N	N	$k \cdot 5N$
Mesh	5	$3N - 2\sqrt{N}$	N	$k \cdot 5N$

4.2 Neuron-to-PE Mapping Problems

Neuron mapping is the procedure that divides neurons into several clusters and mapping these clusters to the mapping algorithm is very important as it determines many properties of the traffic. The key requirement of mapping is to reduce spikes that are sent to the network as much as possible, and hence less congestion and lower power consumption. There are two main considerations to achieve this goal. The first strategy is to group connected neurons to the same PE as much as possible, which makes the communication local, and thus these spikes will not be sent to the network. Another strategy is trying to assign neurons in the same layer to the same PE. Since a spike is broadcast the all the fan-out neurons of the next layer in the same cluster, this method can reduce a large number of duplicate spikes in the global interconnect.

4.2.1 Sequential Mapping

Sequential mapping, as the name suggests, maps neurons sequentially in average based on the neuron index. The steps to implement sequential mapping are described as following:

1. Decide the number of neurons in each PE.
2. Arrange neurons in order based on their index and group them, the number of each group is the result in step 1.
3. Place groups of neurons sequentially in PE based on the index of PEs.

Sequential mapping method does not consider which layer the neurons belong to and the neighbor relationships among neurons. Since delivering traffic pattern contains the information of the number of neurons, the result can be expressed as an array that makes neuron ID and Tile ID corresponds. We assume there is a feed forward network with 8 input, 5 hidden and 3 output neurons and the applied network has 4 tiles, the result is shown in Table 4.2 and is visually represented in 4.4. Since the neuron index are usually ordered from the input layer to the output layer, neurons in the same PE belongs to the same layer in most cases. Thus, sequential mapping can significantly reduce latency by broadcasting.

Table 4.2: Mapping result of neurons in sequential mapping

Neuron ID	**Tile ID**
0,1,2,3	0
4,5,6,7	1
8,9,10,11	2
12,13,14,15	3

4.2.2 Random Mapping

Random mapping is that all the neuron cells are randomly distributed among the PEs in the network for all analyzed topologies. This distribution is easily achieved and it can ensure that the traffic distributions on every link are as even as possible. In this case, if the network can support input traffic, there will be no congestion. However, since the distribution of neurons is totally random, it means that neurons belonging to the same layer will be assigned to different PEs separately and it could be quite far away. Thus, random mapping in most cases can not get significant benefits from broadcast (illustrated in Chapter 3).

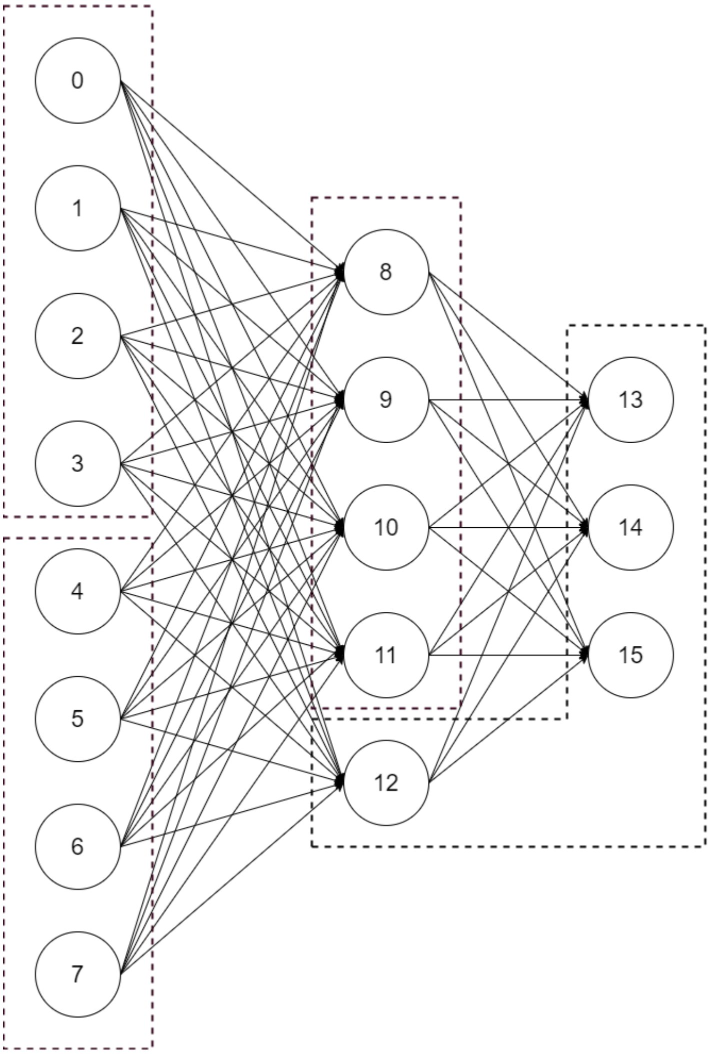


Figure 4.4: An example of sequential mapping for neural network of size 8-5-3

4.2.3 Greedy Mapping

Greedy mapping algorithm was tested as an optional method for mapping in the framework. It is a mapping algorithm to reduce the traffic injected into the network, which could be applied in the ring, MPR, and mesh topology. The idea of this method to achieve this goal is to place connected neurons to the same PE as much as possible. This way can convert more connections to local synapses inside the PE and minimize the tile-to-tile traffic. The steps to place neurons are described as following:

1. Decide the number of neurons in each PE.
2. Select an unplaced neuron with most fan-out neurons and place it to the PE with largest vacancy.
3. Select the neighbor neuron of the neuron in step 2 and place it to the same PE. Repeat this step until there is no neighbors or no vacancy in the PE.

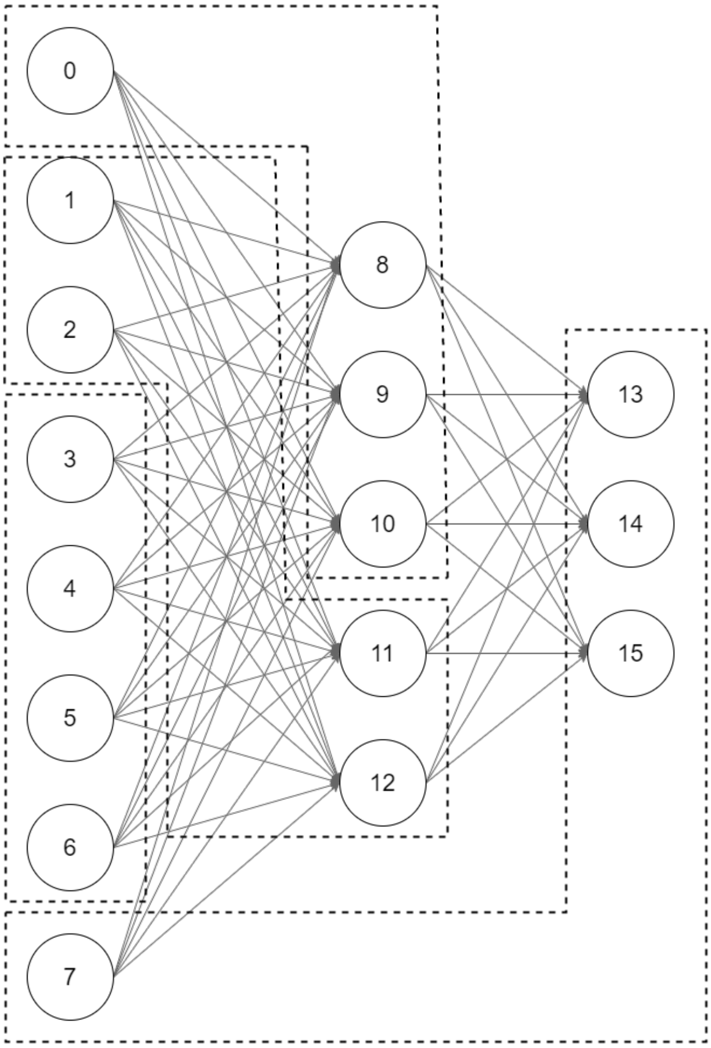


Figure 4.5: An example of greedy mapping for neural network of size 8-5-3

4. If there is no unplaced neuron, the placement is completed. Otherwise, return to step 2.

4.2.4 Center Mapping

Central mapping is proposed as optimization of sequential mapping. It only applies to mesh networks. In large fan-in neural networks, the neurons in the hidden layers and the output layer will be placed on the PE in the lower right corner. This means that some input spikes from the upper left corner will take more delay and greater jump distance to reach the destination, and consume a lot of unnecessary energy. And since all traffic is from top to bottom, from left to right, there will be some hot spots and congestion, and many channels will be underused. Therefore, the center mapping targets to reduce these shortcomings. By mapping the hidden layer and output layer to the center of the mesh, the input layer will be placed around it, which will significantly

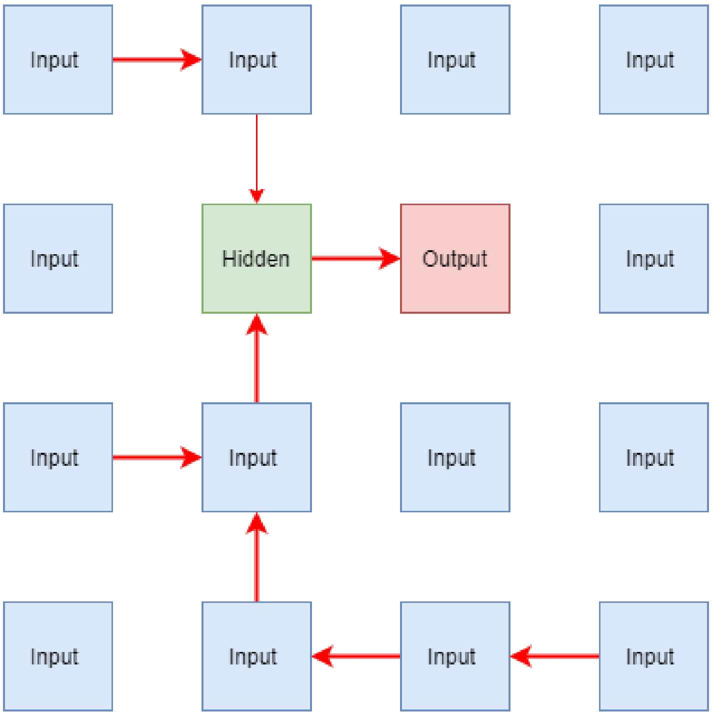


Figure 4.6: Center mapping

reduce the distance from the source to the destination. This does not mean that the hidden layers and the output layer are only placed in one PE. It determines and selects the core PE based on the ratio of neurons in each layer to the total neurons.

1. Calculate the number of neurons in each PE to make neuron distribution in average based on the number of PEs and the total number of neurons .
2. Calculate the number of PEs to be used to place neurons in hidden and output layers.
3. Select most central PEs as the number in step 2 in mesh topology and place neurons in hidden and output layers in these PEs.
4. Assign the rest neurons (neurons in the first layer) around PEs in step 3.

4.3 Multiple channels

Single channel in NoC has small resource consumption. However, it usually works when the number of neurons in one PE is small. But in some cases, when the neuron array is quite large and the injection rate is relatively high or the NoC is running in some specific routing schemes which cause some hot spots in the network, single channel will cause significant congestion. In those cases, multi-channel design is feasible.

The multiple channels design is shown in Figure 4.7. Each channel will correspond to a Fifo (spike queue). As it is mentioned in Chapter 3, when a neuron generates a spike,

Table 4.3: Comparison of different mapping methods

Mapping methods	**Advantages**	**Disadvantages**
Sequential mapping	Easily achieve and good mapping for broadcasting	Large distances and congestions
Random mapping	• Distribute traffic evenly and cause less congestion • Promote utilization of network	Inject large number of spikes into networks
Greedy mapping	Result in more internal connections and less spikes into network	Not suitable for large input pattern
Center mapping	• Relatively short distance • As an improvement method of sequential mapping	• Hotspots generation • Only works for mesh

the spike which has been encoded is sent to the queue and stored in the queue until the request signal is sent from the connected router. If there are multiple channels in the network, a spike will choose which queue it will be sent with a round-robin arbiter. Each spike is not restricted to certain responding channels, which means all spikes can be sent to every channel and the channel choice only depends on the arbitration strategies.

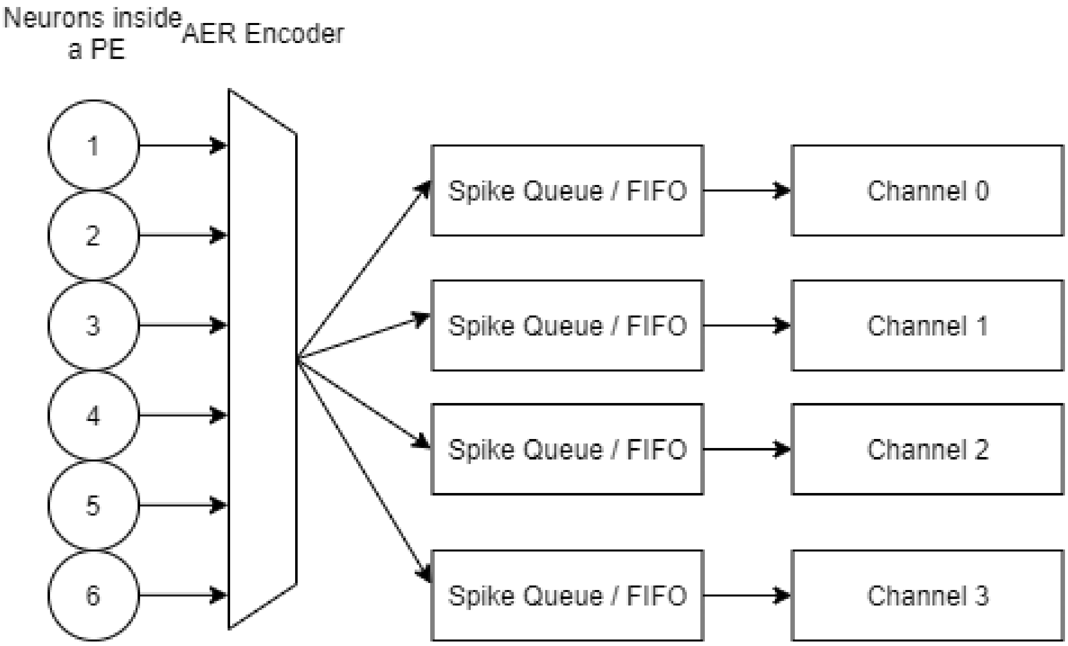


Figure 4.7: An example of multiple channels in PE output ports

Part III

Results

Network simulation and evaluation 5

In this Chapter, some experiments are performed to verify whether the simulator is working as expected and evaluate the effect of some metrics of the network on performance. We succeed in performing simulations with two kinds of feed-forward networks with different input stimulus, the uniform injection rate and spike raster.

Furthermore, this chapter provides a method to roughly estimate power consumption and area. Since energy and area plays a important role in network on chip design, this method can be used in early design to find out the trade-off among performance, energy and area.

5.1 Latency calculation

The latency of each single spike is calculated by two time stamps. It is from the time when the spike is generated by the source neuron to the time when the spike is received by the destination neuron. The latency includes the time when the spike is waiting in the queue before sent to the network. The maximum latency is calculated by traversing all the latency of spikes sent in the network during the simulation time and finding out the maximum result. The average latency is $\frac{\sum t_i}{N_{spike}}$, while t_i is the latency of single spike and N_{spike} is the number of neurons. All the timing values are recorded in cycles.

5.2 Input stimulus

5.2.1 Uniform injection rate

The first traffic pattern for NoC simulation is uniform random process. A uniform random process computes the probability of firing a spike at some interval. We use **injection rate** to represent the probability which varies in 0 to 1. Injection rate is the probability of generating a spike in each cycle. For example, if $p(spike) = 0.5$, which means every neuron will generate a spike in each two cycles in average in a long period of simluating time. To receive the designated spiking rate, clock frequency is needed to take into account. In general, the spiking rate as an input of simulator will be required as a certain value. For example, if the clock speed is 10MHz and the required spike rate of 50Hz, we want one spike every $10M/50 = 200k$ cycles. Therefore, the probability of spiking in any particular cycle is $50/10M = 0.0005\%$.

5.2.2 spike raster

For the neuron arrays, the most important information of spikes or action potentials is timing. Therefore, a more realistic spike traffic pattern can be obtained by using spike rasters generated by the MATLAB simulator. In Figure 5.1, a synchronized raster plot is shown. In the plot, each node represents a generated spike. All the spikes are sent synchronously, which means spikes generated in a period of time are buffered until a synchronization event and sent out together as a event.

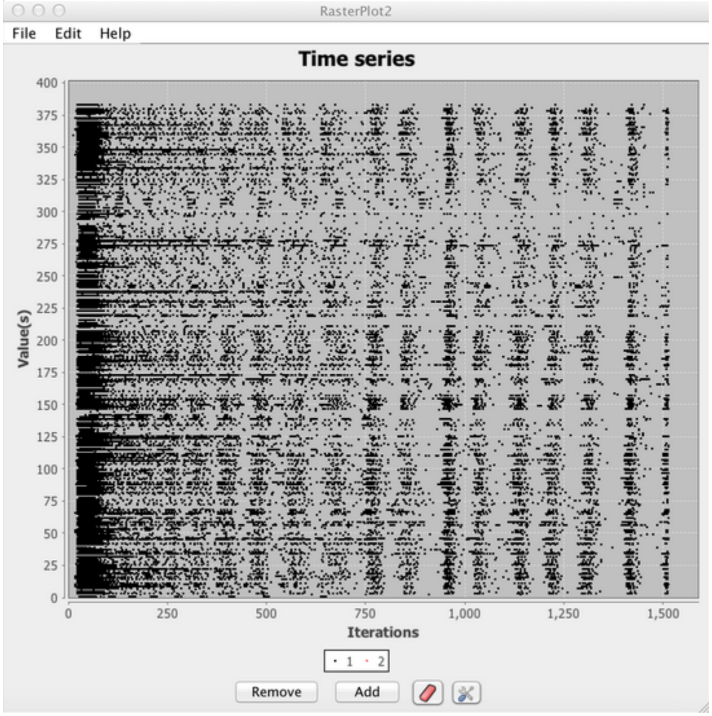


Figure 5.1: Raster plot[6]

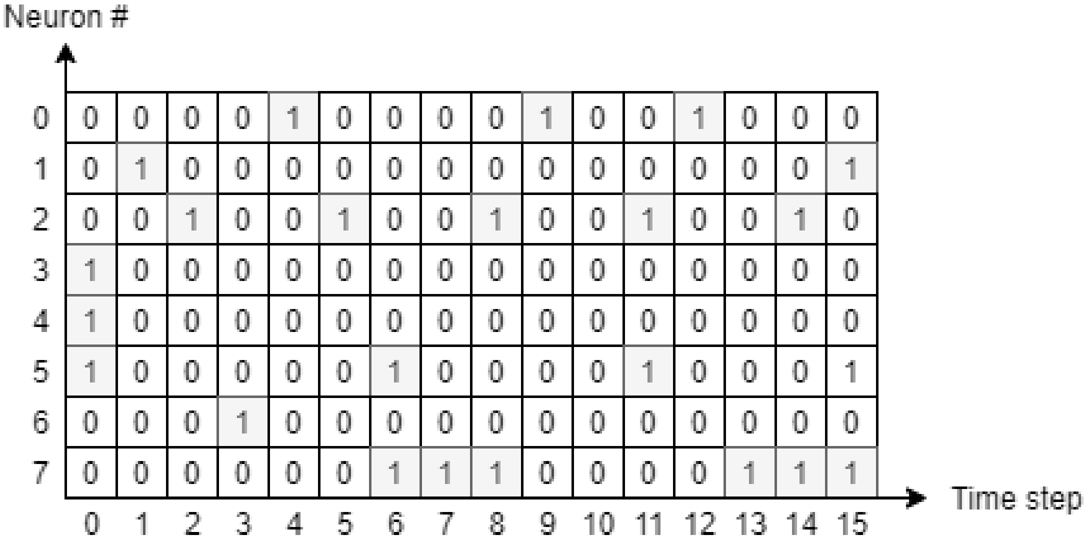
A spike raster is a 2-D binary matrix(shown in Figure 5.2). The first (Y) axis represents the neuron index and the second (X) axis represents a **time step**. At each time step, the column of values shows which neuron generated a spike. For example, in Figure , in time step 0, neurons 3, 4 and 5 generated spikes. A time step depends on the step size, for example a step of 1 microsecond results in spikes being generated at 0us, 1us, 2us.. 15us.

5.3 Performance simulation

5.3.1 Assumptions

To simplify the problem, a set of assumptions are made:

- All neurons, including those in middle layers, generate spikes based on the input stimulus. The behavior of neurons in hidden layers has already taken into account.
- All data packets (spikes) are of same size and each packet has only one flit.
- The simulation with random mapping in the same topology and the same size of



Neuron #	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
2	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	1
6	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	1	1	1	0	0	0	0	1	1	1

Figure 5.2: An example of spike raster

network uses the same random seed and can implement the same neuron distribution.

- The size of queue in PEs is infinite as buffers to store generated spikes in PEs.

5.3.2 Simulation with uniform injection rate

5.3.2.1 Experiment Setting

In this experiment, unless otherwise specified, the setting of the network is presents in Table 5.1. While the buffer size represents the number of buffers in one direction of a router and simulating time represents the time that the simulator was running. The generating time is the time that neurons are generating spikes from the simulation startup, When the time is not within the generation time, the injection rate is 0, neurons no longer produce spikes.

Table 5.1: Configuration of network with simulation of uniform injection rate

The number of layers of neural network	3
The number of neurons in one layer	20
The number of total neurons	60
The number of PEs	20
Buffer size	4
Mesh size (row × column)	4 × 5
Simulating time (cycles)	100k
Generating time (cycles)	20k

5.3.2.2 Simulation results

In the uniform random process, every simulated neurons generate spikes and send to the network. The destination of spikes are randomly generated and this traffic pattern is tested for the average performance and traffic capacity of network.

Figure 5.3 plots the latency curve versus spike injection rate with different topologies. The plot shows that mesh topology has the best performance and ring topology is the worst. The curves of different topologies in the plot have the similar trends. when the injection rate is small, the latency is kept low and does not have significant increase as the injection rate increases. When the injection rate increases to a turning point, the average latency increases rapidly. The turning point, which we could called saturation point, represents the critical injection rate, the peak traffic that the NoC can support. When the injection rate is larger than this value, the number of spikes generated by neurons are more than that the network can transmit in time. Thus, spikes will wait and accumulate in queues and buffers, which increase the latency significantly.

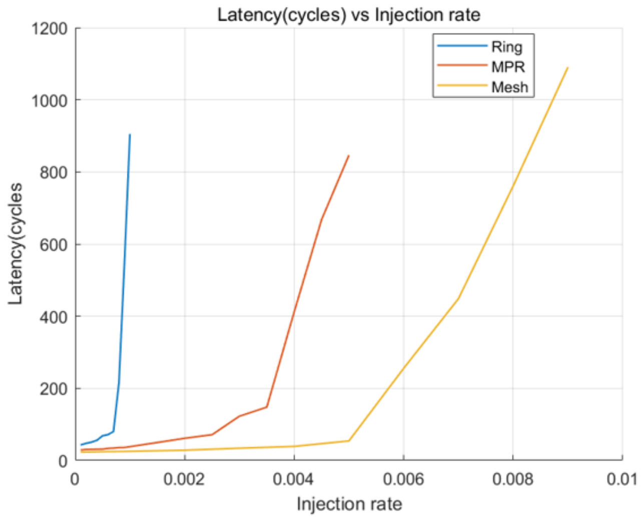


Figure 5.3: Average latency vs. Injection rate in different topologies

Figure 5.4 and Figure 5.5 plots average latency when using different mapping methods in Mesh and MPR topology. We can conclude that random mapping results in the lowest average latency and greedy mapping gets the worst for both mesh and MPR. The reason is that the number of neurons in this traffic pattern is limited and there are only a small number of neurons in each PE. In the given case there are 3 neurons per PE, which means broadcast has little effect on latency and the advantage of the uniform traffic of random mapping is revealed.

5.3.3 Simulation with spike raster

5.3.3.1 Experiment setting

In the simulation, a four-layer feed-forward network which is presented in Figure 5.6 is implemented to evaluate our framework. This spike neural network has 1088 input

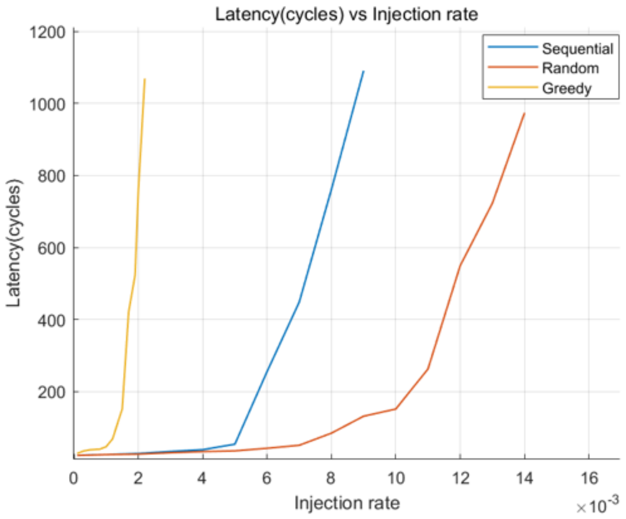


Figure 5.4: Latency comparison of different mapping methods in Mesh topology

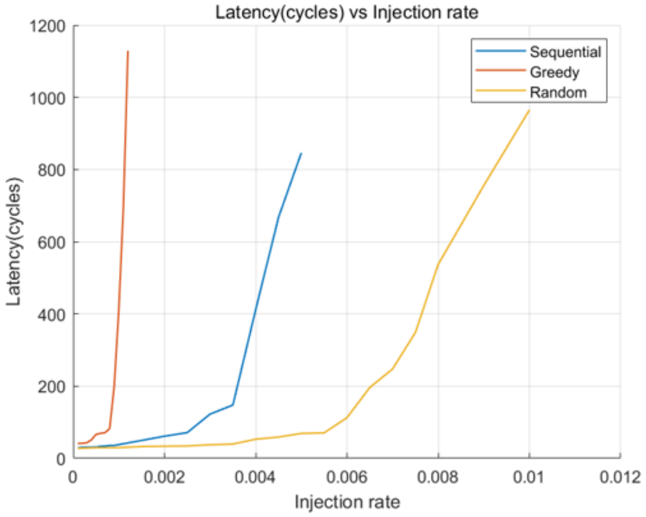


Figure 5.5: Latency comparison of different mapping methods in MPR topology

neurons, 4 output neurons and 20 and 10 neurons in hidden layers respectively. This network is full connected. The configurable parameters and settings include the topologies, the size of network, the buffer size, the number of channels and mapping methods. Unless otherwise specified, the experiment setting of the simulator is presents in Table 5.2.

5.3.4 Performance simulation

5.3.4.1 Topology

In Table 5.3 to 5.5 the results of Mesh, MPR and Ring topology are shown. They include queue size, maximum and average latency of different mapping methods and the

Table 5.2: Configuration of network with simulation of spike raster

The number of PEs	36
The number of total neurons	1122
Buffer size	4
Mesh size	6×6
Simulating time (cycles)	100k
Clock frequency	50MHz
Time step	600 cycles

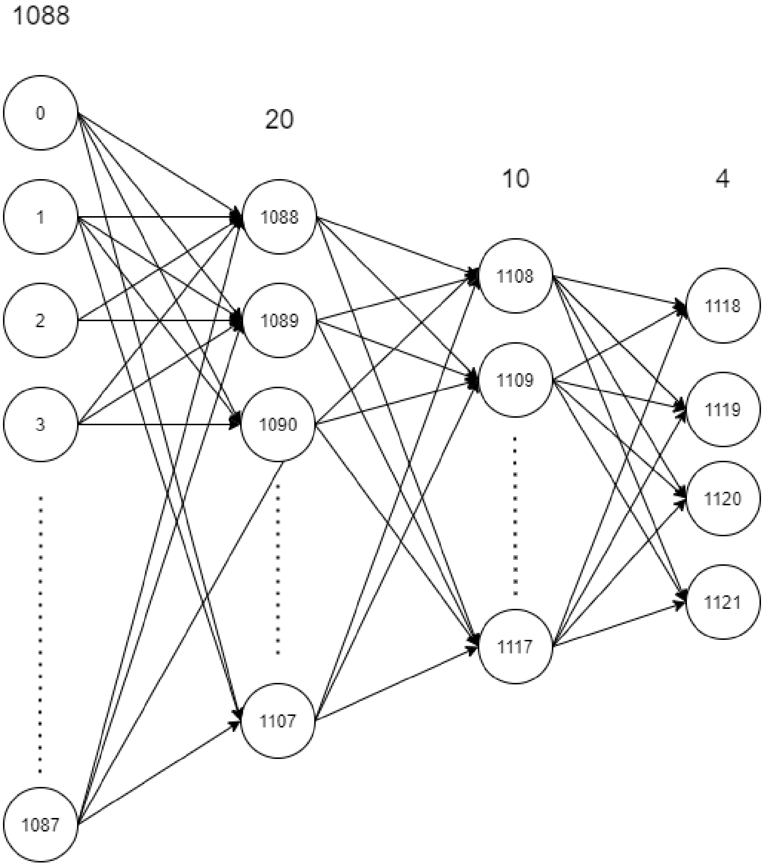


Figure 5.6: a four-layer full-connected feed-forward network (1088-20-10-4)

number of channels. When using spike raster as stimulus, maximum latency determines the highest system frequency, therefore what we concerns most is maximum latency instead of average latency. The influences of single metric is presented in the following sections. To make explanation easier, we mainly use Mesh and MPR.

5.3.4.2 Broadcast

The first issue that should be addressed is how broadcast influence the latency. The traffic throughput and latency evaluation was performed, with and without broadcast in two kinds of mapping methods, sequential mapping and random mapping in mesh

Table 5.3: Performance of Mesh topology

	Number of channels	1	4	8	16	32
Center	maximum latency	2179	547	275	139	71
	queue size	32	8	4	2	1
	average latency	188.287	52.9643	30.4378	19.2261	13.6293
Sequential	maximum latency	2173	547	275	139	71
	queue size	31	8	4	2	1
	average latency	258.45	68.3615	36.8595	21.1122	13.2424
Random	maximum latency	3923	979	484	230	113
	queue size	504	126	63	32	16
	average latency	260.192	65.6971	36.4354	22.752	16.465
Greedy	maximum latency	2157	543	273	139	73
	queue size	117	30	15	8	4
	average latency	84.1347	30.8194	22.4613	18.3971	16.4299

Table 5.4: Performance of MPR topology

	Number of channels	1	4	8	16	32
Sequential	maximum latency	2177	433	273	137	71
	queue size	56	14	7	4	2
	average latency	162.93	29.8837	21.1564	12.0998	7.70895
Random	maximum latency	3250	2951	429	212	108
	queue size	728	741	91	46	23
	average latency	235.142	63.0408	31.134	19.1763	13.4049
Greedy	maximum latency	2177	497	273	137	69
	queue size	491	134	60	30	15
	average latency	151.96	34.8013	21.0227	12.6759	8.87381

Table 5.5: Performance of Ring topology

	Number of channels	1	4	8	16	32
sequential	maximum latency	2131	535	269	137	71
	queue size	56	14	7	4	2
	average latency	320.487	102.596	66.2823	48.1683	39.1112
random	maximum latency	12174	3116	1587	790	389
	queue size	728	182	91	46	23
	average latency	1056.44	266.826	137.916	76.3558	47.4038
greedy	maximum latency	2173	547	305	139	73
	queue size	56	14	7	4	2
	average latency	309.071	118.623	96.556	71.026	63.3873

topology. Table 5.6 presents the comparison of latency performance of network and indicate how broadcast can promote the performance. The total number of spikes that all the neurons generate is 134,088. In the network with sequential mapping, neurons in the same layer are often mapped to the same or neighbor PE. With broadcast, there is sharp decrease in the number of spikes injected to the network and correspondingly, maximum latency in the simulation with broadcast is significantly lower than that in the simulation without broadcast. In contrast, the performance of network with random mapping with broadcast does not improve much as expected because neurons are randomly placed. We can conclude that broadcast is the method to reduce the latency by reducing throughput of network and it can play a bigger role in some adjacent mapping methods like sequential mapping.

Table 5.6: Maximum latency and number of spikes with broadcast/non-broadcast in mesh network

		Broadcast	**Non-broadcast**
Sequential Mapping	Max latency (cycles)	547	11125
	Number of spikes	9208	134088
Random Mapping	Max latency (cycles)	1178	1387
	Number of spikes	110828	134088

5.3.4.3 Number of channels

The effect of changing the number of channels on performance are shown in Figure 5.7. The number of channels varies from 1 to 32. Due to the relatively large numbers of spikes injected to the network based on the spike raster and limited traffic capacity, a single channel results in high latency. The result can be more than 2000 cycles, which is higher than cycles of a time step. Large maximum latency means spikes cannot reach the destination in one step. That leads to errors in the system. When the number of channels increases, the maximum latency decreases. We can conclude that the maximum latency is inversely proportional to the number of channels.

5.3.4.4 Mapping methods

Figure 5.8 shows the maximum latency performance of three different mapping methods in the multipath ring topology. Random mapping results in the highest maximum latency. As explained in the Table 5.6, random mappings get less benefit from broadcasting. Multiple copies of the same spikes from the same neurons are transmitted to the network system and have to wait in the FIFOs of PEs and routers until there are free channels. Sequential mapping and greedy mapping gave approximately close results in the simulation and mapping methods matters more in small numbers of channels. When a large number of channels are used in the network, the maximum delay results of different mapping methods are very close and even the same. The reason is that a large number of channels provide enough traffic capacity to the network, and the injected spikes do not reach the upper limit of the traffic that the network can support.

Figure 5.9 presents the maximum latency performance of different mapping methods in mesh topology. Since the random mapping result in a quite large latency, it is not shown in the figure. For the rest three mapping methods, the sequential mapping has

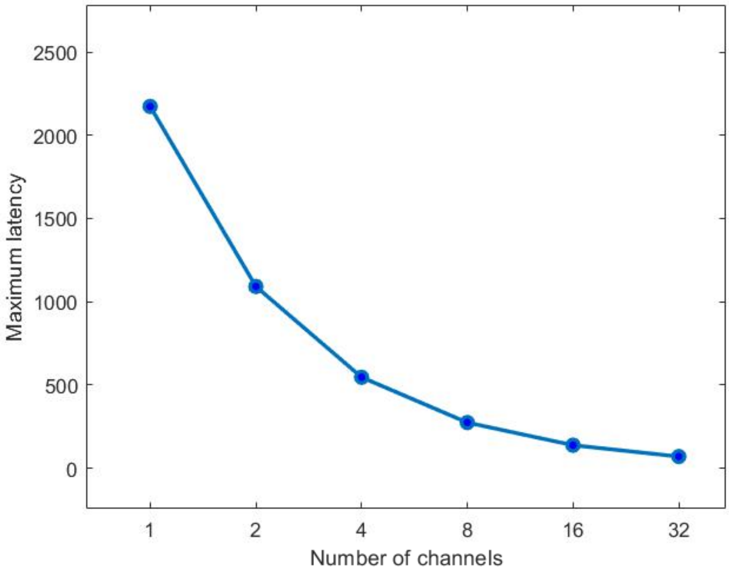


Figure 5.7: Maximum latency vs. numbers of channels with sequential mapping

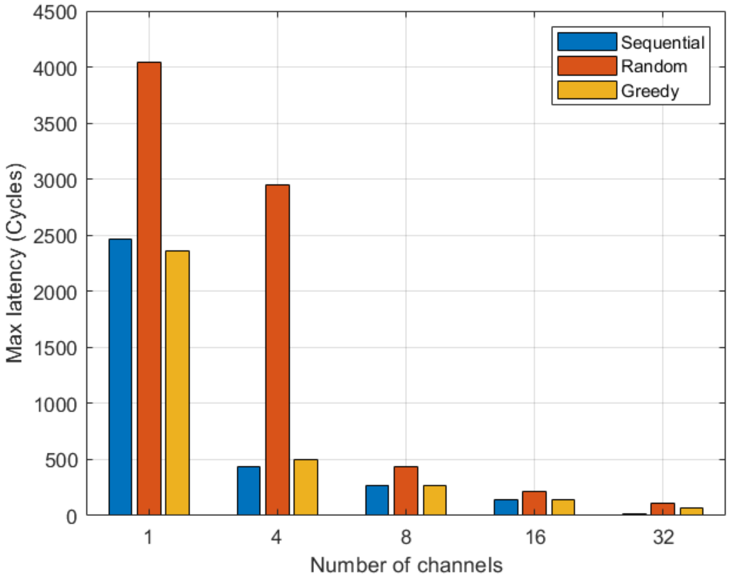


Figure 5.8: Latency comparison of different mapping methods and different numbers of channels in Multipath ring

the worst performance and the greedy mapping has the best. The result is similar to MPR.

We visualize the spiking traffic of 6 × 6 mesh topology in all four different mapping methods in Figure 5.10. Each subfigure is composed of 36 squares, which represents 36 tiles in the network. Each square consists of 5 small colored blocks. Four around

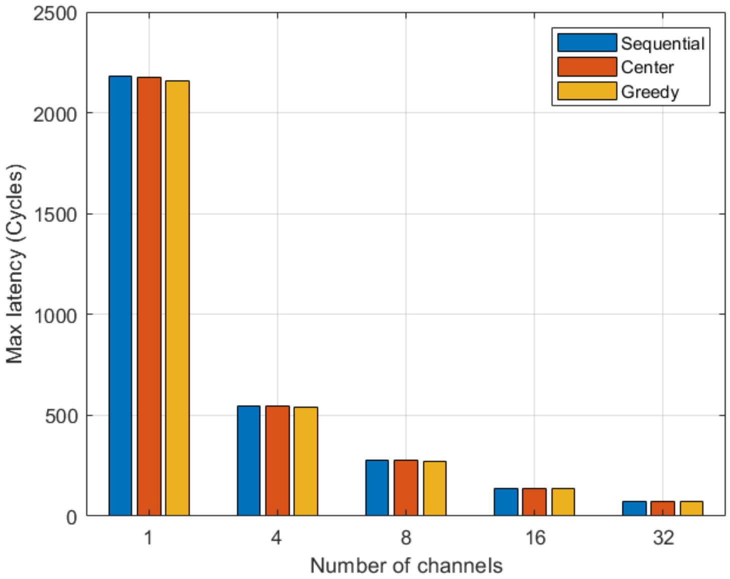


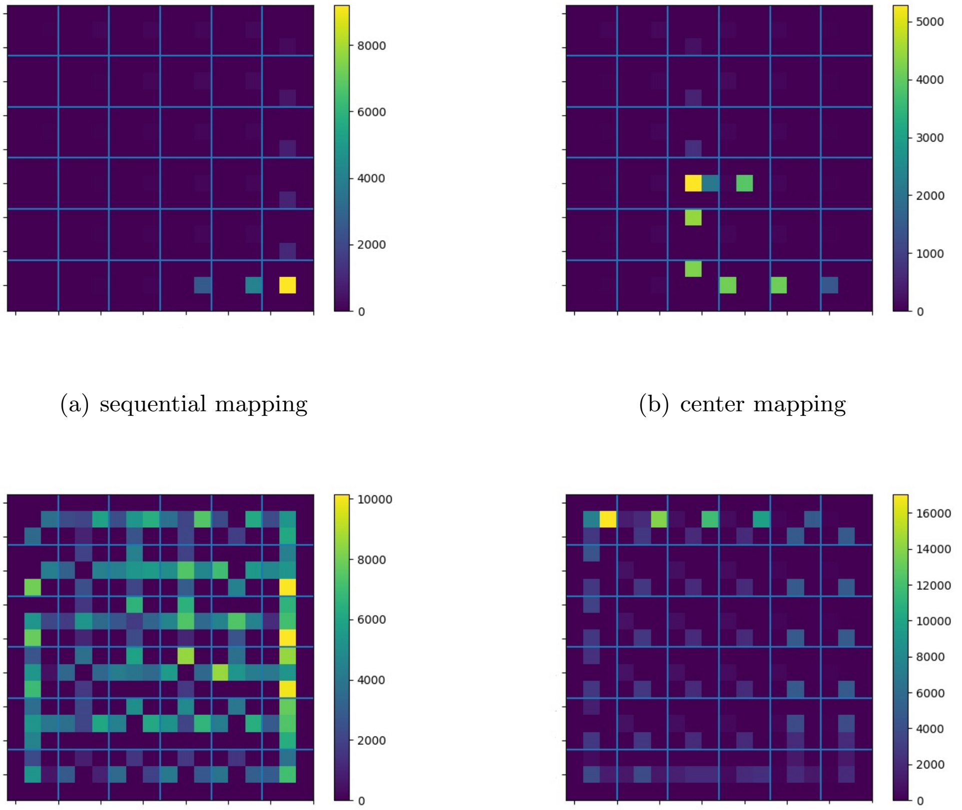
Figure 5.9: Comparison of different mapping methods with different numbers of channels in Mesh topology

represents four directions of the router, and the center block represents the the channel to the local PE. The brighter the color block is, The larger traffic in the corresponding direction of the router is.

Since spikes are unidirectionally transmitted in the feed forward network and the number of neurons in output layer are quite small compared to total number of neurons, the neurons in output layer are all placed in the lower right corner in sequential mapping. Thus, all the traffic transmission is from top left to bottom right and the tile in the lower right corner receives the highest traffic. Similarly, the network using center mapping has the highest traffic in the center. Network using random mapping has the highest and most average throughput. All the visualized traffic meets the design idea and the simulation results.

5.3.4.5 Time step and clock frequency

In common spike rasters for SNN application, the timescale of time step of input is $1\mu s$, and different SNN-based NoC systems could work in different clock frequency. Therefore, The cycles in a time step are different. To make the repeated simulation easier, we change the clock cycles in single time step to simulate different clock period instead of changing system frequency directly. To make the effect of clock frequency more intuitive, we use ring topology which has smaller traffic capacity and a spike raster with a very large injection rate as the stimulus. In this section, the comparison between networks with a time step of 25 cycles and 600 cycles shows in Figure 5.13 and the corresponding clock frequency is in Table 5.7. Obviously, A network with a time step of 25 cycles performs worse, especially in small number of channels. The reason is that the simulation with a time step of 25 cycles does not guarantee that



(a) sequential mapping

(b) center mapping

(c) random mapping

(d) greedy mapping

Figure 5.10: The visualized spiking traffic for four mapping methods in mesh

the spike will reach its destination in 25 cycles. This will cause congestion and this congested spikes will be accumulated in FIFOs and queues during the simulation time and become serious as the simulation runs. Thus, this congestion will be reflected in the performance.

Table 5.7: Correspondence of time step and clock frequency

Time step (cycles)	**Clock frequency (μs)**
25	0.5
600	12

5.3.4.6 Network size

The larger mesh size results in large hop distances, which means more average hops cost for spikes in the network. On the other hand, a large mesh size provides more traffic capacities and leads to less congestion in the router buffer. Figure 5.11 shows

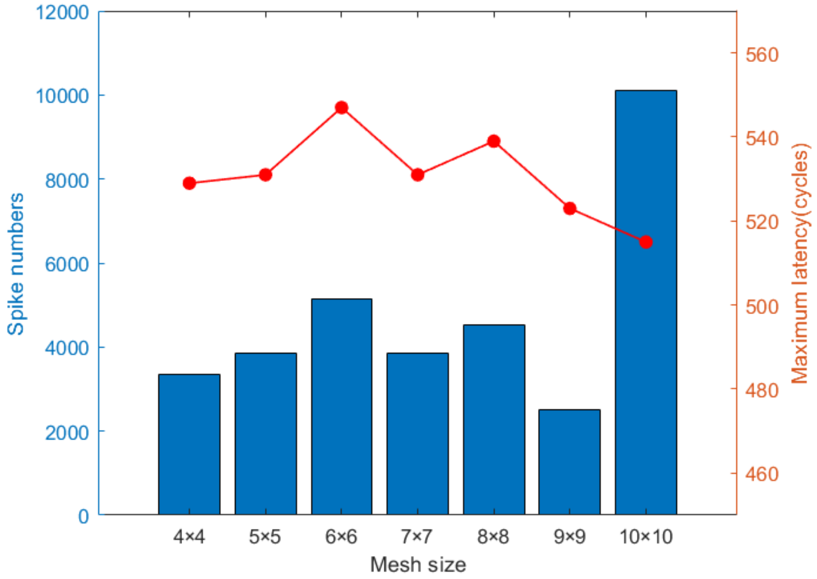


Figure 5.11: The number of spikes and maximum latency in different mesh size

the comparison of the effect of mesh size in the network. In most cases, the maximum latency and the number of spikes are positively correlated and the spike numbers of test cases are all around 4000, except for 10×10 mesh. The number of spikes in 10×10 mesh reaches a very high result and the maximum latency is even smaller. The reason is that larger network size results in fewer neurons in each tile, which means that the extremely large network size could be possible to cause widely distributed neurons of the same layer and the benefit of broadcast would be small. A similar result of MPR topology is shown in Figure 5.12.

5.3.5 Conclusion

On the basis of the above experiments, we can conclude that multipath ring topology with sequential mapping with 25 tiles and 4 channels could be a feasible option for delivering spiking traffic pattern in the experiment. This configuration combination has the lowest maximum latency and relatively low traffic in the network.

5.4 Power and Area estimation

5.4.1 Power

In the digital circuits, the energy includes two components: the dynamic energy, which signals switch from one level to another, and the static energy which is mainly the leakage energy of buffers. The dynamic energy can be expressed as:

$$E_{dynamic} = 0.5\alpha C V_{DD}^2 \tag{5.1}$$

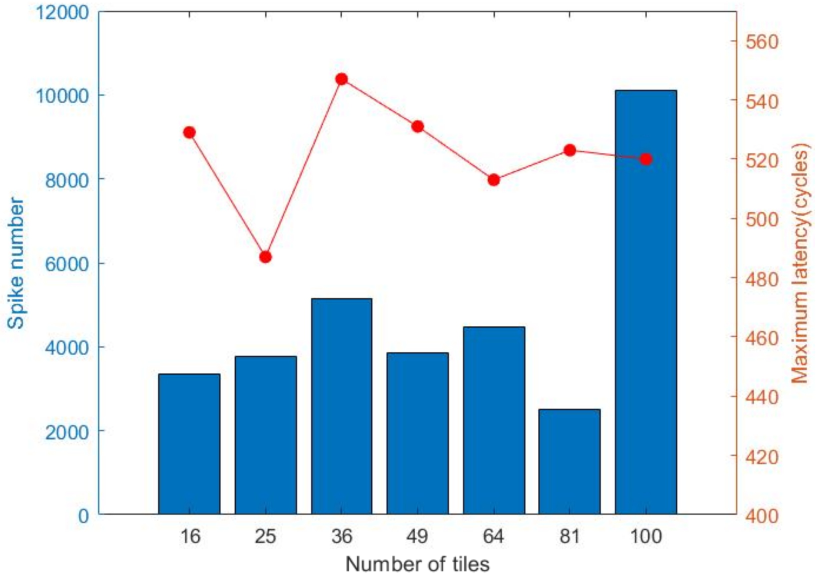


Figure 5.12: The number of spikes and maximum latency with different tile number in MPR topology

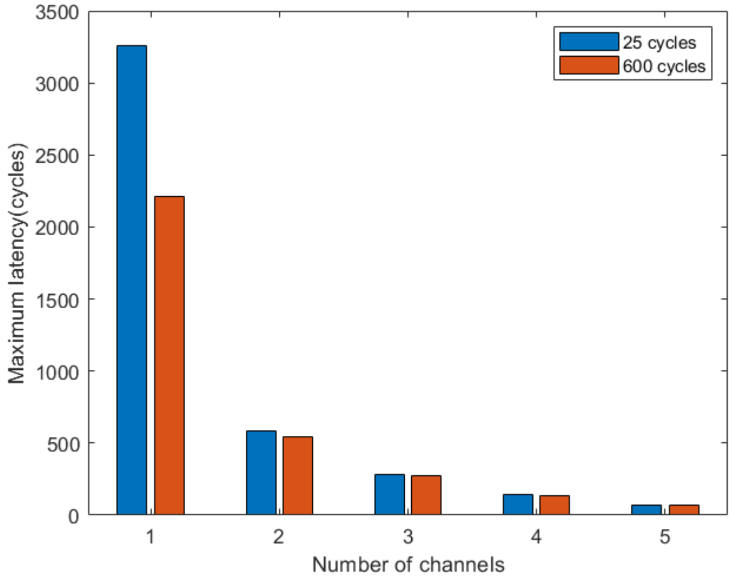


Figure 5.13: Maximum latency vs. number of channels in a time step of different cycles

where α is a value is related to the clock frequency f_{clk}, C is the total switching capacitance and V_{DD} is the supply voltage.

There are various methods that have been proposed to evaluate power consumption and the area of a network. [30][31] proposed a power-area interconnection network simulator named Orion 1.0/2.0. For power consumption estimation, Orion makes a set of architectural power models for interconnection routers. The power models include clock, allocators and arbiters, FIFO buffers, and physical links. These power models are implemented in low-level architectural descriptions. The area models are divided into router area and link area. The area estimation in Orion 2.0 is computing the area of each building block separately and summing them up.

Since almost all the methods in the literature use low-level description models to evaluate power consumption and area, which is not feasible for our framework, a rough estimation is needed. [32] proposed a method by estimating the average energy of a flit based on the hops that it passes through. The power consumption for every single spike can be expressed as follows:

$$E_{spike}(d) = d \cdot E_{link} + (d+1)E_{router} \tag{5.2}$$

where E_{link} and E_{router} is the energy that a spike traverses a link and a router, and d represents the number of links it passes through. The total energy is to sum the energy of all spikes in the network up.

We use the similar normalized method in [33]. For the MPR network, we would consider that the far links will be longer than close links when implementing on the hardware, the E_{link} is given by 5.3:

$$E_{link} = \begin{cases} 1 & \text{,if close link} \\ 2 & \text{,if far link} \end{cases} \tag{5.3}$$

For the mesh network, in physical implementation, the vertical links are longer than the horizontal links. Thus, the E_{link} is given as following:

$$E_{link} = \begin{cases} 1 & \text{,if horizontal link} \\ 4 & \text{,if vertical link} \end{cases} \tag{5.4}$$

Figure 5.14 shows the comparison of power consumption of mesh and MPR NoC with different mapping methods.

Figure 5.15 presents how energy changes when applying different mesh sizes and Figure 5.16 shows the average distance of spikes in NoCs of the different mesh sizes. The average distance is to calculate the average number of hops that all spikes in the network traverse. Compared Figure 5.15, 5.16 and 5.11, the energy is highly proportional relationship to average hop distance and the number of spikes sending to the network.

5.4.2 Area

The rough area estimation is expressed as following:

$$A = N_{router} \times N_{bufferSize} \times A_{buffer} + \sum A_{link} \tag{5.5}$$

where N_{router} is the number of routers in the network, $N_{bufferSize}$ is the number of buffers in a single router. In the simulation of this project, all the $N_{bufferSize}$ is always 16. The area of a single buffer is set to 30.

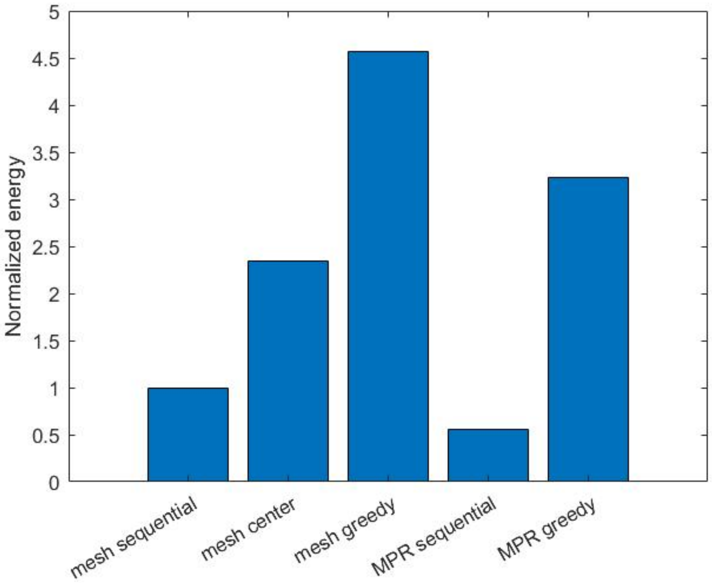


Figure 5.14: Normalized energy in mapping methods in MPR and mesh

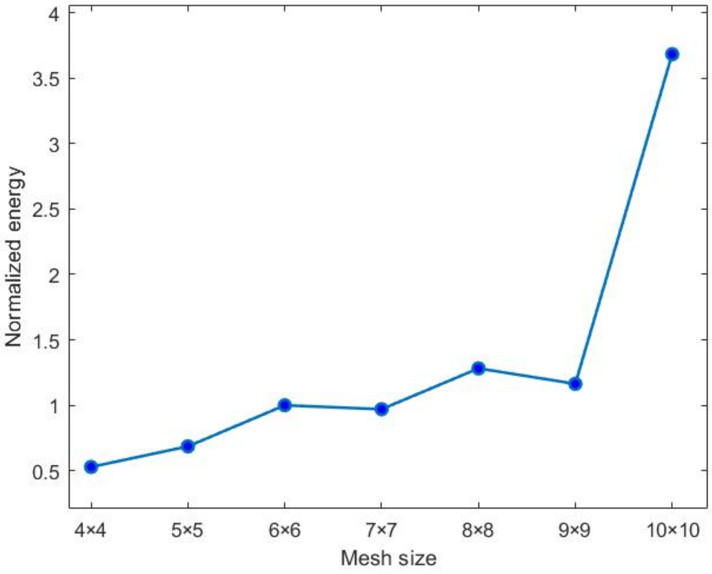


Figure 5.15: Normalized energy in different mesh size

We set a parameter for links in different topology, for the MPR network:

$$A_{link} = \begin{cases} 1 & \text{,if close link} \\ \sqrt{2} & \text{,if far link} \end{cases} \tag{5.6}$$

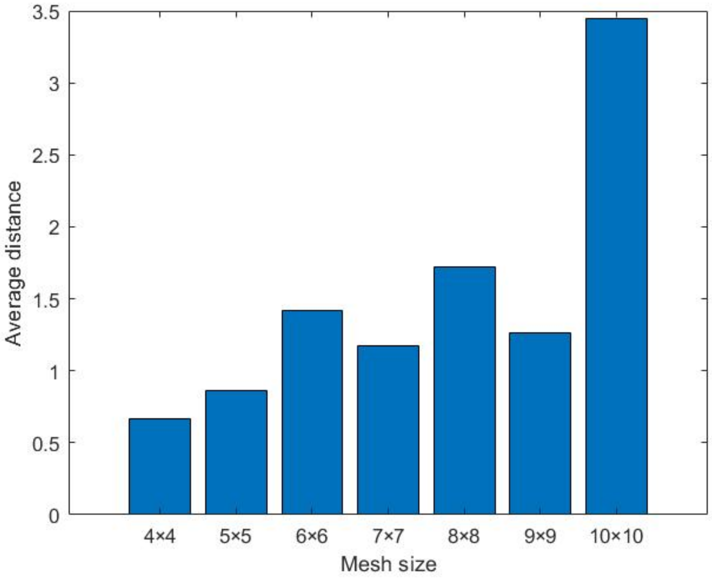


Figure 5.16: Average distance of spikes in different mesh size

For mesh network:

$$A_{link} = \begin{cases} 1 & \text{,if horizontal link} \\ 2 & \text{,if vertical link} \end{cases} \tag{5.7}$$

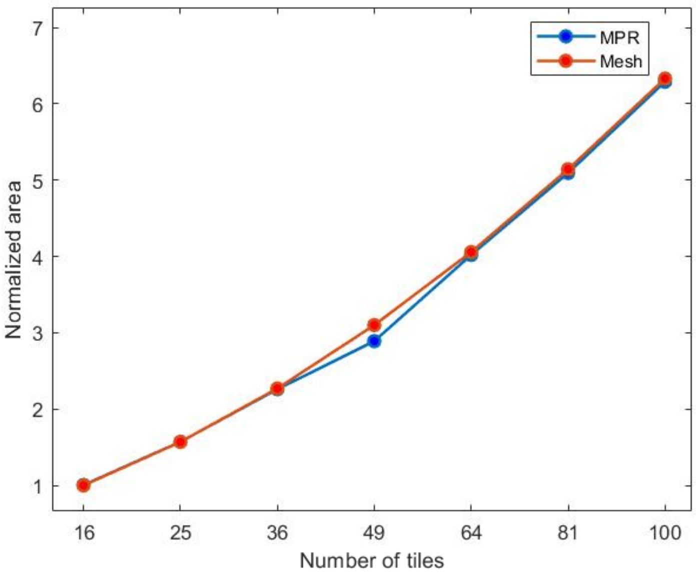


Figure 5.17: Area estimation of network with different mesh size

In this case, we can derive the normalized area comparison in Figure 5.17. The area is proportional to the size of the network. Using different topologies while using the same number of cores will not make much of a difference in area. This is because we set the tile area as a relatively large number and the link area doesn't occupy much normalized area in the system. However, the accurate parameter depends on the area parameters of the real device and should be set based on the selected process and device size. It is worth emphasizing that this section just provides a possible method for area estimation and cannot be used for accurate calculation.

Conclusion and future work 6

6.1 Conclusion

In this thesis, we have implemented a cycle-accurate network simulation framework for SNN-application that can receive spike input traffic patterns. The framework is configurable with three topologies, four different mapping methods, different network size,s and the number of channels. In addition, this thesis also provides a rough method to estimate the energy and area cost of the network on chip, which could present a trade-off among performance, power, and area.

To find an efficient solution, the framework is tested by using a spike raster as the input stimulus to test its functions and the influence of every single metric on performance and compare broadcast and non-broadcast. The result shows that multipath ring topology with sequential mapping with 25 tiles and 4 channels has good performance for this specific case.

6.2 Future work

Since this thesis is still at the exploratory stage, there could be a lot of scope for future work to investigate deeply.

- More topologies can be achieved such as tree, small world and torus topology. Topology has significant effect on the global latency performance. Other topologies can also be tested and can receive better performance.

- For the evaluation framework, building a router that can support multicast could improve the performance of network. In most cases, fan-outs of a single neuron cannot be assigned to one PE. Thus, a number of copies of a spike will be sent to network, which increase the traffic load of networks. The multicast technology can significantly reduce the traffic.

- For the routing schemes, especially in mesh topology, more routing schemes should be achieved. At present, the path of transmitting spikes is relatively fixed and duplicated, which results in hotspots and high possibility of congestion. New routing schemes like dynamic routing algorithms could be a solution for congestion. Unlike static routing, dynamic routing algorithm allows routers to select proper destination instead of fixed destination.

- To achieve low-power operation, the interconnection communication systems that transport spikes between neuromorphic arrays and external interfaces will be developed as asynchronous circuits. The project can be extended to explore different asynchronous switch designs that route address events between neuromorphic arrays.

Bibliography

[1] Jennifer Walinga and Charles Stangor. Introduction to psychology-1st canadian edition. 2014.

[2] Jornt R De Gruijl, Paolo Bazzigaluppi, Marcel TG de Jeu, and Chris I De Zeeuw. Climbing fiber burst size and olivary sub-threshold oscillations in a network setting. *PLoS Comput Biol*, 8(12):e1002814, 2012.

[3] Ben Varkey Benjamin, Peiran Gao, Emmett McQuinn, Swadesh Choudhary, Anand R Chandrasekaran, Jean-Marie Bussat, Rodrigo Alvarez-Icaza, John V Arthur, Paul A Merolla, and Kwabena Boahen. Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations. *Proceedings of the IEEE*, 102(5):699–716, 2014.

[4] Paul A Merolla, John V Arthur, Rodrigo Alvarez-Icaza, Andrew S Cassidy, Jun Sawada, Filipp Akopyan, Bryan L Jackson, Nabil Imam, Chen Guo, Yutaka Nakamura, et al. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science*, 345(6197):668–673, 2014.

[5] James Aweya. *Switch/Router Architectures: Systems with Crossbar Switch Fabrics.* CRC Press, 2019.

[6] Simbrain 3.0 documentation. Accessed: 16-08-2021.

[7] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[8] Frank Rosenblatt. Perceptron simulation experiments. *Proceedings of the IRE*, 48(3):301–309, 1960.

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[10] Daniel Roggen, Stéphane Hofmann, Yann Thoma, and Dario Floreano. Hardware spiking neural network with run-time reconfigurable connectivity in an autonomous robot. In *NASA/DoD Conference on Evolvable Hardware, 2003. Proceedings.*, pages 189–198. IEEE, 2003.

[11] B Glackin, T Martin McGinnity, Liam P Maguire, QX Wu, and Ammar Belatreche. A novel approach for the implementation of large scale spiking neural networks on fpga hardware. In *International Work-Conference on Artificial Neural Networks*, pages 552–563. Springer, 2005.

[12] Filipp Akopyan, Jun Sawada, Andrew Cassidy, Rodrigo Alvarez-Icaza, John Arthur, Paul Merolla, Nabil Imam, Yutaka Nakamura, Pallab Datta, Gi-Joon

Nam, 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*, 34(10):1537–1557, 2015.

[13] Eugene M Izhikevich, Joseph A Gally, and Gerald M Edelman. Spike-timing dynamics of neuronal groups. *Cerebral cortex*, 14(8):933–944, 2004.

[14] Louis Lapique. Recherches quantitatives sur l'excitation electrique des nerfs traitee comme une polarization. *Journal of Physiology and Patholoigy*, 9:620–635, 1907.

[15] Peter Dayan and Laurence F Abbott. *Theoretical neuroscience: computational and mathematical modeling of neural systems.* Computational Neuroscience Series, 2001.

[16] Lei Deng, Yujie Wu, Xing Hu, Ling Liang, Yufei Ding, Guoqi Li, Guangshe Zhao, Peng Li, and Yuan Xie. Rethinking the performance comparison between snns and anns. *Neural Networks*, 121:294–307, 2020.

[17] Vincenzo Catania, Andrea Mineo, Salvatore Monteleone, Maurizio Palesi, and Davide Patti. Noxim: An open, extensible and cycle-accurate network on chip simulator. In *2015 IEEE 26th international conference on application-specific systems, architectures and processors (ASAP)*, pages 162–163. IEEE, 2015.

[18] Daniel Svozil, Vladimir Kvasnicka, and Jiri Pospichal. Introduction to multi-layer feed-forward neural networks. *Chemometrics and intelligent laboratory systems*, 39(1):43–62, 1997.

[19] Kwabena A Boahen. A burst-mode word-serial address-event link-i: Transmitter design. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 51(7):1269–1280, 2004.

[20] Kwabena A Boahen. A burst-mode word-serial address-event link-ii: Receiver design. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 51(7):1281–1291, 2004.

[21] Kwabena A Boahen. A burst-mode word-serial address-event link-iii: Analysis and test results. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 51(7):1292–1300, 2004.

[22] Steve B Furber, Francesco Galluppi, Steve Temple, and Luis A Plana. The spinnaker project. *Proceedings of the IEEE*, 102(5):652–665, 2014.

[23] Eustace Painkras, Luis A Plana, Jim Garside, Steve Temple, Francesco Galluppi, Cameron Patterson, David R Lester, Andrew D Brown, and Steve B Furber. Spinnaker: A 1-w 18-core system-on-chip for massively-parallel neural network simulation. *IEEE Journal of Solid-State Circuits*, 48(8):1943–1953, 2013.

[24] Saber Moradi, Ning Qiao, Fabio Stefanini, and Giacomo Indiveri. A scalable multicore architecture with heterogeneous memory structures for dynamic neuromorphic asynchronous processors (dynaps). *IEEE transactions on biomedical circuits and systems*, 12(1):106–122, 2017.

[25] Johannes Schemmel, Daniel Brüderle, Andreas Grübl, Matthias Hock, Karlheinz Meier, and Sebastian Millner. A wafer-scale neuromorphic hardware system for large-scale neural modeling. In *2010 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1947–1950. IEEE, 2010.

[26] Ning Qiao, Hesham Mostafa, Federico Corradi, Marc Osswald, Fabio Stefanini, Dora Sumislawska, and Giacomo Indiveri. A reconfigurable on-line learning spiking neuromorphic processor comprising 256 neurons and 128k synapses. *Frontiers in neuroscience*, 9:141, 2015.

[27] Érika Cota, Alexandre de Morais Amory, and Marcelo Soares Lubaszewski. *Reliability, Availability and Serviceability of Networks-on-chip*. Springer Science & Business Media, 2011.

[28] Natalie Enright Jerger, Tushar Krishna, and Li-Shiuan Peh. On-chip networks. *Synthesis Lectures on Computer Architecture*, 12(3):27–28, 2017.

[29] Duncan J Watts and Steven H Strogatz. Collective dynamics of 'small-world'networks. *nature*, 393(6684):440–442, 1998.

[30] Hang-Sheng Wang, Xinping Zhu, Li-Shiuan Peh, and Sharad Malik. Orion: A power-performance simulator for interconnection networks. In *35th Annual IEEE/ACM International Symposium on Microarchitecture, 2002.(MICRO-35). Proceedings.*, pages 294–305. IEEE, 2002.

[31] Andrew B Kahng, Bin Li, Li-Shiuan Peh, and Kambiz Samadi. Orion 2.0: A power-area simulator for interconnection networks. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 20(1):191–196, 2011.

[32] George BP Bezerra, Stephanie Forrest, Melanie Moses, Al Davis, and Payman Zarkesh-Ha. Modeling noc traffic locality and energy consumption with rent's communication probability distribution. In *Proceedings of the 12th ACM/IEEE international workshop on System level interconnect prediction*, pages 3–8, 2010.

[33] Andrei Ardelean, Amir Zjajo, Sumeet Kumar, and Rene van Leuken. Energy-efficient multipath ring network for heterogeneous clustered neuronal arrays. In *2018 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*, pages 190–193. IEEE, 2018.