Scalable Networks-on-Chip Interconnected Architecture for Astrocyte-Neuron Networks

Scalable Networks-on-Chip Interconnected Architecture for Astrocyte-Neuron Networks

Junxiu Liu, *Member, IEEE*, Jim Harkin, *Member, IEEE*, Liam P. Maguire, Liam J. McDaid, John J. Wade, and George Martin

***Abstract*—Spiking astrocyte-neuron networks (ANNs) have the potential to emulate the self-repair capability in the mammalian brain. Recent research has explored the mimicking of this capability in hardware with the aim to make electronic circuits autonomous with self-detection and repair. The provision of hardware architectures and interconnectivity between the massive numbers of spiking neurons and astrocytes is a significant research challenge, as the neuron and astrocyte networks have different communication patterns. In particular they have large volumes of information exchanges. This paper presents a novel interconnected architecture for ANN hardware systems based on the hierarchical astrocyte network architecture (HANA). HANA supports the information exchanges between astrocyte cells and addresses the interconnection challenge by providing a novel hierarchical networks-on-chip (NoC) structure of neurons and astrocytes cells. The proposed HANA incorporates a priority scheduling mechanism to increase the information exchange rate for global astrocyte cells, thus reducing the global communication latency and providing a balance between the local and global astrocyte network traffic. Experimental results demonstrate that the proposed HANA architecture can provide efficient information exchange rates for ANN, while the hardware synthesis results demonstrates that it has a low area utilization and power consumption which supports scalability.**

***Index Terms*—Spiking neural networks, astrocyte, networks-on-chip, hierarchical, self-repair, fault tolerance, FPGA.**

I. Introduction

Spiking neural network (SNN) is a promising neural network model to date as it emulates information processing and communication capabilities of the mammalian brain and provides a closer approach to modelling biological neurons than traditional artificial neural network [1]. SNNs use the timing of spikes (i.e. spike events) for communication and employ various network topologies for the connections of different neuron layers. They also update the synaptic weights (i.e. synaptic plasticity) to mimic the information processing of neurons. Additionally, it is now accepted that the interaction between spiking neurons and glial cells (especially astrocytes) offers a much richer computation paradigm [2]–[4]. For example the authors have developed a computational model in software [5], [6] and hardware [7] which captures this behaviour and has demonstrated how astrocytes cells used within spiking neurons in a spiking astrocyte-neuron network (ANN) can perform distributed and fine-grained self-repair under the presence of faults. These papers have shown that normally active neurons interacting with astrocytes through the endocannabinoid retrograde signalling messenger, can give rise to a stable and low synaptic transmission probability. In this state only a few presynaptic spikes cause the release of neurotransmitter into the synaptic cleft but because there are many synaptic inputs the postsynaptic neuron will be active after a period of learning. However, when synaptic transmission is interrupted by, for example, faults across a portion of the synapses, then this causes the activity of the postsynaptic neuron to fall and eventually become silent. In this state the retrograde messenger stops causing the probability of release at all synaptic sites to increase, which effectively re-starts the learning process causing the healthy (undamaged) input synapses to strengthen. This control of the learning process by the activity of the postsynaptic neuron is closely aligned with the BCM learning rule [8]. Furthermore, the work in [5], [6] is based on experimental evidence regarding astrocyte neuron coupling at the tripartite synapse [3], [9]–[11] and motivated by other independent experimental evidence showing a direct link between the retrograde messenger signalling and the probability of release at synaptic sites [12]. Specifically, the endocannabinoid retrograde messenger is synthesised by active postsynaptic neurons and is fed back to the presynaptic neuron by a direct pathway, which cause the probability of release at nearby synapses to decrease. The endocannabinoid messenger also causes the release of calcium from stores within astrocytes (the indirect pathway), which subsequently results in an increase of the probability of release at both locally (nearby synaptic sites) and globally. Therefore, both the direct and indirect pathways exercise control over the probability of release and this is the key biological function that the research in [5], [6] builds upon.

For a large neural network, hardware implementations offer superior execution speed compared to sequential software approaches due to the inherent parallelism of hardware. Recent research demonstrated that the hardware systems can provide information communications for large numbers of neurons in a robust and power-efficient manner [13]–[16]. However, the traditional hardware architecture (e.g. direct neuron-to-astrocyte connection) and topologies (e.g. two dimensional mesh) are not suitable for the interconnections between the

Junxiu Liu, Jim Harkin, Liam P. Maguire, Liam J. McDaid, John J. Wade, and George Martin are with the School of Computing and Intelligent Systems, University of Ulster, Magee campus, Northern Ireland, UK BT48 7JL. (e-mail: {j.liu1, jg.harkin, lp.maguire, lj.mcdaid, jj.wade, martin-g11}@ulster.ac.uk).

astrocytes and neurons, and a new interconnection strategy needs to be explored given that: (a) there are significant numbers of spiking neurons and astrocytes in the ANN. Using simple direct connections between them prohibits system scalability as one astrocyte communicates with many neurons and the direct links increase hugely with increasing numbers of astrocytes; also, astrocytes additionally communicate with one another and (b) the neurons and astrocytes have different communication patterns, e.g. high speed temporal spike event for neuron networks, low speed numerical inositol trisphosphate information exchange for the astrocyte networks. Therefore, different topologies for individual networks should be explored and each of which should meet the individual communications requirements. Existing interconnected architectures do not meet these requirements.

Hence, these aforementioned constraints make the deployment of ANN hardware systems a challenging on-chip interconnect problem, where a balance between scalability and biological real-time requirements needs to be achieved. A novel hardware interconnected architecture is required which can address these issues. Inspired by the morphology and structure of biological neural networks, this paper presents a novel hierarchical interconnected astrocyte network (HANA) architecture for the information exchange between astrocytes and neuron cells in hardware. The key aim is to address the interconnection problems of hardware ANNs using a hierarchical networks-on-chip structure. The contribution of this approach is to provide a solution which meets the requirements of communications and scalability of the individual neurons and astrocyte networks, and achieve a trade-off between the scalability, information exchange rate and power consumption for large scale ANNs. Section II provides a state of the art review on hardware interconnection strategies for neuron networks and section III presents the proposed HANA architecture. Section IV reports on the experimental results regarding the communication latency and the hardware system performance and section V provides a summary and conclusion.

II. Related Work

This section provides a brief overview of related work. The hardware implementations for SNNs are given first; then the astrocyte cell hardware implementations and its inherent self-repair capability are outlined. Finally, the motivation for investigating the interconnected architecture for ANN systems is outlined. This section only focuses on the work that has shown practical demonstration of scalability for large-scale neural network hardware implementations. A further detailed review can be found in [17], [18].

Recently, researchers started to explore the full-custom hardware design for neural network implementations. The US DARPA-funded SyNAPSE program developed neuromorphic systems to mimic the mammalian brain in electronic hardware. Several neuromorphic chips have been designed to capture the functional properties of specific brain regions [19], [20], e.g. the TrueNorth chips [15], [21] were designed by the IBM brain lab which can accommodate 1 million digital neurons and 256 million synapses, and have the potential to propel neuromorphic computing to real-world applications, e.g. image and voice recognitions. The NeuroGrid project [22] developed a mixed-analog-digital chip that can simulate the biological neural systems in real time. It realized all electronic circuits (except the axonal arbours) in analog to maximize energy efficiency. Thus, using only 16 neurocores can simulate a million neurons and billions of synaptic connections in real time, with a low power consumption of three watts. Recently, the Networks-on-Chip (NoC) technique is now employed for the interconnection between neurons [23]. The NoC is composed of routers, channels and processing elements and the concept is similar to computer networks where packets of information are forwarded via paths between the source and destination processing elements. The SpiNNaker project [13] used NoC for the interconnections between neurons where the CPU processors are the neuron process units (i.e. ARM9 cores). It consisted of an array of ARM9 cores with the goal of simulating up to a billion neurons in real time (biological time scale). The EMBRACE project (e.g. [16], [24]–[26]) also used the NoC as an interconnection fabric to achieve a trade-off between the scalability, throughput, neuron/synapse ratio and power consumption for the large scale SNN implementations, in particular a novel hierarchical architecture H-NoC [16], [27] was proposed to address the scalability issue of SNN hardware system. The aforementioned approaches only investigated hardware architectures for SNNs where the main components are neurons and synapses.

However, the astrocyte cell is also an important contributor in metabolic maintenance in the neural network and very much involved in neuronal activity and information processing in the nervous systems [28]. Astrocytes and their interactions between neurons underpin the brain's distributed and fine-grained repair capability [2], [4]. Some progress has been made in implementing astrocyte cells in hardware where the aim is to explore basic spiking neural network behaviour such as synchronisation. For example, an Izhikevich model [29] and a FitzHugh Nagumo model [30] were used to describe the behaviours of neurons (e.g. spiking and bursting activities) in the digital circuits [31], [32]; and the behaviours of astrocyte cells (e.g. the intracellular calcium waves) were designed in hardware in research works of [33]–[35]. These studies demonstrated that the neuron-astrocyte designs on hardware provided a possible solution for the development of the fundamental computational unit of a digital astrocyte. Using the same models in [31], the digital implementation of neuron-astrocyte signalling was further optimised [36] using single constant multiply and linear approximation techniques. Current research to date has only implemented the function of astrocyte cells in hardware, and no research has explored the interconnectivity requirements between spiking neurons and astrocytes in hardware. To the best of the authors' knowledge, this is the first time that a novel interconnected hardware architecture for spiking neurons and astrocyte cells is proposed. The next section introduces the proposed hardware architecture in detail.

III. Hierarchical Astrocyte-Neuron Network Architecture

This section presents an overview of the proposed HANA for spiking astrocyte-neuron networks. The astrocyte-neural network model is first presented briefly; then the internal structure of architecture is discussed and the information exchange and communication strategies between the astrocyte cells are presented. In addition, the proposed interconnected architecture employs a priority scheduling mechanism to balance the traffic load of the entire astrocyte-neuron network, which is also discussed in this section.

A. Astrocyte-Neural Network Model

In this work, the astrocyte cell is based on the astrocyte model originally developed by De Pitta et al. [37]. The interaction between the astrocyte cells and neurons is described as follows. Recent experimental research showed that ~50% of synapses are composed of an intimate connection between an astrocyte and several neurons, i.e. a synapse actually exchanges signals at three terminals, known as the tripartite synapse [38]. In a tripartite synapse, when an action potential arrives neurotransmitter (glutamate) is released across the synapse cleft and binds to receptors on the post-synaptic dendrite, causing a depolarization of the post-synaptic neuron. When the post-synaptic neuron is sufficiently depolarized, voltage gated calcium channels on the dendrite allow the influx of Ca^{2+} into the dendrite causing endocannabinoids to be synthesized and subsequently released from the dendrite. It is known that endocannabinoids are a type of retrograde messenger which travel back from the post-synaptic terminal to the pre-synaptic terminal. In addition, the release of 2-arachidonyl glycerol (2-AG), a type of endocannabinoid, is known to bind to type 1 Cannabinoid Receptors (CB1Rs) on an astrocyte, which enwraps the synapse. This increases inositol trisphosphate (IP_3) levels within the astrocyte and triggers a transient intracellular release of Ca^{2+}. When a post synaptic neuron fires, 2-AG is released which is given by:

$$\frac{d(AG)}{dt} = \frac{-AG}{\tau_{AG}} + r_{AG}\delta(t - t_{sp}) \tag{1}$$

where AG is the quantity of the released 2-AG; τ_{AG} and r_{AG} are the decay and production rates of 2-AG respectively; t_{sp} is the time of the post-synaptic spike. When the 2-AG binds to CB1Rs on the astrocyte, IP_3 is generated which is dependent on the amount of released 2-AG and is given by:

$$\frac{d(IP_3)}{dt} = \frac{IP_3^* - IP_3}{\tau_{ip_3}} + r_{ip_3}AG \tag{2}$$

where IP_3 is the quantity within the cytoplasm, IP_3^* is the baseline of IP_3 when the cell is in a steady state and receiving no input, τ_{ip_3} and r_{ip_3} is the decay and production rate of IP_3 respectively.

In addition, the astrocyte cells are also linked together where the boundary conditions are imposed, as described in the approach of [39]. Each astrocyte cell is connected to the nearest neighbour using molecular gap junctions which facilitate astrocyte to astrocyte communications (i.e. calcium pulses/waves) over long distances. The propagating calcium pulses are elicited following the gap-junction transfer of IP_3 second messenger molecules [39]. In this approach, a linear diffusion gap junction model is considered to describe the exchange of IP_3 between any two astrocyte [39] as

$$J_{i\to j} = F\Delta_{ij}IP_3 \tag{3}$$

where

$$\Delta_{ij}IP_3 = IP_3^i - IP_3^j, i, j \in [1, m] \tag{4}$$

and the coupling strength (or permeability) F depends on the number of gap junction channels and their unitary permeability. The detailed astrocyte-neural network dynamics are described in our previous works of [5], [6], [40].

B. Interconnected architecture for spiking astrocyte-neuron networks

In our previous work [16], a hierarchical NoC architecture (H-NoC) was designed for the SNN hardware system, which is shown in Fig. 1(a). The H-NoC [16] implemented the connections for clusters of neurons and permitted large-scale interconnectivity with balanced delays. It has three facilities at different levels – *neuron*, *tile* and *cluster*. The *neuron facility* is at the bottom level, which connects a number of neurons together (e.g. ten in [16]) using the star connection topology. Spike events are generated by the neuron cells which are located in these *neuron facilities* and then they are packetized and issued to the NoC. The packets are then received by a router with the targeted neuron cells, where spike processing is done. The *neuron facilities* are connected to a tile router to compose a *tile facility* (second layer), shown in the middle of Fig. 1(a). Several *tile facilities* are connected by a cluster router to compose a *cluster facility* (i.e. top layer). As a result, a group of 10 neuron cells is connected in a single *neuron facility*; 10 *neuron facilities* are gathered together in each *tile facility*; and 4 *tile facilities* are placed in a single *cluster facility*. Therefore, each cluster can accommodate a total of 400 neural cells [16]. Furthermore, the *cluster facility* can be also replicated to form a grid of clusters using a mesh topology which has the potential capability to implement a large scale SNN hardware system. Table I summarizes the architecture parameters of H-NoC.

The H-NoC architecture in Fig. 1(a) addresses the scalability issue of large scale SNNs by creating a modular array of

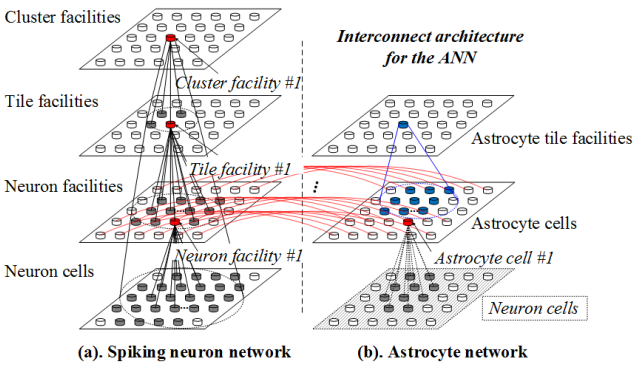

Fig. 1. The interconnected architecture diagram for the ANN. (a) H-NoC for the SNN. (b) The interconnection architecture for the astrocyte networks.

clusters of neurons using a hierarchical structure of low (i.e. neuron) and high (i.e. cluster) level NoC routers [16]. However, H-NoC was only designed for the interconnection between neurons in SNN hardware. As discussed in previous sections, the astrocytes communicate with synapses/neurons, and more importantly with other astrocytes as well. The basic astrocyte-neuron interaction mechanism or types of data exchanged between them was defined in software and hardware by the authors' previous work of [5], [7], [40]. Therefore, this paper focuses on the interconnected architecture for the ANN, in particular the interconnection strategies for the astrocyte networks.

TABLE I. THE ARCHITECTURE PARAMETERS OF H-NOC AND HANA

Layer level	The approach					
	H-NoC [16]			HANA		
	Facility	Connects to	Topology	Facility	Connects to	Topology
High	Cluster	4 Tile facilities	Mesh	Astrocyte tile	10 astrocyte cell facilities	Mesh
↑	Tile	10 neuron facilities	Star*			
Low	Neuron	10 neuron cells		Astrocyte cell	1 neuron facility	Ring

* Tile and neuron facilities connect to the nodes in next layer via star topology.

Fig. 1 presents the interconnected architecture diagram for the ANN. The H-NoC architecture for the spiking neuron network is illustrated in Fig. 1(a) which clearly shows the hierarchical structure. According to [41], the astrocyte can enwrap up to $\sim 10^5$ synapses and contact ~6 neurons within the cortex and hippocampus. In order to establish the interconnection between the neuron cells and the astrocyte cell, a dedicated connection between them (i.e. the red wire connections in Fig. 1(b)) is created, e.g. the neuron facility #1 of H-NoC is connected to astrocyte cell #1. Using this connection, the astrocyte cell can communicate with a group of neurons (up to 10 neurons in this approach) via the node router inside the neuron facility (e.g. the neuron facility #1), therefore the interconnection between the astrocyte and neuron cells are established which are equal to the virtual direct connections shown by the right bottom part of Fig. 1(b). More detail about the data exchanged between astrocyte and neurons can be found in our previous work [5], [7], [40].

Fig. 1(b) provides an overview of the interconnected architecture for the astrocyte networks, but more detail on the internal hierarchical astrocyte network architecture (HANA) is provided by Fig. 2. It can be seen that HANA has two layers – *astrocyte* ***cell*** and ***tile*** *facilities*. The HANA approach exploits locality between *astrocyte cells*, by allocating a group of *m astrocyte cells* (e.g. 10 in this approach) together which is located at the bottom of the hierarchy. The *astrocyte cells* connect to neurons cells as discussed previously, which is shown by at the bottom right of Fig. 2. In each *astrocyte cell* group, the *astrocyte cells* are connected to an *astrocyte hub* using a ring topology. The NoC interconnection technique is employed in this approach and provides various information exchange strategies (e.g. broadcast, point-to-point etc.) between the astrocyte cells. The *astrocyte hub* is then connected to a higher level router (i.e. the *astrocyte tile router*) to comprise an *astrocyte tile facility*. The *astrocyte tile facilities* are connected by a two dimensional mesh topology which provides the communication for the astrocyte cells in different *astrocyte cell* groups. Table I summarizes the architecture parameters of HANA where a two-layer interconnection topology is used, i.e. the *astrocyte cell* group for the local astrocyte connectivity, and the *astrocyte tile facility* for global astrocyte connectivity. As a result, one *astrocyte tile facility* includes 10 *astrocyte cells*, where each *astrocyte cell* communicates with 10 *neurons*; therefore one *astrocyte tile facility* can accommodate 10 *astrocyte cells* and 100 *neuron cells*. If more *astrocyte tile facilities* are required, the *astrocyte tile facility* is simply replicated by forming a grid of *astrocyte tiles* using the mesh topology. Hence, a mesh can be realized with a number of *astrocyte tile facilities*, "*T*", allowing the total number of astrocyte/neuron cells in the HANA to scale by a linear factor of "*T*".

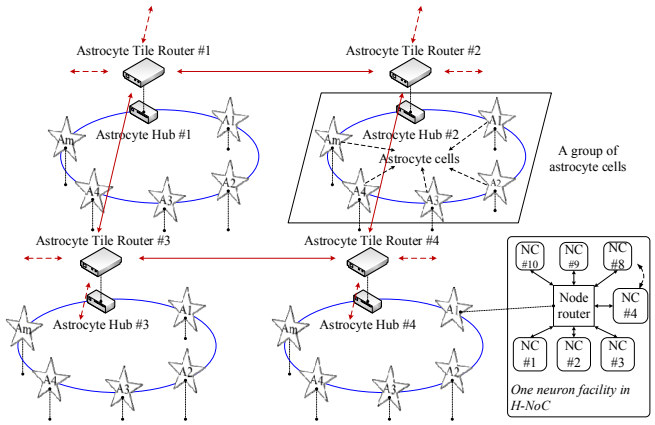

Fig. 2. The HANA diagram for the astrocyte network.

Based on the structure shown in Fig. 2, the scalability of HANA is analysed based on two key factors, i.e. area/power footprint and NoC data throughput. In terms of the area/power footprint, the proposed HANA affords a small overhead due to the ring topology interconnect for the astrocyte facilities. As the degree of the ring topology (i.e. the number of channels connected to one node) is much smaller than other topologies (e.g. torus, star, tree etc.), the required hardware resources for the implementation of the ring topology is much lower (see section IV, for the analytic results). For the second layer of astrocyte tile facilities, the 2D mesh topology makes the HANA more easily implemented and physically scalable in hardware. In addition, in terms of throughput, the proposed HANA shows a fixed path delay for both global and local astrocyte cell communications. Therefore, HANA allows the scale up of the entire spiking astrocyte-neuron network. Analysis and experimental results are provided in section IV. Note that inside the astrocyte cell group, although the selected ring topology does not provide a good path diversity, i.e. the alternative communication path is limited, it is an efficient method (i.e. low area overhead) to connect the astrocyte cells together. In addition, contrary to the high speed temporal communications of the spiking neuron cells, astrocytes communicate with each other at a much slower rates (i.e. typical ~10Hz according to the

approach of [39]) therefore the sequential pathways of the ring topology can exploit this slow data rate.

C. HANA uniform communication mechanism

Inside the astrocyte tile facility, the astrocyte cells are modelled as being linked together in a ring structure [6]. The astrocyte to astrocyte communications over long distances use slow changing calcium waves and the IP_3 exchange is the key process as described in (4). The HANA employs the following communication protocol and hardware facilities to provide an efficient IP_3 exchange mechanism.

- Communication protocols: a token ring technique is used in this work which defines two different packet types (e.g. token/data) to manage the packet forwarding mechanism. Additionally, a packet priority scheduling mechanism is proposed in section III.D, which defines different levels for node (P_N), packet (P_T) and appointed (P_A) priorities, to balance the local/global traffics.
- Hardware facilities: a two-layer interconnection architecture is employed in this work including the astrocyte cells and astrocyte tile facilities. The overall diagram is shown in section III.B, and the detailed structures are discussed in sections III.E and III.F.

The rest of this subsection discusses the communication patterns in HANA and defines the uniform packet layout used in the token ring technique. Equation (4) demonstrates that the exchange process of IP_3 occurs in every two astrocyte cells, therefore the broadcast propagation is the most appropriate communication mode for the astrocyte network. The communication patterns of the astrocyte network are illustrated in Fig. 3. The broadcast propagation inside the same astrocyte tile facility is illustrated by Fig. 3(a) where the source astrocyte cell in blue sends information to all others coloured in green. Beside this communication pattern, the astrocyte cell can also broadcast to astrocytes in another astrocyte tile facility (see Fig. 3(b)) via the astrocyte tile router, or communicate to another specific astrocyte cell inside (see Fig. 3(c)) or outside (see Fig. 3(d)) current astrocyte facility by point-to-point connection. Fig. 3 clearly shows that the astrocyte hub and astrocyte tile router are the connection bridge for the different astrocyte tile facilities, which enables the astrocyte to astrocyte cell communications over long distances.

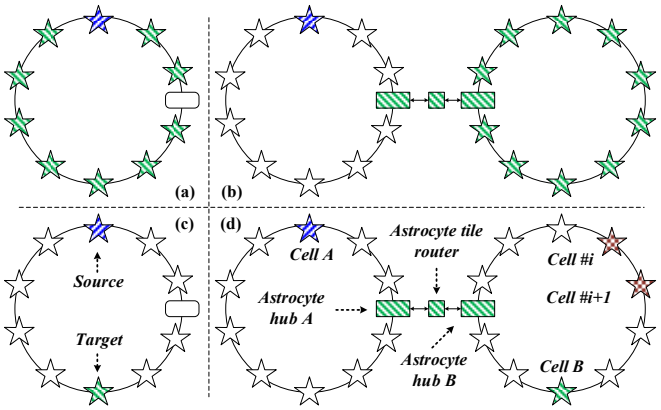

Fig. 3. The communication patterns in the astrocyte network. (a) Broadcast inside the tile. (b) Outside broadcast to astrocyte tile. (c) Point-to-Point connection inside the tile facility. (d) Point-to-Point to the cell outside the astrocyte tile facility.

In the astrocyte tile facility, as the communication channels are shared by multiple astrocyte cells, the token ring technique is employed for the local communication access protocol. A token packet travels around the ring topology and only when the astrocyte cell gets the token, does it create a communication session, e.g. transmits packets to other cells; otherwise it can only 'listen' and receive packets. For example, in the broadcast communication pattern shown in Fig. 3(a), the source astrocyte cell in blue gets the token first, it then sends the packet to the next astrocyte cell. When next astrocyte cell receives the packet, it copies the packet and then forwards to the next one. Therefore, this packet travels along all the astrocyte cells within the ring topology. When this packet travels back to the source astrocyte cell, the source astrocyte cell realizes that the packet has completed the ring traversal. This channel access control mechanism provides fair access for all the astrocyte cells, and eliminates collisions due to the multiple access requests.

HANA uses a uniform packet layout for the local and global communications. The packet layout is defined in Table II where two different packet types are used, i.e. token and data packets. The token packet consists of four fields, i.e. header, access control, astrocyte cell address, and reserved field. The data packet has two additional fields of tile router address and payload. The header field defines the packet type and

TABLE II. PACKET LAYOUT DEFINITION

Token Packet							
Header	Access control		Astrocyte cell address		Reserved		
	P_T	P_A					
4-bit	2-bit	2-bit	4-bit		x-bit		
Data Packet							
Header	Access control		Tile router address		Astrocyte cell address		Payload
	P_T	P_A	X	Y	Source	Destination	
4-bit	2-bit	2-bit	4-bit	4-bit	4-bit	4-bit	x-bit
Details							
Header	*Packet type*						
0001	Data packet - Broadcast communication inside the astrocyte tile facility						
0010	Data packet - Point-to-Point communication inside the astrocyte tile facility						
0011	Data packet - Broadcast communication from/to the astrocyte tile facility outside						
0100	Data packet - Point-to-Point communication from/to the astrocyte tile facility outside						
….	Reserved						
1111	Token packet						

P_T: Packet priority. P_A: The appointed priority

communication patterns as given by the bottom of Table II, e.g. whether it's a token or data packet, for broadcast or P2P communications, connecting to the local or global astrocytes. The tile router/astrocyte cell address fields provide the address information of the astrocyte tile router and astrocyte cell respectively. The payload field of the data packet includes the exchange information (e.g. IP_3 as described in (4)) between the astrocyte cells. As different data types (e.g. float, double, or fixed-point data) can be used to represent the value of IP_3, an x-bit width is used in the payload field. More discussion about this data width is given in section IV. The most important part in the token/data packets is the access control field. It includes a 2-bit packet priority (P_T) and a 2-bit appointed priority (P_A). As the astrocyte cell typically communicates with other cell(s) over long distances, e.g. the global connections between two distant astrocyte facilities. Packets from distant astrocyte cells are given a higher priority to allow them to promptly access the token ring topology. Therefore, the packet priority scheduling mechanism is proposed to provide a method for scheduling the local and global traffic accesses (flagged using the access control field of the packets).

D. Packet priority scheduling mechanism

HANA employs a packet priority scheduling mechanism to balance the local and global traffic throughput of the astrocyte cells. In the astrocyte tile facility, each node (e.g. astrocyte hub or cell) is assigned a node priority (denoted by P_N). When the astrocyte hub or cell receives the packet, it compares the packet priority (i.e. P_T in the access control field of packets) with their node priorities, P_N. If $P_N \geq P_T$, the node can receive and process the packets; otherwise, it forwards the packets to the next node directly without processing. The packet priority scheduling is a two-level priority based mechanism. If a global data transmission request from astrocyte tile router is received by the astrocyte hub (i.e. the astrocyte cell(s) outside attempt to exchange IP_3 with local cell(s) over a long distance), this request will be processed promptly. Therefore, the astrocyte hub is assigned with a high P_N and other astrocyte cells are assigned with a lower P_N. The astrocyte hub is the only node in an astrocyte tile facility which can appoint the priority level.

The proposed packet priority scheduling mechanism is illustrated in Fig. 4 and based on the example in Fig. 3(d). In the astrocyte networks of Fig. 3(d), two astrocyte tile facilities *A* and *B* are connected by an astrocyte tile router. Assume that at the beginning astrocyte cell $\#i$ in astrocyte tile facility *B* has the token at time step 1 (i.e. T1 in Fig. 4). Therefore, it has the permission to create a communication session and transmit data packets to other cells (T2 in Fig. 4). The data packets are forwarded one by one in the *B* astrocyte tile facility. At this time, also assume that the astrocyte hub *B* in Fig. 3(d) receives a global data access request from astrocyte tile router (T3), e.g. the astrocyte cell *A* in astrocyte tile facility *A* sends a packet to cell *B* in facility *B*;(this is a global information exchange request, it should be given the priority to access the channel as quickly as possible). Therefore when the astrocyte hub *B* receives the data packet from astrocyte cell $\#i$ (T4), it appoints the access priority by setting the P_A high in the access control

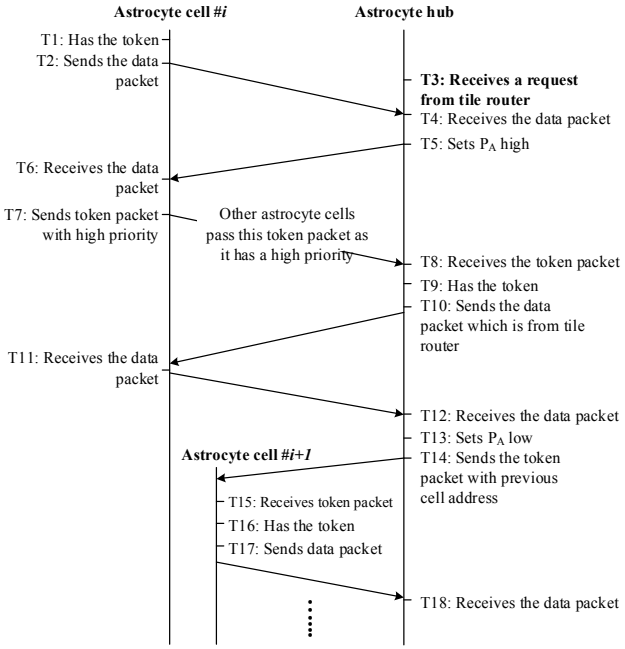

Fig. 4. Packet priority scheduling mechanism of HANA.

field of the received data packet (T5); then when the cell $\#i$ receives the sent data packet by itself, it discards this packet and this communication session is completed (T6). At the same time, it is aware that the astrocyte hub has a global data transmission request waiting as the P_A is high, so it sets the token packet priority (i.e. P_T) to be high (T7) and sends the token packet out. As all other astrocyte cells have a low node priority P_N, they forward this token packet to the astrocyte hub directly. When the astrocyte hub receives the token packet (T8), it gets permission (T9) to transmit the global data packet (T10). This data packet travels around the local astrocyte tile facility (e.g. T11), and is received by the target cell *B*. Then it goes back to the astrocyte hub (T12), this communication session initiated by the astrocyte hub is completed. The appointed priority is set low (T13) and the token packet is sent out with the previous address (T14, e.g. cell $\#i+1$ in Fig. 3(d)). Similar processes (T15-T18) are repeated for the next astrocyte cell node.

The packet priority scheduling mechanism can be summarized briefly as follows: when the astrocyte hub receives a global data transmission request from a astrocyte tile router, it appoints the access priority in the current communication session of astrocyte cell $\#i$. Then, the astrocyte hub will get the token in the next communication session and respond to the global access request by forwarding the data packet to the target cells promptly. After the global data transmission is complete, the astrocyte cell $\#i+1$ gets the token and then has permission to forward the data packets. The benefits of using the proposed packet priority scheduling mechanism lies in the waiting time of the global data, as it is significantly decreased which can balance the traffic loads from the global astrocyte cells (i.e. a high packet delay due to the long distance communication) and the local cells (i.e. a low packet delay). More analysis is provided in the section IV.

E. *Astrocyte cell nodes*

The astrocyte tile facilities in HANA consists of three modules: 1) astrocyte cell nodes, 2) astrocyte hub, and 3) astrocyte tile router. The astrocyte cell node is the fundamental component in the astrocyte facility. The internal structure is shown in Fig. 5(a) which includes an astrocyte cell and a network controller. The astrocyte cells mainly exchange IP_3 with each other, therefore an astrocyte cell network controller is employed to convert the IP_3 from an astrocyte cell into a NoC data packet, and also to transmit the data packet to other nodes in the astrocyte tile facility. It can be seen in Fig. 5(a) that the astrocyte cell network controller has two pairs of input and output ports – one is for the data exchange with the local astrocyte cells, and the other is for the communication with other nodes in the astrocyte tile facility. It has several components inside, e.g. an interface, two FIFOs, and one controller. The interface converts the IP_3 value to the standard data packet according to the packet layout in Table II. Two FIFOs store the temporary data packets from the astrocyte cell and other astrocyte cell nodes. The controller is the key component which manages the work flow of the astrocyte cell node.

The work flow of the astrocyte cell network controller is defined in Fig. 5(b). A group of m astrocyte cells are gathered inside the astrocyte tile facility. Only one astrocyte cell needs to generate a token packet, as the token travels around all the astrocyte cells and is used in a round robin manner. For the astrocyte cell with token generation, after the reset and token is generated, the astrocyte cell network controller receives a packet and is in state S1 (i.e. packet type judge). If the packet is token-based then the controller is in state S2. For the token packet, if $P_N \geq P_T$, $P_N \geq P_A$ and $Addr_T == Addr_N$ (i.e. the token packet address is equal to current astrocyte cell node address), this astrocyte cell node gets the token successfully and the controller moves into state S3 (i.e. collects data from the local astrocyte cell). If the local astrocyte cell has data (e.g. IP_3) for transmission, then a data packet is generated and sent to the output port (state S4). After forwarding the data packet, the controller goes to state S5 (i.e. wait for receiving packets). In this state after a packet is received, the controller goes to state S1 again; otherwise it stays in S5. In addition, if the conditions in state S2 are not met or the local astrocyte cell does not have data for transmission in state S3, the controller will send out the token packet directly to the next astrocyte cell node (i.e. state S7).

The received packet in state S1 can also be a data packet. In this case, the controller goes to state S8 and compares P_N with P_T. If $P_N < P_T$, i.e. the current astrocyte cell node is not allowed to receive this packet, the controller will forward the received packet directly without processing (state S12), and then go to state S5. If $P_N \geq P_T$, the controller processes the received data packet according to its communication pattern, as shown in Fig. 3:

a). If the data packet is a broadcast packet (state S9) and $Addr_N! = Addr_S$, the controller will send the data packet to the local astrocyte cell (state S11) and forward the received data packet (S12). However, if $Addr_N == Addr_S$, i.e. the received data packet is the one that the current cell node sent out previously, then the communication session initiated by the current astrocyte cell node is completed (ring propagation done); the token packet address is incremented by one (state S6) and the token packet is sent out again (S7).

b). If the received packet is a P2P data packet and the destination address of the packet is equal to the current cell node address ($Addr_N == Addr_D$), then the controller is in state S13, i.e. the current node receives a P2P data packet pointing to itself. The controller will send the packet to the local astrocyte cell (state S11) and forward the received packet (state S12).

c). If the received packet is a P2P data packet and the source address of the packet is equal to the current cell node address ($Addr_N == Addr_S$), then the controller moves to state S14, i.e. the current node receives the P2P data packet initiated by itself. Thus, the current communication session is completed and the controller goes to state S6 and increments the token packet address and sends it out (state S7).

In the workflow of the astrocyte cell network controller, note that a). In state S6, if a packet priority is appointed (i.e. $P_A \geq P_T$), then the token packet priority (P_T) is set to be the appointed

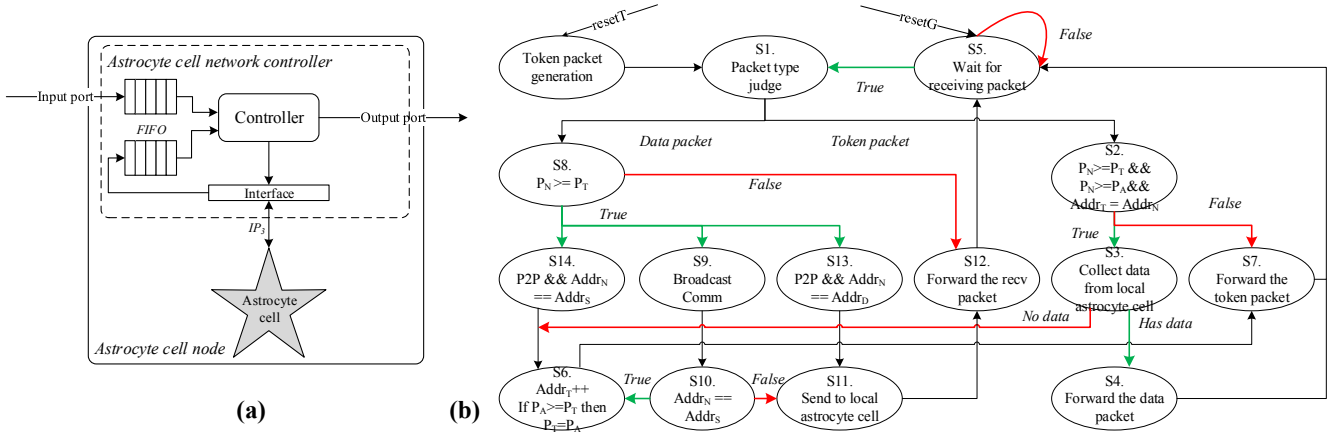

Fig. 5. (a). Astrocyte cell node diagram. (b). Workflow of astrocyte cell network controller ($Addr_T$ is the token packet address in the data packet, $Addr_N$ is the astrocyte cell node address, $Addr_S$ is the source, and $Addr_D$ is the destination. The signal of 'resetT' and block of 'Token packet generation' are only for the node which initially generates the token packet).

priority (P_A). Based on the packet priority scheduling mechanism in section III.D the token packet with the appointed priority is only processed by the astrocyte hub; therefore the astrocyte hub will get the token in the next communication session which guarantees that the global traffic can be transmitted promptly; b) for one astrocyte facility, only one astrocyte cell needs to generate the token packet and its reset signal is labelled by '*resetT*'. The astrocyte cell network controller of the other cells go to State S5 directly after reset, i.e. the '*resetG*' signal in Fig. 5(b).

F. Astrocyte hub and tile router

The astrocyte hub is another component in the astrocyte tile facility. It is the communication bridge between the local astrocyte cell nodes and the global astrocyte tile facilities. For the local traffic, it just forwards the packets to the neighbouring nodes. For the global traffic, it sends the local data to another astrocyte tile facility via the astrocyte tile router, and also receives the global data from the astrocyte tile router and forwards it to the local astrocyte cell nodes. Fig. 6(a) shows the structure of the astrocyte hub. It has two pairs of input and output ports – one is for the data exchange with the local astrocyte cell nodes in the same astrocyte tile facility, and the other is for communication with the astrocyte tile router itself. Its internal components have two FIFOs, an astrocyte tile router interface, a priority setting component and a controller. The FIFOs are used for storing packets temporarily. The astrocyte tile router interface converts the local packets to the global packet format. The priority setting component manages the priorities for the local and global packets, e.g. the global packet from the tile router has a high priority for transmission inside the astrocyte tile facility. The controller is the key component which manages the work flow of the astrocyte hub.

Fig. 6(b) presents the work flow of the astrocyte hub. After reset, the astrocyte hub is in state S1 i.e. waiting for receiving packets. If a packet is received, it goes to state S2 (i.e. packet type judge). For the token packet (when conditions $P_N > P_T$ and $Addr_T == Addr_N$) it goes to state S3, i.e. the token packet is for the astrocyte hub and has permission (a high priority) to process it. Then, the astrocyte hub attempts to collect the data from the tile router (state S4). If there is a request from the astrocyte tile router, the corresponding packet will be forwarded (state S5); otherwise the token packet containing the next node address will be sent out (states S6 and S7). After forwarding the data or token packet, the astrocyte hub returns to state S1 and waits for packets. In addition, if the received token packet in state S2 has a higher priority, e.g. $P_T == P_N$ (P_N of the astrocyte hub is higher than the astrocyte cell nodes as described previously), the astrocyte hub will move to state S8, collects the data from the astrocyte tile router (state S4), and forwards the data packet (state S5) or the token packet (states S6 and S7).

If the received packet in state S2 is a data packet, the astrocyte hub goes to state S9. For the global broadcast data packets (i.e. when header = 0011_2) in state S10, if $Addr_N == Addr_S$ (i.e. this data packet is sent by the astrocyte hub itself and the current communication session is completed), then the astrocyte hub sends out the token packet (states of S6 and S7) and returns to state S1. Otherwise, this data packet is for the astrocyte cells in another astrocyte tile facility, therefore it is sent to the astrocyte tile router (states S12 and S13). For the P2P data packet in state S9, if $Addr_N == Addr_D$, i.e. the destination address of the packet is equal to the current astrocyte hub address, then the hub is in state S14 and sends the packet to the astrocyte tile router (states S12 and S13) and returns to state S1. However, if $Addr_N == Addr_S$, i.e. the source address of the packet is equal to the current astrocyte hub address, then the hub is in state S15, i.e. the hub receives a P2P data packet initiated by itself. Thus, the current communication session is completed and the hub increments the token packet address (S6), sends it out (S7), and returns to default state S1.

The key states for the priority scheduling are highlighted in grey colour in Fig. 6(b). While the astrocyte hub forwards the received data packets, if there is a request from the astrocyte tile router, the astrocyte hub appoints the priority in state S13 by setting the P_A of the packets high. In the next step, the astrocyte cell node increases the priority of the token packet. At that point the astrocyte hub can get the token (state S8), respond to the request from the tile router (S4), and forward the packet promptly (S5). After the communication session is complete, the priority of the token packet is reset to a default value (S6) and the token is passed to the next cell node of the previous communication session (S7). More details about the packet priority scheduling mechanism between the astrocyte cell

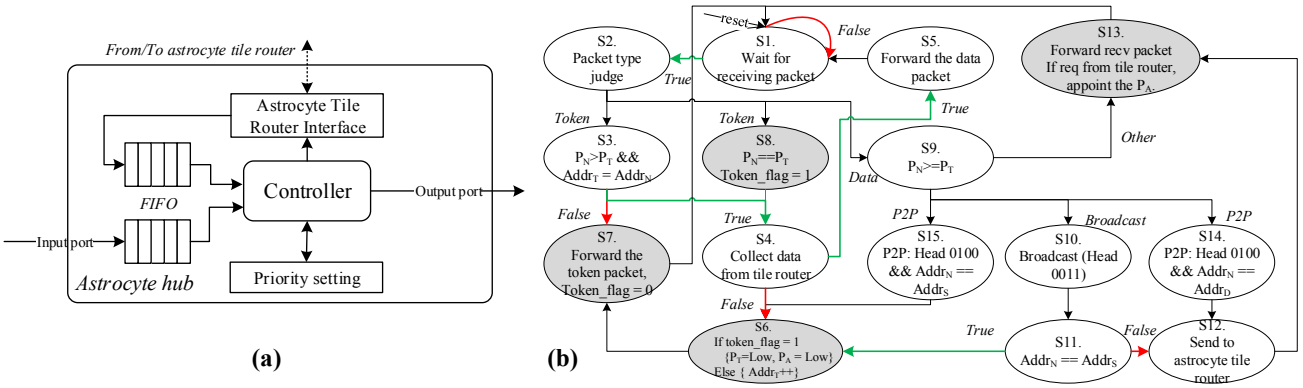

Fig. 6. (a). Astrocyte hub diagram. (b). Workflow of astrocyte hub. States coloured grey are the key processes for priority scheduling.

nodes and astrocyte hub is described in section III.D.

The top layer in HANA is the astrocyte tile router which connects the astrocyte tile facilities together. The adaptive NoC router in our previous work [24] is used as the astrocyte tile router. It has four ports to facilitate north, east, south and west inter-tile router connectivity, and a fifth network interface (NI) port. The NI port provides the connections between the astrocyte tile router and the local astrocyte hub. The astrocyte tile router employs an adaptive arbitration policy module combing the fairness of the round-robin and the priority schedule scheme of first-come-first-server approach. This improves the router throughput according to the traffic behaviour presented across the astrocyte network. Additionally, it also provides an adaptive routing decision module which facilities router adaption according to the traffic status via the channel congestion detector modules. This aids in minimising traffic congestion.

IV. Experiment Results

This section presents the test bench setup and the data traffic exchange rate analysis for the local and global astrocyte cells in HANA. A set of equations are derived to express the packet delays and analyse the system scalability. The proposed HANA is also evaluated in hardware regarding area utilization and power consumption for varied payload sizes. A performance comparison with other approaches is also provided.

A. Testbench setup

A 2x3 proof-of-concept array of astrocyte tile facilities of HANA has been described in VHDL and evaluated based on simulations. Fig. 7(a) shows the ring connections between astrocyte cells in a tile. Fig. 7(b) shows the evaluation platform for HANA, where the connections of the astrocyte hub in dashed lines are channels from/to the astrocyte cells. The 2x3 array of the astrocyte tile facilities has also been implemented and tested on an Xilinx Virtex-7 XC7VX485T-2FFG1761C FPGA to verify its real-time hardware performance [42]. All simulations were evaluated using Modelsim and the system frequency was 200 MHz in the setup. Moreover, to analyse the area utilization and the power consumption, the VHDL descriptions of a single astrocyte facility were synthesized based on a SAED 90-nm CMOS technology [43].

B. Analysis of the IP_3 exchange rate (traffic analysis)

The IP_3 exchange rate gives a measure of how fast the astrocyte cells can exchange information with each other. In this approach, the analysis of IP_3 exchange rate is performed in two stages, namely within an astrocyte tile (intra) and between astrocyte tile facilities (inter) to demonstrate the throughput performance advantage of HANA. The first stage uses one astrocyte tile facility, i.e. astrocyte tile facility [1, 1] shown in Fig. 7(b), to analyse the exchange rate inside a tile facility. The communication and information exchange inside the astrocyte tile facility is the most common communication pattern in HANA, as discussed previously in section III.C. The time duration of one communication session is used to denote how long one astrocyte cell needs to complete the IP_3 exchange with all other cell nodes (i.e. One-to-All). This duration depends on the number of astrocyte cell nodes in an astrocyte tile facility. Fig. 8 shows the results of time duration of one complete communication session T_{CS} (line format) and the IP_3 exchange rate r (bar format) inside an astrocyte tile facility. When the number of astrocyte cells, m, increases, the T_{CS} value increases, e.g. from 45 ns ($m = 1$) to 225 ns ($m = 10$). As expected with the ring topology, T_{CS} increases linearly with m. Additionally, for one round of IP_3 exchanges between any two astrocyte cells, all m astrocyte cells in a tile are required to complete their individual communication sessions, where each session takes T_{CS} for completion. Thus, the IP_3 exchange rate (r) has an exponential decrease when m increases, e.g. from 22 MHz ($m = 1$) to 444 kHz ($m = 10$), as shown in Fig. 8. Typically, biological astrocyte cells exchange information at around 10 Hz [39]. Therefore, IP_3 exchange rates are orders of magnitude faster than biology, which support hardware accelerations. In this regard, a single astrocyte tile facility is able to provide high information exchange rates, i.e. between 444 kHz to 22 MHz. Note that the number of accommodated astrocyte cells (m) can be further increased and the timing of biological information exchange can still be maintained. In this approach, the value of m is chosen as ten according to the previous computational model of the astrocyte network [6].

The second stage uses two astrocyte tile facilities, e.g. astrocyte tile facility [1, 1] and [2, 1] of Fig. 7(b) to analyse the IP_3 exchange rate (i.e. r) between them. When the astrocyte cell node sends a packet to the cells in another astrocyte tile facility (i.e. One-to-All), the IP_3 exchange rate is determined

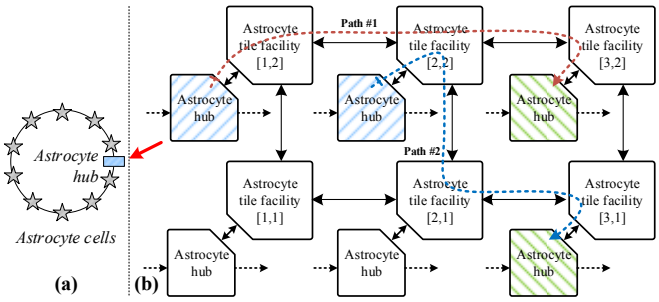

Fig. 7. Evaluation platform of HANA using a 2x3 array of astrocyte tile facilities. (a) Connection and communication inside the astrocyte tile. (b) Communication between different astrocyte tile facilities.

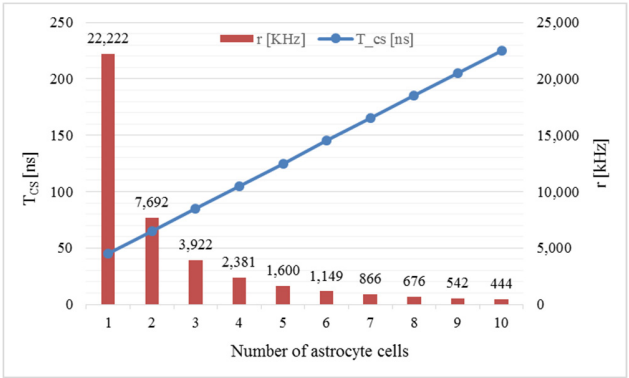

Fig. 8. Time duration of one communication session T_{CS} (line) and IP_3 exchange rate r (bars) inside an astrocyte tile facility.

by not only the packet transmission time between the two facilities, but also the waiting time of the astrocyte hub in the target astrocyte tile facility which is looking to forward packets to the destination cells. The waiting time depends on how long the astrocyte hub takes to get the token as the bottom layer of HANA uses a token-based NoC, and the node only has permission to send packets after it receives the token. In this approach, the proposed priority scheduling mechanism is employed to minimise the waiting time. After the current communication session is completed (i.e. during which the astrocyte hub receives the packet from an astrocyte tile router), the astrocyte hub can get the token immediately without waiting for the completion of communication sessions of all other cell nodes. For example, without using the priority scheduling mechanism, the astrocyte hub can only get the token after the current astrocyte cell node (the one that holds the token) and the rest of the astrocyte cell nodes (between current astrocyte cell node and astrocyte hub) complete their communication sessions. Therefore the location of the current astrocyte cell node determines the length of waiting time. Fig. 9 presents the results of IP_3 exchange rate between two astrocyte tile facilities with and without the use of the priority scheduling mechanism. In this figure, m is assumed to be a constant value (i.e. ten) and the ID of the astrocyte hub is eleven (i.e. $m + 1$). The x-axis denotes the ID (i.e. location) of the current astrocyte cell node which holds the token. Without using the proposed scheduling mechanism, when the ID of the current astrocyte cell node increases, i.e. the current astrocyte cell node is closer to the astrocyte hub, the waiting time of the astrocyte hub decreases. As the packet transmission time between the facilities are fixed, the IP_3 exchange rate between two neighbouring astrocyte tile facilities increases from 361 kHz to 1,342 kHz. However, if using the proposed priority scheduling mechanism, the waiting time is a constant value as the astrocyte hub gets the token when current communication session is completed and does not need to wait for other astrocyte cell nodes to complete. Thus the IP_3 exchange rate is unaffected by the location of the current astrocyte cell node, and can sustain a maximum constant rate of 1,342 kHz as illustrated in Fig. 9. The advantage of the proposed priority scheduling mechanism lies in that it can keep the IP_3 exchange rate at the max rate which is independent of the location of the astrocyte cell node that holds the token. Using this method, the IP_3 exchange rate between the astrocyte tile facilities can be maintained and the global data is allowed to be forwarded quickly, providing a high data throughput rate.

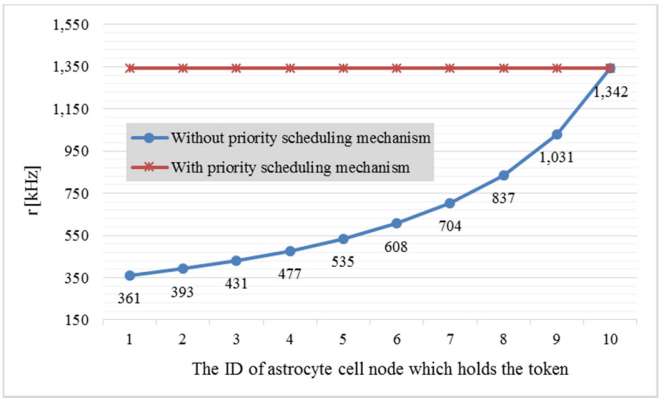

Fig. 9. IP_3 exchange rate (r) between two neighbouring astrocyte tile facilities under different locations of astrocyte cell node (holding the token) when the astrocyte hub receives the packet from astrocyte tile router.

C. *Analysis of packet delay*

The packet delay is defined as the time taken from the moment a data packet is generated until it reaches the target astrocyte cell node(s), i.e. communication pattern is One-to-All. It gives a measure of how fast astrocyte cells can exchange information with each other. In this subsection, the communication between three astrocyte tile facilities, i.e. [1, 2], [2, 2] and [3, 2] in Fig. 7(b), is used to analyse the packet delay. The aim of this experiment is to measure the time it takes for the packets to bypass an intermediate astrocyte tile facility (e.g. the astrocyte cell in [1, 2] and send a broadcast packet to the cells in [3, 2] via [2, 2]). The result of this experiment together with the results in section IV.B are used to derive an analytical equation to model the packet delay (σ) between the astrocyte tile facilities.

In order to measure packet delay σ, the time duration of one communication session (T_{CS}) is calculated first, which defines how long one astrocyte cell needs to finish the IP_3 exchange inside the astrocyte tile facility and is expressed by:

$$T_{CS} = \tau_t + (m + 1) \times \tau_n \quad (5)$$

where τ_t is a time constant that represents the time for the current cell node to receive the token packet from a previous cell node (5 clock cycles in this approach), m is the number of astrocyte cells within one astrocyte tile facility, τ_n is the packet forwarding time of the astrocyte node (4 clock cycles). T_{CS} is a fundamental time unit used in the calculation of packet delay σ.

The packet delay σ between the astrocyte tile facilities is composed of three parts, namely upstream delay (cell-to-tile), downstream delay (tile-to-cell), and delay between tile hops. The upstream delay (T_{up}) represents the packet delay from the time a packet is generated by the astrocyte cell node until the time it arrives at the astrocyte tile router, and can be expressed by:

$$T_{up} = T_{cb} + \tau_{up} \quad (6)$$

where T_{cb} is the time duration from the packet generation by the astrocyte cell to the time it arrives at the astrocyte hub, τ_{up} is a constant that represents the duration from the time when the packet leaves the astrocyte hub until the time it arrives at the astrocyte tile router (6 clock cycles). T_{cb} depends on the location of the source astrocyte cell node and is expressed by (7), where the maximum value of $m * \tau_n$ is used during the following performance analysis.

$$\tau_n \leq T_{cb} \leq m * \tau_n \quad (7)$$

The downstream delay (T_{down}) represents the packet delay from the time when packet leaves the astrocyte tile router of the target tile facility until the time it is absorbed by the destination astrocyte cell nodes, and is expressed by:

$$T_{down} = \tau_{down} + 2 * T_{CS} \quad (8)$$

where τ_{down} is a constant value that represents the duration from when the packet leaves the astrocyte tile router until the time it arrives at the astrocyte hub (6 clock cycles), $2 * T_{CS}$ is the total waiting time of the astrocyte hub in getting the token and the packet propagation time from the astrocyte hub to the target astrocyte cell nodes. In addition, as a comparison, the downstream delay without using the proposed priority scheduling mechanism (T'_{down}) is given by (9) where i is the location of the astrocyte cell node with the token, and $(m + 1 - i) \times T_{CS}$ is the waiting time for the astrocyte hub. It can be seen that T'_{down} is larger than T_{down} the majority of the time, in particular when the astrocyte cell node that holds the token is far away to the astrocyte hub (e.g. $i = 1$). T'_{down} has an accumulative decrease in unit delay as when i is low, the delay is high and as i increases the unit delay minimise, reaching the max rate of exchange.

$$T'_{down} = \tau_{down} + (m + 1 - i) \times T_{CS} + T_{CS} \quad (9)$$

Based on (6) and (8), the total packet delay σ can be expressed by (10) where N_h represents the total number of astrocyte tile facilities that a packet needs to go through to arrive at the target astrocyte tile facility, and varies depending on the communication path. τ_h is the time taken for a packet to bypass an astrocyte facility (3 clock cycles in this approach). All time constants for HANA can be found in Table III.

$$\sigma = T_{up} + N_h \times \tau_h + T_{down} \quad (10)$$

In addition, the astrocyte tile router uses an adaptive routing algorithm to choose the traffic free path to forward packets [24]. This communication path is also the minimum path for the source-destination nodes. The 2x3 array of astrocyte tile facilities in Fig. 7(b) illustrates the adaptive routing policy. The astrocyte tile facility [1,2] sends data packets to [3, 2] (as shown by path #1), and [2,2] sends packets to [3,1]. If congestion occurs in the east direction of [2,2], the astrocyte tile router [2,2] will choose alternative traffic free paths to forward the packets (e.g. the south direction of [2,2] in the path #2) which is also the minimum path. Therefore, the total packet delay σ between the astrocyte tile facilities are mainly determined by N_h (i.e. the total number of astrocyte tile facilities that a packet needs to go through). Further details on this adaptive routing algorithm are provided in [24].

TABLE III. TIME CONSTANTS OF HANA.

Constant	Constant Description	#CLK cycles
τ_t	The time for the current cell node to receive the token packet from a previous cell node	5
τ_n	The packet forwarding time of the astrocyte node	4
τ_{up}	The duration from the packet leaves the astrocyte hub until it arrives at the astrocyte tile router	6
τ_{down}	The duration from the packet leaves the astrocyte tile router until it arrives at the astrocyte hub	6
τ_h	The time for the packet bypasses the astrocyte facility	3

D. Scalability analysis

The packet delay analysis in the previous subsection is used as a starting point to evaluate the scalability of the proposed HANA architecture in this section. Five different large scale systems based on the astrocyte tile facilities are used. They are in sizes of 10x10, 20x20, 30x30, 40x40, and 50x50. In this approach, assume m is ten, and therefore can accommodate 1K, 4K, 9K, 16K, 25K astrocyte cells, and communicate with 10K, 40K, 9K, 160K, and 250K neurons, respectively. These large scale systems are designed at RTL level using VHDL and simulated using Modelsim. The system performances with and without the priority scheduling mechanism are also provided in terms of the maximum packet delay, σ, and minimum IP_3 exchange rate, r. They are mainly determined by the critical path (number of packet hops) of the network, $\emptyset$, which, based on HANA's top-level 2D mesh topology, can be calculated by:

$$\emptyset = DIM_x + DIM_y - 2 \quad (11)$$

DIM_x and DIM_y are the number of astrocyte tile facilities located in a row and column respectively. The critical path can be seen as the largest path, with the minimal number of intermediate hops between the source and destination astrocyte tile facilities, i.e. the path a packet needs to travel upon in order to reach its destination. For a 10x10 system, the critical path $\emptyset$ is 18 hops, however $\emptyset$ increases to 98 for a 50x50 system. The critical path $\emptyset$ determines the N_h while calculating the packet delay using (10), and the scalability of the proposed HANA is affected by the size of the astrocyte tile facility array. Fig. 10 shows the maximum packet delay σ and minimum IP_3 exchange rate r under different array sizes for the HANA systems; with and without priority scheduling mechanism. When the array size increases, the critical path $\emptyset$ is larger and the maximum packet delays increase for both the approaches with/without the priority scheduling mechanism. But much lower packet delays are achieved by using the priority scheduling mechanism, e.g. ~55% (for the 10x10 array) and ~40% (for the 50x50 array) lower than the approach without the priority scheduling mechanism. For the IP_3 exchange rates, they decrease when the array size increases, e.g. from 469 kHz to 300 kHz, and from 995 kHz to 453 kHz for the systems without and with priority scheduling, respectively. Using the priority scheduling mechanism achieves a higher IP_3 exchange rate than the system without the priority scheduling mechanism. The high IP_3 exchange rate is very beneficial for

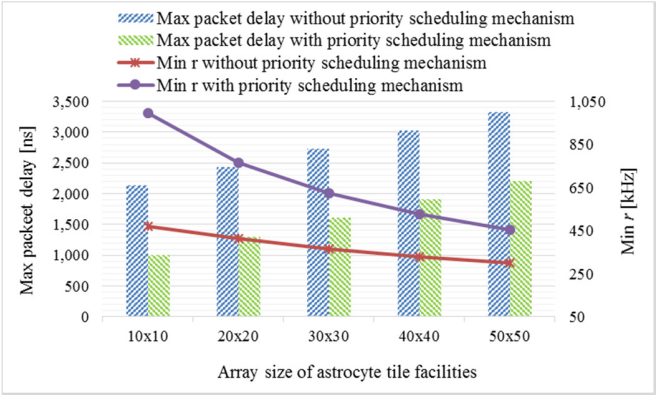

Fig. 10. Maximum packet delay σ (bars) and minimum IP_3 exchange rate r (lines) for different array sizes of astrocyte tile facilities.

information exchange between the astrocyte cells. For a large scale ANN system, e.g. a system with 50x50 array size accommodating 25K astrocyte cells and communicating with 250K neurons, HANA has the capability to provide a high IP_3 exchange rate of 453 kHz for the astrocyte cells, which is much higher than the biological exchange rate ~10 Hz (over 4 orders of magnitude faster). This enables system scalability to be maintained and opens the possibility of hardware acceleration of the astrocyte-neural networks.

E. Area utilization and power consumption

The area utilization and power consumption for the proposed HANA is performed based on a SAED 90nm CMOS technology [43]. HANA includes several components at different levels, e.g. the astrocyte cell network controller with/without the token generation function inside the astrocyte cell nodes, the astrocyte hub and astrocyte tile router. The packet payload in HANA is mainly used to exchange the IP_3 information between the astrocyte cells. The data width of the IP_3 value (i.e. payload) determines the communicated packet size which affects the hardware area and power consumption. HANA was implemented under various data widths of IP_3 values, e.g. 16, 32, 64-bit, in order to analyse system performance. The area utilization and power consumption for the different components under various data widths are shown in Table IV.

TABLE IV. HARDWARE AREA AND POWER CONSUMPTION OF DIFFERENT COMPONENTS IN THE HANA.

Component	Area [mm^2]			Power [mW]		
Data width of payload IP_3 (bits)	16	32	64	16	32	64
Astrocyte cell network controller (with token generation)	0.023	0.032	0.05	2.4	3.12	5.22
Astrocyte cell network controller (without token generation)	0.023	0.031	0.05	2.31	3.13	5.11
Astrocyte hub	0.023	0.032	0.05	2.34	3.05	5.19
Astrocyte tile router	0.156	0.23	0.367	11.68	17.28	28.12

When the data width increases, all area utilizations and power consumptions of the different components increase as expected. The astrocyte tile router occupies more area and consumes more power than the other components, i.e. 0.15 mm^2 and 11.68 mW using a 16-bit data width. This is due to its large interconnection degree and path diversity capability. Note that the results shown in Table IV only take into account the components regarding the astrocyte cell node, astrocyte hub and astrocyte tile router micro-architecture, and do not include the components due to the astrocyte cells itself. Therefore, results only show the contribution of the interconnection fabric which is the focus and contribution of the paper. More details on the astrocyte cell hardware implementation can be found in our previous work [7]. The advantage of using HANA lies in that the proposed hierarchical architecture interconnects the astrocyte cells via various topologies (i.e. ring and mesh) which enables the trade-off between performance and resource utilization. For example, for one astrocyte tile facility with ten astrocyte cells, using HANA requires 0.409 mm^2 (i.e. the areas of ten astrocyte cell network controllers, one astrocyte hub and one astrocyte tile router); however if using a mesh router to connect the ten astrocyte cells together, this requires 1.56 mm^2 (0.156 mm^2 x 10). Therefore the HANA achieves three times lower hardware resource utilization than the traditional straight connection using the mesh topology.

Table IV illustrates that a large data width of IP_3 introduces additional hardware area and power dissipation. The data width determines the IP_3 value precision. The IP_3 can be represented by different data types, e.g. floating point (32-bit), double point (64-bit). According to the astrocyte cell model and parameters, the value of IP_3 ranges from 0 to ~2 microMolar [5], [6]. Therefore an unsigned fixed-point data type with a total 16-bit word length (2-bit integer length and 14-bit fraction length) can be used to represent the IP_3 value, which gives a data range between 0 and 4 and has sufficient precision to model the astrocyte-neuron interactions. By using the payload data width of 16-bit, a fair comparison regarding the hardware resource of the proposed HANA with other existing approaches is shown in Table V. Most of these approaches have the same or smaller payload data width. Table V shows that most of the approaches are based on 90nm CMOS technology and the FTDR routers of [44] are based on TSMC 65nm technology. To make a comparison between the 90nm and 65nm node technology implementations, a first order scaling factor can be used for evaluation. For example, the area of a NAND 2x1 gate is 1.44 µm^2 for TSMC 65nm and 5.5296 µm^2 for SAED 90nm [43], therefore a first order scaling parameter, δ=5.5296/1.44, can be used to convert the area occupied using 65nm node to 90nm. The first order 90nm area calculations for the FTDRs with different error control coding modules are 0.63 mm^2 and 0.611 mm^2, respectively. Therefore, for the 2D mesh topology the HANA astrocyte tile router achieves a relatively low area utilization (only 0.156 mm^2) compared to other approaches (from 0.182 mm^2 [45] to 0.63 mm^2 [44]). Note the approach of [24] has a lower area overhead but the packet size is much smaller. The astrocyte network controller and astrocyte hub have very low area utilizations, which are only a combined ~50% of the total router area of [24], ~6% of the FTDR router in [44]. Table V highlights that compared to other approaches the interconnection fabric of HANA has a relatively low area utilization, which enables scalability to be maintained for larger astrocyte-neuron network hardware implementations.

TABLE V. HARDWARE AREA UTILIZATIONS OF DIFFERENT APPROACHES.

	The approach	Topology	Area overhead (mm^2)	
			Router	Device technology
	[24]	2D Mesh	0.056	90nm CMOS
	[45]	2D Mesh	0.182	SAED 90nm
	FTDR	2D Mesh	0.102	TSMC 65nm
	FTDR ECC(1) [44]	2D Mesh	0.164	TSMC 65nm
	FTDR ECC(2) [44]	2D Mesh	0.159	TSMC 65nm
	EDAR router [46]	2D Mesh	0.241	SAED 90nm
	CG router [47]	2D Mesh	0.237	SAED 90nm
	FG router [47]	2D Mesh	0.267	SAED 90nm
This work	**Astrocyte network controller**	**Ring**	**0.023**	**SAED 90nm**
	Astrocyte hub	**Ring**	**0.023**	**SAED 90nm**
	Astrocyte tile router	**2D Mesh**	**0.156**	**SAED 90nm**

V. CONCLUSION

The interconnected hardware architecture for the spiking astrocyte-neuron network (HANA) was proposed in this paper.

It builds on the H-NoC interconnect strategy from our previous work [16] for spiking neuron connections, with the proposed HANA providing a novel interconnection strategy for the additional accommodation of large scale astrocyte network hardware requirements. HANA is a hierarchical interconnected architecture that enables the efficient connections and communications between multiple tiles of hardware astrocyte cells using a two-level networks-on-chip. HANA achieves high system scalability, offers modular on-chip communication infrastructure for inter-astrocyte connectivity, and also provides a priority scheduling mechanism to increase the global IP_3 data rate of exchange to meet real-time processing requirements.

HANA is designed and implemented using VHDL and evaluated and verified in FPGA hardware. The experimental results demonstrate that the proposed HANA architecture offers an IP_3 exchange rate of ~1 MHz for the local astrocyte cells and ~453 kHz for the global cells (e.g. 50x50 array size), which is much higher than the biological exchange rate and therefore, provides over 4 orders of magnitude in hardware acceleration. HANA was also synthesized using 90nm CMOS technology and the results demonstrate that it has a relatively low area utilization (0.023 and 0.156 mm^2) and power consumption (5.11 and 28.12 mW) for the network controllers and router in a single astrocyte tile facility, respectively.

The work in this paper is a part of the long term EMBRACE project which aims to emulate the brain-inspired self-detection and self-repairing mechanisms in hardware systems [48]. The proposed HANA is an efficient solution in overcoming the interconnection problems of large scale spiking astrocyte-neuron hardware implementations. Future work will explore adaptive mapping algorithms for the large volume of neurons and astrocytes, i.e. how to map the spiking neurons and astrocytes to the current HANA architecture in an efficient manner (e.g. power-efficient). Work will also explore refined computational models for area-optimised hardware glial cells.

REFERENCES

[1] S. Ghosh-dastidar and H. Adeli, "Spiking Neural Networks," International Journal of Neural Systems, vol. 19, no. 4, pp. 295–308, 2009.

[2] M. De Pittà, N. Brunel, and A. Volterra, "Astrocytes: Orchestrating Synaptic Plasticity?," Neuroscience, pp. 1–19, 2015.

[3] M. De Pittà and N. Brunel, "Modulation of Synaptic Plasticity by Glutamatergic Gliotransmission: A Modeling Study," Neural Plasticity, vol. 7607924, pp. 1–30, 2016.

[4] L. E. Clarke and B. A. Barres, "Emerging Roles of Astrocytes in Neural Circuit Development.," Nature Reviews Neuroscience, vol. 14, no. 5, pp. 311–321, 2013.

[5] J. Wade, L. McDaid, J. Harkin, V. Crunelli, and S. Kelso, "Self-Repair in a Bidirectionally Coupled Astrocyte-Neuron (AN) System based on Retrograde Signaling," Frontiers in Computational Neuroscience, vol. 6, no. 76, pp. 1–12, Jan. 2012.

[6] M. Naeem, L. J. McDaid, J. Harkin, J. J. Wade, and J. Marsland, "On the Role of Astroglial Syncytia in Self-Repairing Spiking Neural Networks," IEEE Transactions on Neural Networks and Learning Systems, vol. 26, no. 10, pp. 2370–2380, 2015.

[7] J. Liu, J. Harkin, L. Maguire, L. McDaid, J. Wade, and M. McElholm, "Self-Repairing Hardware with Astrocyte-Neuron Networks," in IEEE International Symposium on Circuits and Systems (ISCAS), 2016, pp. 1350–1353.

[8] E. L. Bienenstock, L. N. Cooper, and P. W. Munro, "Theory for the Development of Neuron Selectivity: Orientation Specificity and Binocular Interaction in Visual Cortex," The Journal of Neuroscience, vol. 2, no. 1, pp. 32–48, 1982.

[9] V. Volman, E. Ben-Jacob, and H. Levine, "The Astrocyte as a Gatekeeper of Synaptic Information Transfer," Neural Computation, vol. 19, no. 2, pp. 303–326, 2007.

[10] S. Nadkarni and P. Jung, "Modeling Synaptic Transmission of the Tripartite Synapse," Physical Biology, vol. 4, no. 1, pp. 1–9, 2007.

[11] P. G. Haydon and G. Carmignoto, "Astrocyte Control of Synaptic Transmission and Neurovascular Coupling," Physiological Reviews, vol. 86, no. 3, pp. 1009–1031, 2006.

[12] M. Navarrete and A. Araque, "Endocannabinoids Mediate Neuron-Astrocyte Communication," Neuron, vol. 57, no. 6, pp. 883–893, 2008.

[13] S. B. Furber, D. R. Lester, L. A. Plana, J. D. Garside, E. Painkras, S. Temple, and A. D. Brown, "Overview of the SpiNNaker System Architecture," IEEE Transactions on Computers, vol. 62, no. 12, pp. 2454–2467, 2013.

[14] X. Lagorce, E. Stromatias, F. Galluppi, L. a Plana, S. Liu, S. B. Furber, X. Lagorce, E. Stromatias, F. Galluppi, and L. a Plana, "Breaking The Millisecond Barrier On SpiNNaker: Implementing Asynchronous Event-Based Plastic Models With Microsecond Resolution," Frontiers in Neuromorphic Engineering, vol. 9, no. 206, pp. 1–14, 2015.

[15] R. F. Service, "The Brain Chip," Science, vol. 345, no. 6197, pp. 614–616, Aug. 2014.

[16] S. Carrillo, J. Harkin, L. J. McDaid, F. Morgan, S. Pande, S. Cawley, and B. McGinley, "Scalable Hierarchical Network-on-Chip Architecture for Spiking Neural Network Hardware Implementations," IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 12, pp. 2451–2461, 2013.

[17] J. Misra and I. Saha, "Artificial Neural Networks in Hardware: A Survey of Two Decades of Progress," Neurocomputing, vol. 74, no. 1–3, pp. 239–255, 2010.

[18] L. P. Maguire, T. M. McGinnity, B. Glackin, A. Ghani, A. Belatreche, and J. Harkin, "Challenges for Large-Scale Implementations of Spiking Neural Networks on FPGAs," Neurocomputing, vol. 71, no. 1–3, pp. 13–29, 2007.

[19] P. Merolla, J. Arthur, F. Akopyan, N. Imam, R. Manohar, and D. S. Modha, "A Digital Neurosynaptic Core using Embedded Crossbar Memory with 45pJ per Spike in 45nm," in IEEE Custom Integrated Circuits Conference (CICC), 2011, pp. 1–4.

[20] J. Seo, B. Brezzo, Y. Liu, B. D. Parker, S. K. Esser, R. K. Montoye, B. Rajendran, J. A. Tierno, L. Chang, D. S. Modha, and D. J. Friedman, "A 45nm CMOS Neuromorphic Chip with a Scalable Architecture for Learning in Networks of Spiking Neurons," in IEEE Custom Integrated Circuits Conference (CICC), 2011, pp. 1–4.

[21] A. S. Cassidy, R. Alvarez-Icaza, F. Akopyan, J. Sawada, J. V. Arthur, P. A. Merolla, P. Datta, M. G. Tallada, B. Taba, A. Andreopoulos, A. Amir, S. K. Esser, J. Kusnitz, R. Appuswamy, C. Haymes, B. Brezzo, R. Moussalli, R. Bellofatto, C. Baks, M. Mastro, K. Schleupen, C. E. Cox, K. Inoue, S. Millman, N. Imam, E. McQuinn, Y. Y. Nakamura, I. Vo, C. Guo, D. Nguyen, S. Lekuch, S. Asaad, D. Friedman, B. L. Jackson, M. D. Flickner, W. P. Risk, R. Manohar, and D. S. Modha, "Real-Time Scalable Cortical Computing at 46 Giga-Synaptic OPS/Watt with ~100X Speedup in Time-to-Solution and ~100,000X Reduction in Energy-to-Solution," in International Conference for High Performance Computing, Networking, Storage and Analysis, 2014, pp. 27–38.

[22] B. V. Benjamin, P. Gao, E. McQuinn, S. Choudhary, A. R. Chandrasekaran, J. M. Bussat, R. Alvarez-Icaza, J. V. Arthur, P. a. Merolla, and K. Boahen, "Neurogrid: A Mixed-Analog-Digital Multichip System for Large-Scale Neural Simulations," Proceedings of the IEEE, vol. 102, no. 5, pp. 699–716, 2014.

[23] L. Benini and D. M. Giovanni, "Networks on Chips: A New SoC Paradigm," Computer, vol. 35, no. 1, pp. 70–78, 2002.

[24] S. Carrillo, J. Harkin, L. McDaid, S. Pande, S. Cawley, B. McGinley, and F. Morgan, "Advancing Interconnect Density for Spiking Neural Network Hardware Implementations using Traffic-aware Adaptive Network-on-Chip Routers," Neural Networks, vol. 33, no. 9, pp. 42–57, 2012.

[25] S. Pande, F. Morgan, G. Smit, T. Bruintjes, J. Rutgers, B. McGinley, S. Cawley, J. Harkin, and L. McDaid, "Fixed latency on-chip interconnect for hardware spiking neural network architectures," Parallel Computing, vol. 39, no. 9, pp. 357–371, Apr. 2013.

[26] S. Pande, F. Morgan, S. Cawley, T. Bruintjes, G. Smit, B. McGinley, S. Carrillo, J. Harkin, and L. McDaid, "Modular Neural Tile Architecture for Compact Embedded Hardware Spiking Neural Network," Neural Processing Letters, Jan. 2013.
[27] S. Carrillo, J. Harkin, and L. McDaid, "Hierarchical Networks-on-Chip Architecture for Neuromorphic Hardware," in Evolvable Hardware, 2015, pp. 297–330.
[28] A. Araque, G. Carmignoto, P. G. Haydon, S. H. R. Oliet, R. Robitaille, and A. Volterra, "Gliotransmitters Travel in Time and Space," Neuron, vol. 81, no. 4, pp. 728–739, 2014.
[29] E. M. Izhikevich, "Simple Model of Spiking Neurons," IEEE Transactions on Neural Networks, vol. 14, no. 6, pp. 1569–1572, 2003.
[30] R. FitzHugh, "Impulses and Physiological States in Theoretical Models of Nerve Membrane," Biophysical Journal, vol. 1, no. 6, pp. 445–466, 1961.
[31] S. Nazari, M. Amiri, K. Faez, and M. Amiri, "Multiplier-less Digital Implementation of Neuron–astrocyte Signalling on FPGA," Neurocomputing, vol. 164, pp. 281–292, 2015.
[32] M. Hayati, M. Nouri, S. Haghiri, and D. Abbott, "A Digital Realization of Astrocyte and Neural Glial Interactions," IEEE Transactions on Biomedical Circuits and Systems, vol. 10, no. 2, pp. 518–529, 2016.
[33] D. E. Postnov, R. N. Koreshkov, N. A. Brazhe, A. R. Brazhe, and O. V. Sosnovtseva, "Dynamical Patterns of Calcium Signaling in a Functional Model of Neuron-astrocyte Networks," Journal of Biological Physics, vol. 35, no. 4, pp. 425–445, 2009.
[34] H. Soleimani, M. Bavandpour, A. Ahmadi, and D. Abbott, "Digital Implementation of a Biological Astrocyte Model and Its Application," IEEE Transactions on Neural Networks and Learning Systems, vol. 26, no. 1, pp. 127–139, 2015.
[35] D. E. Postnov, L. S. Ryazanova, and O. V. Sosnovtseva, "Functional Modeling of Neural-Glial Interaction," BioSystems, vol. 89, no. 1, pp. 84–91, 2007.
[36] S. Nazari, K. Faez, M. Amiri, and E. Karami, "A Digital Implementation of Neuron–astrocyte Interaction for Neuromorphic Applications," Neural Networks, vol. 66, pp. 79–90, 2015.
[37] M. De Pittà, V. Volman, H. Levine, and E. Ben-Jacob, "Multimodal Encoding in a Simplified Model of Intracellular Calcium Signaling," Cognitive Processing, vol. 10, no. Suppl 1, pp. 55–70, 2009.
[38] A. Araque, V. Parpura, R. P. Sanzgiri, and P. G. Haydon, "Tripartite Synapses: Glia, the Unacknowledged Partner," Trends in Neurosciences, vol. 22, no. 5, pp. 208–215, 1999.
[39] M. Goldberg, M. De Pittà, V. Volman, H. Berry, and E. Ben-Jacob, "Nonlinear Gap Junctions Enable Long-Distance Propagation of Pulsating Calcium Waves in Astrocyte Networks," PLoS Computational Biology, vol. 6, no. 8, pp. 1–14, 2010.
[40] J. J. Wade, L. J. McDaid, J. Harkin, V. Crunelli, J. A. S. Kelso, and V. Beiu, "Exploring Retrograde Signaling via Astrocytes as a Mechanism for Self Repair," in International Joint Conference on Neural Networks (IJCNN), 2011, pp. 3149–3155.
[41] M. M. Halassa, T. Fellin, H. Takano, J.-H. Dong, and P. G. Haydon, "Synaptic Islands Defined by the Territory of a Single Astrocyte," The Journal of Neuroscience, vol. 27, no. 24, pp. 6473–6477, 2007.
[42] J. Liu, J. Harkin, Y. Li, L. Maguire, and A. Linares-Barranco, "Low Overhead Monitor Mechanism for Fault-tolerant Analysis of NoC," in IEEE 8th International Symposium on Embedded Multicore/Many-core Systems-on-Chip, 2014, pp. 189–196.
[43] Sysopsys, "SAED EDK90 Core Digital Standard Cell Library." pp. 1–102, 2009.
[44] C. Feng, Z. Lu, A. Jantsch, M. Zhang, and Z. Xing, "Addressing Transient and Permanent Faults in NoC with Efficient Fault-Tolerant Deflection Router," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 21, no. 6, pp. 1053–1066, Jun. 2013.
[45] J. Liu, J. Harkin, Y. Li, and L. Maguire, "Online Traffic-Aware Fault Detection for Networks-on-Chip," Journal of Parallel and Distributed Computing, vol. 74, no. 1, pp. 1984–1993, 2014.
[46] J. Liu, J. Harkin, Y. Li, and L. Maguire, "Low Cost Fault-tolerant Routing Algorithm for Networks-on-Chip," Microprocessors and Microsystems, vol. 39, no. 6, pp. 358–372, 2015.
[47] J. Liu, J. Harkin, Y. Li, and L. P. Maguire, "Fault Tolerant Networks-on-Chip Routing with Coarse and Fine-Grained Look-ahead," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 35, no. 2, pp. 260–273, 2016.
[48] J. Harkin, F. Morgan, L. McDaid, S. Hall, B. McGinley, and S. Cawley, "A Reconfigurable and Biologically Inspired Paradigm for Computation Using Network-On-Chip and Spiking Neural Networks," International Journal of Reconfigurable Computing, vol. 2009, pp. 1–13, 2009.

Junxiu Liu received the PhD degree from University of Ulster, UK, in 2015. His research interests relate to: neural glial system modelling, and its hardware implementations.

Jim Harkin received the BTech degree in electronic engineering, the M.Sc. degree in electronics and signal processing, and the Ph.D. degree from the University of Ulster, Derry, Northern Ireland, U.K., in 1996, 1997, and 2001, respectively. He is a Senior Lecturer with the School of Computing and Intelligent Systems at Ulster. His research focuses on the design of intelligent embedded systems to support self-repairing capabilities and the hardware/software implementation of spiking neural networks.

Liam P. Maguire received MEng and PhD degrees in Electrical and Electronic Engineering from the Queen's University of Belfast, UK, in 1988 and 1991, respectively. He is currently Dean of the Faculty of Computing and Engineering and also Director of the Intelligent Systems Research Centre at the University of Ulster. His research interests are in two primary areas: fundamental research in bio-inspired intelligent systems and the application of existing intelligent techniques in different domains. He is the author of over 200 research papers including 70 journal papers. He has an established track record of securing research funding and has also supervised 15 PhD and 3 MPhil students to completion.

Liam J. McDaid received his B.Eng. (Hons) degree in electrical and electronics engineering from the University of Liverpool in 1985 and in 1989; he received a PhD in solid-state devices from the same institution. He is currently Professor of Computational Neuroscience at the Ulster University and leads the Computational Neuroscience and Neural Engineering (CNET) Research Team. His current research interests include modelling the role of glial cells in the functional and dysfunctional brain and he is also involved in the development of software/hardware models of neural-based computational systems, with particular emphasis on the mechanisms that underpin self-repair in the human brain. He has received several research grants in this domain and is currently a collaborator on an HFSP and EPSRC funded project. He has co-authored over 120 publications in his career to date.

John J. Wade received his B.Eng. in Electronics and Computing and M.Sc. in Computing and Intelligent Systems degrees from the University of Ulster, Northern Ireland, in 2004 and 2005 respectively. He was awarded his Ph.D. degree in Computational Neural Systems at the University of Ulster in 2010. Dr. Wade is currently employed as a researcher at the Intelligent Systems Research Center (ISRC) at the Magee campus of the University of Ulster where he is part of the Computational Neuroscience and Neural Engineering (CNET) Research Team. His main research interest lies in developing computational models of neural and glial systems to aid understanding of how the brain functions and learns.

George Martin received the B.Eng. in Electronics and Computing from University of Ulster (2014). He is currently completing a PhD at Ulster. His PhD research focuses on hardware interconnect strategies for self-repairing neuro-glia networks.