SPANNER: A Self-Repairing Spiking Neural Network Hardware Architecture

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

***Abstract*— Recent research has shown that a glial cell of astrocyte underpins a self-repair mechanism in the human brain, where spiking neurons provide direct and indirect feedbacks to presynaptic terminals. These feedbacks modulate the synaptic transmission probability of release (PR). When synaptic faults occur, the neuron becomes silent or near silent due to the low PR of synapses; whereby the PRs of remaining healthy synapses are then increased by the indirect feedback from the astrocyte cell. In this paper, a novel hardware architecture of Self-rePAiring spiking Neural NEtwoRk (SPANNER) is proposed, which mimics this self-repairing capability in the human brain. This paper demonstrates that the hardware can self-detect and self-repair synaptic faults without the conventional components for the fault detection and fault repairing. Experimental results show that SPANNER can maintain the system performance with fault densities of up to 40%, and more importantly SPANNER has only a 20% performance degradation when the self-repairing architecture is significantly damaged at a fault density of 80%.**

***Index Terms*— Astrocytes, electronic systems, fault tolerance, field-programmable gate array (FPGA), hardware, self-repair, spiking neural network (SNN).**

I. Introduction

ELECTRONIC systems are ubiquitous and underpin nearly all aspects of industrial and social endeavor. Fault tolerance in electronic systems is a design challenge due to downscaling of semiconductor devices. A wide range of permanent and temporary failures [1] in electronic systems are a result of either manufacturing defects (e.g., stuck-at faults), environmental effects (e.g., power supply voltage fluctuation and temperature variation), or soft errors [2] (e.g., single event upset, single event transient errors due to cosmic rays, and so on). These lead to varied levels of performance degradation [3], [4] and ultimately unreliable systems. The ability to sense failure, classify it and implement corrective action in order to sustain functional operation of a system, is an ultimate design requirement for electronic engineers of safety critical systems and even devices of large scale. Current fault-tolerant computing methods incorporate redundancy or replication models [5]–[9], techniques for error correcting [10], and field-programmable gate array (FPGA)-based radiation hardening [7], however, they fail to provide the capability to detect faults and implement repair at fine levels [5]–[8], [10], [11]. As a result, alternative methods, which allow fault detection, diagnosis, and repair at finer levels of granularity [10], are required. "Evolutionary" approaches have taken inspiration from biology in providing system reliability via self-repair and self-organization properties [12], and their success underpins the belief that future systems will need to harness similar mechanisms found in nature. Existing bioinspired approaches [13] have exploited the reconfigurability of FPGAs to provide adaptive repair at finer levels of granularity although the FPGA building blocks are typically coarse, which means repair is done at a coarse level. More importantly, the repair decision process is not distributed and, therefore, can itself be easily compromised. We know that as devices scale in size due to large many-core systems, the random nature of faults means this is a significant challenge [9]. Therefore, we need to explore new approaches to assist in developing highly adaptive, distributed computing systems [10].

Significant advances in neuroscience have provided us with insights into how networks in the brain process and communicate information in a robust and power-efficient manner. Currently, spiking neural networks (SNNs) are the most promising model as they capture the key information processing and communication capabilities of the brain [14]. Compared with traditional neural networks, SNNs reflect the dynamic behaviors and information processing mechanisms of a biological neural system, and also exhibit temporal pattern processing [15] and fault-tolerant capabilities as seen in their biological counterparts [16]. For example, SNNs are able to enhance the fault-tolerant capability of neural network due to the inherent nature of distributed computing and therefore, unlike traditional computing devices, any degradation in the computational capability correlates with the density of faulty connections or neurons. However, until recently the key question of what coordinates the repair process in the brain has been unanswered. Similar to the structural plasticity observed in biological neural networks, a novel learning algorithm is proposed in the approach of [17] whereby it can rewire the readout networks of the liquid state machine. If synapses have a low probability of release that reduces the firing rate of the neuron, these synapses would exhibit reduction of

Manuscript received May 11, 2016; revised September 14, 2016, January 10, 2017, and February 17, 2017; accepted February 20, 2017. Date of publication March 6, 2017; date of current version March 15, 2018. The work of J. Liu was supported in part by the National Natural Science Foundation of China under Grant 61603104, in part by the Guangxi Natural Science Foundation under Grant 2015GXNSFBA139256, Grant 2016GXNSFCA 380017, in part by the funding of Overseas 100 Talents Program of Guangxi Higher Education, in part by the Research Project of Guangxi University of China under Grant KY2016YB059, in part by Guangxi Key Lab of Multi-source Information Mining & Security under Grant MIMS15-07, and in part by the Doctoral Research Foundation of Guangxi Normal University.

J. Liu is with the Guangxi Key Lab of Multi-Source Information Mining and Security, Faculty of Electronic Engineering, Guangxi Normal University, Guilin 41004, China (e-mail: liujunxiu@mailbox.gxnu.edu.cn).

J. Harkin, L. P. Maguire, L. J. McDaid, and J. J. Wade are with the School of Computing and Intelligent Systems, Ulster University, Magee campus, Derry BT48 7JL, U.K. (e-mail: jg.harkin@ulster.ac.uk; lp.maguire@ulster.ac.uk; lj.mcdaid@ulster.ac.uk; jj.wade@ulster.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2017.2673021

local synaptic variables and the inactive connections would get swapped by more active connections. However, in the proposed paper, we go further and provide a solution to the problem of how repair is done not only within a group of synapses but across a multilayer network. The proposed network model is much richer and captures the self-repairing capability of biological neural networks.

Progress has been made in investigating how the astrocytes, one subtype of glial cells, play a crucial role in the fine-grained self-repairing capability of neural networks [18], [19]. Astrocyte cell was only considered as abundant glial cell previously, which supports physical structuring of brain, however, recently researchers reconsidered its function and thought that it involves a number of activities in the brain, including the behavior regulation of ion pathways, modulation of synaptic formation, and especially the neural network repair [18], [20]. Recent computational models [21], [22] have successfully demonstrated that astrocytes regulate the synaptic transmissions and exhibit a fine-grained self-repairing capability in the spiking neural network, however, to date no research has explored this capability in hardware. Current approaches are mainly focused on the hardware implementation of astrocyte cell and its fundamental interactions with spiking neurons. The digital circuits for neuron-astrocyte interactions were proposed in the approach of [23] and [24], where an Izhikevich [25] model and a FitzHugh [26] Nagumo model (i.e., simplified Hodgkin Huxley model) were used as neuron models, and dynamic models in the research work of [27] and [28] and [27] and [29] were employed to describe the behaviors of astrocyte. Based on the same models in the approach of [23], the neuron-astrocyte network model was further optimized for the digital hardware implementations in the research work of [30], which achieves a relatively low hardware overhead and maintains the scalability of neural networks.

In the previous works, Naeem *et al.* [21] and Wade *et al.* [22] have proposed a computational astrocyte-neuron network model which captures the self-repairing behaviors of SNNs, and demonstrates that the network can still maintain the target functions even when the faults are present. In this paper, we propose, for the first time, a hardware architecture of astrocyte-neuron network, i.e., Self-rePAiring spiking Neural NEtwoRk (SPANNER), to emulate the fault-tolerant capability in the hardware devices. A preliminary version of this paper is published in [31]; however, this paper provides new sections to present more technical details on the SPANNER hardware architecture, and extensive new experimental results to demonstrate the self-repairing capability and system scalability. In particular, graceful degradation performance for both spatial (quantity) and temporal degrees of injected faults are assessed in this paper. These are all new results, not published elsewhere. The results demonstrate that the implementation of astrocyte-neuron network in FPGA can mimic the structure of biological neuron network via the inherent parallelism of hardware, where the fault-tolerant capability of hardware system is enhanced using the self-repairing model. This has crucial implications for future reliable computing where applications can be mapped to self-repairing SNN architectures and implemented in hardware to provide highly adaptive and robust computing systems. The remainder of this paper is organized as follows: Section II outlines the authors' astrocyte-neuron network computational model and presents the hardware architecture of SPANNER in detail. Section III presents experimental results which analyze the hardware system performance and also demonstrates the repair capability of SPANNER under different fault conditions. Section IV provides a detailed discussions regarding the performance of the SPANNER, and Section V concludes this paper.

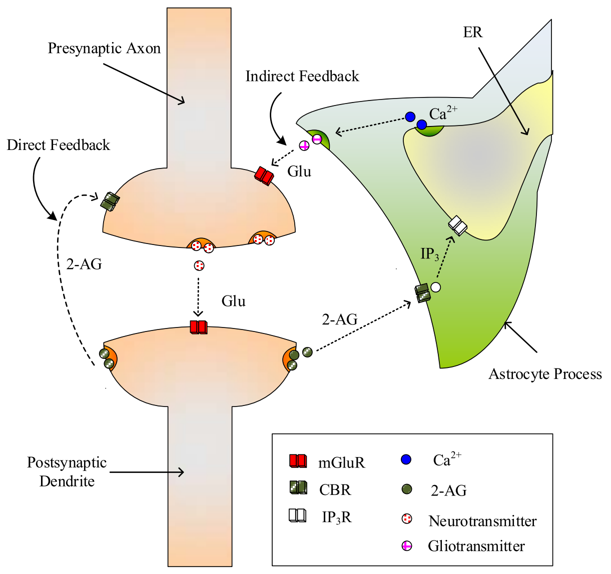


Fig. 1. Tripartite synapse.

II. Self-Repair Mechanism in SPANNER Hardware

A. Self-Repair Arising From a Coupled Astrocyte-Neuron System

This section provides the interaction mechanism between the astrocytes and neurons and how it gives rise to a self-repairing SNN. The research work of [32] showed that about half synapses have very close connections with neurons and also astrocytes, i.e., they actually communicate at three terminals, which are defined by the *tripartite synapse*. When various neurotransmitters bind to respective receptors at the astrocyte cell, the calcium level (Ca^{2+}) inside the astrocyte increases; and this transmit increase in Ca^{2+} propagates the calcium waves and releases the astrocytic gliotransmitters. Astrocytes are in communications with neurons via an indirect feedback mode, and modulate the transmission PRs of all synapses in the astrocytes domain of coverage. This synaptic modulation is the key process for the self-repairing mechanism when the synapses are broken or have low transmission probabilities.

Fig. 1 describes a tripartite synapse. When an action potential arrives at presynaptic terminal, a type of neurotransmitter, i.e., glutamate, is released which binds to the receptors at the postsynaptic dendrite. It depolarizes the postsynaptic neuron. After sufficient depolarization, Ca^{2+} flows into the dendrite via the voltage gated calcium channels. This process synthesises and releases endocannabinoids from the postsynaptic dendrite. The endocannabinoid, a type of retrograde messenger, is known to travel back to presynaptic terminals from the postsynaptic dendrite. In this approach, the endocannabinoid is 2-arachidonyl glycerol (2-AG) and it feeds back in two ways: first, *direct feedback*. The released 2-AG binds to type 1 Cannabinoid Receptors (CB1Rs) directly on the presynaptic terminal. It decreases the transmission PR of the synapse and is termed as depolarization-induced suppression of excitation (DSE) and second, *indirect feedback*. Another released amount of 2-AG binds to the CB1Rs of an astrocyte cell. This increases IP_3 levels inside the astrocyte, and then triggers a transient release of calcium. Consequently the glutamate is released from astrocyte, which feeds back to presynaptic group I metabotropic Glutamate Receptors at the presynaptic terminal. It increases the transmission PR of the synapse and is termed as e-SP.

B. Self-Repairing Model

When a postsynaptic neuron fires, 2-AG is released, which is modeled as

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

where AG is the quantity of the released 2-AG, τ_{AG} is decay rate of 2-AG, r_{AG} is production rate of 2-AG, and t_{sp} is the time of the postsynaptic spike. When the 2-AG binds to CB1Rs on the astrocyte, IP_3 is generated depending on the amount of released 2-AG and can be given by

$$\frac{d(\mathrm{IP}_3)}{dt} = \frac{\mathrm{IP}_3^* - \mathrm{IP}_3}{\tau_{\mathrm{ip}_3}} + r_{\mathrm{ip}_3}\mathrm{AG} \tag{2}$$

where IP_3 is the quantity within the cytoplasm, IP_3^* is the baseline of IP_3 when the astrocyte cell is in a steady state with no input received, τ_{ip3} is decay rate of IP_3, and r_{ip3} is production rate of IP_3.

The Li-Rinzel model [33] is employed to model the Ca^{2+} dynamics within the astrocyte cell. It uses three channels, i.e., J_{pump}, J_{leak}, and J_{chan}, for modeling. J_{pump} models how Ca^{2+} is stored within the Endoplasmic Reticulum (ER) by pumping Ca^{2+} out of the cytoplasm into the ER via Sarco-Endoplasmic-Reticulum Ca^{2+}-ATPase (SERCA) pumps, J_{leak} is Ca^{2+} which leaks out from the ER and goes into the cytoplasm, and J_{chan} models the opening of Ca^{2+} channels by the mutual gating of Ca^{2+} and IP_3 concentrations. The following equations are used in the model [34]:

$$\frac{d(Ca^{2+})}{dt} = J_{\mathrm{chan}}(Ca^{2+}, h, \mathrm{IP}_3) + J_{\mathrm{leak}}(Ca^{2+}) - J_{\mathrm{pump}}(Ca^{2+}) \tag{3}$$

$$\frac{dh}{dt} = \frac{h_\infty - h}{\tau_h} \tag{4}$$

where J_{chan} is Ca^{2+} release depending on the Ca^{2+} and IP_3 concentrations, J_{pump} is the amount of stored Ca^{2+} within the ER via the SERCA pumps, J_{leak} is the Ca^{2+} leaking out of the ER, and h is the fraction of activated IP_3R_s. The parameters h_∞ and τ_h are given by

$$h_\infty = \frac{Q_2}{Q_2 + Ca^{2+}} \tag{5}$$

and

$$\tau_h = \frac{1}{a_2(Q_2 + Ca^{2+})} \tag{6}$$

where

$$Q_2 = d_2\frac{\mathrm{IP}_3 + d_1}{\mathrm{IP}_3 + d_3}. \tag{7}$$

J_{chan} is given by

$$J_{\mathrm{chan}} = r_C m_\infty^3 n_\infty^3 h^3 (C_0 - (1 + C_1)Ca^{2+}) \tag{8}$$

where r_C is the maximal calcium induced calcium release (CICR) rate, C_0 is the total free Ca^{2+} cytosolic concentration, C_1 is the ER/cytoplasm volume ratio, and m_∞ and n_∞ are the IP_3 induced calcium release (IICR) and CICR channels, respectively, which are given by

$$m_\infty = \frac{\mathrm{IP}_3}{\mathrm{IP}_3 + d_1} \tag{9}$$

and

$$n_\infty = \frac{Ca^{2+}}{Ca^{2+} + d_5}. \tag{10}$$

J_{leak} and J_{pump} are described by

$$J_{\mathrm{leak}} = r_L(C_0 - (1 + C_1)Ca^{2+}) \tag{11}$$

and

$$J_{\mathrm{pump}} = v_{\mathrm{ER}}\frac{(Ca^{2+})^2}{k_{\mathrm{ER}}^2 + (Ca^{2+})^2} \tag{12}$$

where r_L is the Ca^{2+} leakage rate, v_{ER} is the maximum SERCA pump uptake rate, and k_{ER} is the SERCA pump activation constant.

Having modeled the 2-AG release and the Ca^{2+} dynamics within the cell, the direct and indirect effects from DSE and e-SP can be given as follows. The DSE is assumed to change linearly with the released 2-AG, which is described by

$$\mathrm{DSE} = \mathrm{AG} \times K_{\mathrm{AG}} \tag{13}$$

where AG is the amount of released 2-AG and K_{AG} is the scaling factor for the DSE and released 2-AG. Inside the astrocyte cell, calcium dynamic modeled by (3) regulates the release of glutamate, which is described by

$$\frac{d(\mathrm{Glu})}{dt} = \frac{-\mathrm{Glu}}{\tau_{\mathrm{Glu}}} + r_{\mathrm{Glu}}\delta(t - t_{Ca}) \tag{14}$$

where Glu is the quantity of released glutamate, τ_{Glu} is decay rate of glutamate, r_{Glu} is production rate of glutamate, and t_{Ca} is the time of the Ca^{2+} crosses the threshold. The released glutamate drives the generation of e-SP. The level of e-SP is modeled by

$$\tau_{\mathrm{eSP}}\frac{d(\mathrm{eSP})}{dt} = -\mathrm{eSP} + m_{\mathrm{eSP}}\mathrm{Glu}(t) \tag{15}$$

where τ_{eSP} is the decay rate of e-SP generation and m_{eSP} is a weighting constant used to control the weight of e-SP.

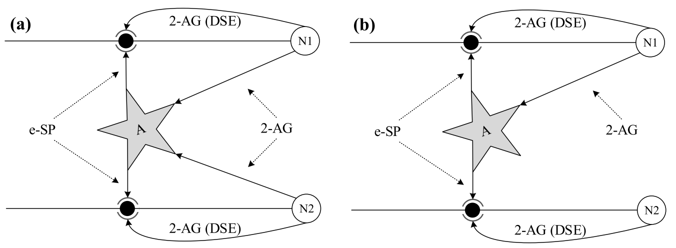


Fig. 2. Network fragments illustrating endocannabinoid medicated self-repair [22].

In this approach, the leaky integrate and fire (LIF) [35] is used as the neuron model, which is given by

$$\tau_m \frac{dv}{dt} = -v(t) + R_m \sum_{i=1}^{m} I_{\text{syn}}^{i}(t) \tag{16}$$

where τ_m is the time constant, v is the membrane potential of the neuron, R_m is the membrane resistance, and $I_{\text{syn}}^{i}(t)$ is the current injected to the membrane from synapse i. The refractory period for the neuron model is a period of 2 ms.

For the synapse model, a probabilistic-based model employed which is based on the failure and success mechanisms of synaptic neurotransmitter release observed in the approaches of [22] and [36]. A uniformly distributed pseudo-random number generator generates a random number, i.e., $rand$. If this random number is less than or equal to the PR, a fixed current I_{inj} is injected into the neuron, which is shown by

$$I_{\text{syn}}^{i}(t) = \begin{cases} I_{\text{inj}}, & \text{rand} \le \text{PR} \\ 0, & \text{rand} > \text{PR}. \end{cases} \tag{17}$$

The associated PR of each synapse is determined by the DSE and e-SP together, which is given by

$$\text{PR}(t) = \left(\frac{\text{PR}(t_0)}{100} \times \text{DSE}(t)\right) + \left(\frac{\text{PR}(t_0)}{100} \times \text{eSP}(t)\right) \tag{18}$$

where $\text{PR}(t_0)$ is the initial PR for each synapse. When a fault is simulated, the synapse is damaged and for this condition, $\text{PR}(t_0)$ is set to the fault PR value.

In summary, an SNN network fragment, shown in Fig. 2, is used to illustrate the self-repairing principle of the astrocyte-neuron networks. This network includes one astrocyte (A), two neurons (N1 and N2), and each neuron has several synapse inputs. From Fig. 2, it can be seen that the 2-AG (DSE) is a local signal for the synapses associated with each neuron. It tunes the PRs of all synapses associated with one neuron. However, as the astrocyte cell *A* connects to all the synapses in the network, the e-SP is a global signal for all the synaptic terminals. Two different statuses of the network are used to demonstrate the self-repairing principle. First case is a healthy network, which is shown by Fig. 2(a). For this network, after the input spike trains are presented, the 2-AG is released from postsynaptic neuron which lead to the generation of DSE and e-SP. They compete at the presynaptic terminals and archive a stable state for the transmission PR of each synapse. Second case is the network with partial faults, which is shown by Fig. 2(b). In this network, several synapses connected to the neuron *N2* are damaged, which cause the stop of the direct/indirect feedbacks from N2. Consequently, the balance between DSE and e-SP is broken for the synapses of N2. However, due to the indirect feedback of e-SP from astrocyte cell, the PRs of healthy synapses of N2 are enhanced to help the neuron N2 maintain the target output. This is a brief introduction of the self-repairing principle of the astrocyte-neuron network, and more details of this principle can be found in our previous research work of [21] and [22]. Note that the astrocyte-neuron model is based on cellular level which can handle rate codes and the firing frequency is limited only by the refractory period of neurons in principle.

C. SPANNER Hardware Architecture

This section presents the concepts and design details of the proposed SPANNER hardware. The SPANNER hardware architecture is based on the astrocyte-neuron SNN of Sections II-A and II-B. It includes three facilities—probabilistic tripartite synapse, LIF model [35], and the Pitta *et al.* [34] astrocyte model. These three facilities are introduced in detail in the following text.

1) Neuron Facility: The role of the *neuron facility* is to receive the currents from the connected synapses, and output spikes if the membrane potential, i.e., $v(t)$ in (16), is greater than the neuron threshold. Furthermore, while the neuron is depolarizing, it also generates the postsynaptic DSE and 2-AG signals. The LIF model is used in the neuron facility as it is one of the most common neuron models requiring relatively low-computational resources [37] which is more suited to implement large-scale network hardware systems.

Fig. 3 illustrates the internal block architecture of the *neuron facility* where the components in gray implement the release process of 2-AG and DSE. The input signals of *neuron facility* are from the synapse components. The output signals are neuron spike, 2-AG and DSE. Total current injected to the neuron from all synapses is calculated by the *neuron current generator* block. According to (16) of the LIF model, the injected current introduces the membrane potential voltage change. The *voltage change rate generator* in the *neuron facility* is employed to calculate the quantity of change. As the LIF neuron is modeled by the differential equation of (16), a *neuron voltage register* block is used to store the membrane potential voltage which is updated by the voltage change rate generator at each time step. The *neuron voltage register* outputs the membrane potential to a *spike generator* which compares the membrane potential (v) with a firing threshold (v_{th}). If $v < v_{\text{th}}$, the *spike generator* does not output spikes; otherwise a spike is generated. After spiking, the neural membrane potential goes to reset voltage for a refractory period (e.g., 2 ms in this approach). This process is implemented by the *neuron refractory period judge* component inside the *neuron facility*.

After the neuron outputs a spike and sufficiently depolarized, the endocannabinoid of 2-AG is generated, which feeds back to the presynaptic terminal via DSE. This process is implemented by the four components of the last row in Fig. 3

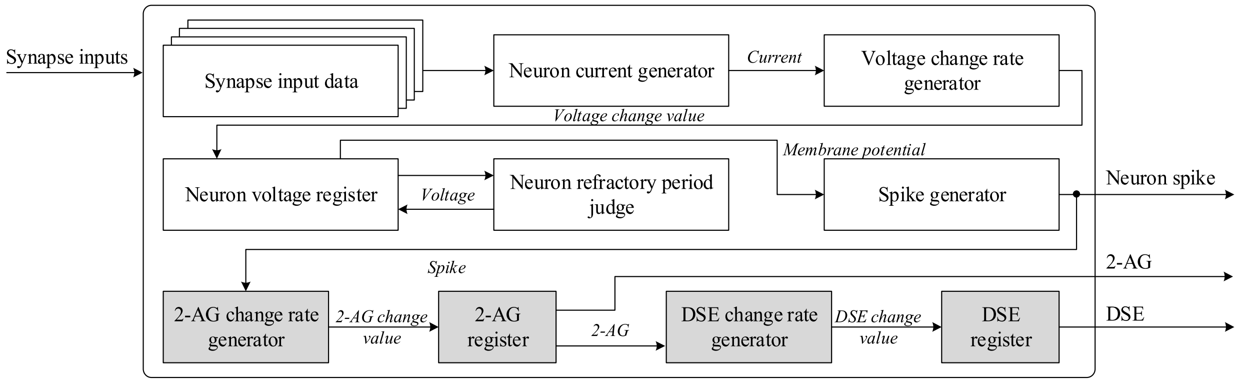


Fig. 3. Neuron facility.

(highlighted in gray). Similar to the change of membrane potential, a 2-AG *change rate generator* and 2-AG *register* blocks are employed to implement the 2-AG release behaviors. The 2-AG value is also used by the DSE *change rate generator* to calculate the change of DSE value according to (13). Similarly, a DSE register stores and updates the DSE value at each time step which is an output of the neuron facility.

The key functions of the *neuron facility* are summarized as follows: It receives input from synapses, calculates and updates the neural membrane potential, and outputs spikes, 2-AG and DSE signals [see (1) and (13)]. The 2-AG binds to the astrocyte and drives the e-SP which leads to the increase of synaptic PRs indirectly, however, the DSE decreases the synaptic PRs directly, see (18). The astrocyte and synapse facilities and their functions are described in the following text. Note that Fig. 3 shows the diagram of a neuron facility which contains only one neuron, however, more neurons can be included in a single neuron facility using the time-multiplexing [38] and packet switching [39] techniques. By using these techniques, multiple neurons can share the same physical computing components to save the silicon area; the neurons and astrocytes communicate with each other by routing and transferring the packets. This optimization is not explored in this paper as the key focus is on exploring the self-repair dynamics in hardware.

2) Astrocyte Facility: The *astrocyte facility*, shown in Fig. 4, receives the input of 2-AG from the postsynaptic neurons, and drives the signal of e-SP. After the *astrocyte facility* receives the 2-AG, the first step is to generate IP$_3$ [see (2)] which is calculated by the IP$_3$ generator; and the value of IP$_3$ is updated by the IP$_3$ register. Then the released IP$_3$ binds to IP$_3$ on the ER, which triggers the release of Ca^{2+}. Three channels, J_{chan}, J_{leak}, and J_{pump} [see (3)], are used to model the Ca^{2+} dynamics within the astrocyte cell. These components are highlighted in yellow in Fig. 4. The Ca^{2+} generator and Ca^{2+} register are the core component (highlighted in blue in Fig. 4) to calculate the quantal release of Ca^{2+}. The Ca^{2+} generator receives the signals from J_{chan}, J_{leak}, and J_{pump} components, calculates the change value of Ca^{2+}; and the Ca^{2+} register stores and updates the Ca^{2+} value.

For the three channels of J_{leak}, J_{pump}, and J_{chan}, the J_{leak} and J_{pump} calculator components receive the input signal of Ca^{2+} from the Ca^{2+} register and calculate the values of J_{leak} and J_{pump} for the Ca^{2+} generator. In comparison to the channels of J_{leak} and J_{pump}, the calculation of J_{chan} is more complex. The J_{chan} generator has four inputs: Ca^{2+}, m_∞ (i.e., IICR), n_∞ (i.e., CICR), and h (i.e., the fraction of IP$_3R_s$). The first input Ca^{2+} is from the Ca^{2+} register. The second input m_∞ is generated by the m_∞ calculator whose input signal IP$_3$ comes from the IP$_3$ register. The third input n_∞ is generated by the n_∞ calculator whose input signal, Ca^{2+}, comes from Ca^{2+} register. The fourth input h is generated by the h register. Equation (4) illustrates that h (i.e., the fraction of activated IP$_3R_s$) is calculated from τ_h and h_∞. In the *astrocyte facility*, τ_h is generated by the τ_h calculator whose input signal, Ca^{2+}, is from the Ca^{2+} register. The other value of h_∞ is generated by the h_∞ calculator which has two inputs [see (5)]-Ca^{2+} and Q_2. The Ca^{2+} input is from the Ca^{2+} register, and Q_2 is generated by the Q_2 calculator whose input signal IP$_3$ comes from the IP$_3$ register. Using the aforementioned Q_2, h_∞, and τ_h calculators, the change of h (i.e., Δh) is calculated by the h generator. Then h is updated by the h register and sent to the J_{chan} generator. Based on all four inputs (Ca^{2+}, m_∞, n_∞, and h) the J_{chan} generator calculates the value of J_{chan} which is output to the Ca^{2+} generator.

After the Ca^{2+} generator receives all the input values of J_{leak}, J_{pump}, and J_{chan}, it calculates the change in Ca^{2+} (i.e., ΔCa^{2+}) and sends it to the Ca^{2+} register for updating. The change of Ca^{2+}, i.e., calcium dynamics, releases the glutamate inside the astrocyte, and then leads to the e-SP. In the glutamate and e-SP generation processes, the Ca^{2+} spike generator in the *astrocyte facility* describes the behaviors of calcium dynamics; the glutamate generator and its corresponding register calculates and updates the quantity of released glutamate (14); similarly the e-SP generator and corresponding register calculates and updates the released e-SP (15), which is the output signal of the *astrocyte facility*.

The output signals of e-SP (from the astrocyte facility) and DSE (from neuron facility) feedback to the synapse facilities, which regulate the PRs of presynaptic terminals. More details are given in next section. In addition, research work of [40] showed that the astrocyte communicates with large number of synapses ($\sim 10^5$) and several neurons (~ 8) in the human brain. Thus for a large scale spiking neural network, only a small number of *astrocyte facilities* are required, e.g., six *neuron facilities* for every one *astrocyte*. However, the astrocytes can

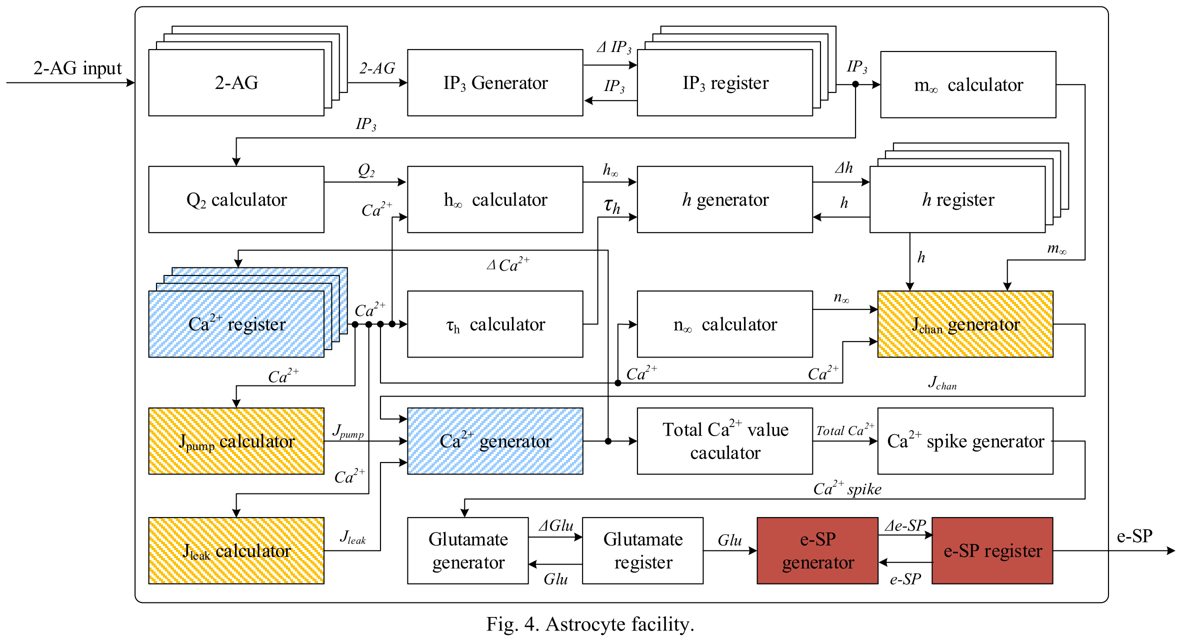


Fig. 4. Astrocyte facility.

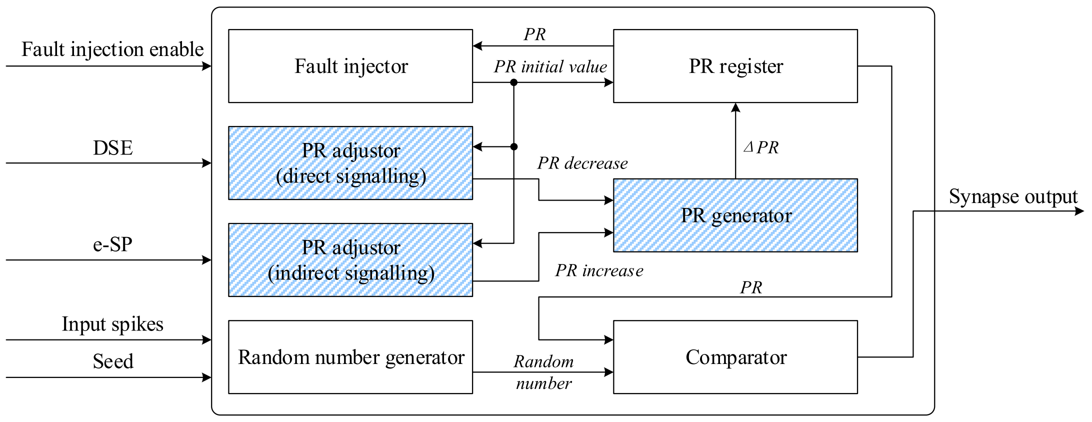


Fig. 5. Synapse facility.

greatly enhance the fault-tolerant capabilities of the systems and still maintain the scalability due to the low ratio of requirements. In addition, Carrillo *et al.* [39] proposed a hierarchical Networks-on-Chip (H-NoC) diagram in previous work to address the interconnection challenge within the spiking neurons, and an extended hierarchical astrocyte network architecture in [41] to provide information exchanges between astrocyte cells and neurons. These interconnected strategies can be employed to support the interconnectivity and scalability of the astrocyte-neuron hardware systems.

3) Synapse Facility: The internal structure of *synapse facility* is shown in Fig. 5, which is based on a probabilistic synapse model. The input signals include input spikes, DSE signal (from neuron facility), e-SP (from astrocyte facility), and a seed signal which controls the random number generation. Additionally, a fault injection enable signal controls the fault injection to the synapse. If it is active, the synapse is set to be faulty by reducing the transmission PR to zero. Only one output (i.e., synapse output) is associated with the *synapse facility*, and it connects to the component of *neuron facility*.

Each synapse has an initial transmission PR, e.g., it is 0.5 in the research work of [22]. However, if faults occur at *synapse facility*, the PR goes to very low. In this approach, it is simulated by enabling the fault injection signal in Fig. 5 and the PR is set to be zero. The PR value is stored in the PR register component which connects to a comparator. The PR value is compared with a random number. If the PR is larger, then the *synapse facility* has a valid output; otherwise invalid output is presented, see (17). A random number generator [42] is used for this process, which only generates a random number when the input spike is presented.

If no faults occur at *synapse facility*, the synaptic PRs are controlled by the DSE and e-SP. Two PR adjustors, highlighted by blue in Fig. 5, are used to change the PR. The PR adjustor that connects to the DSE is a direct feedback, and reduces the PR value; the other PR adjustor associated with the input of e-SP increases the PR value. The PR generator component

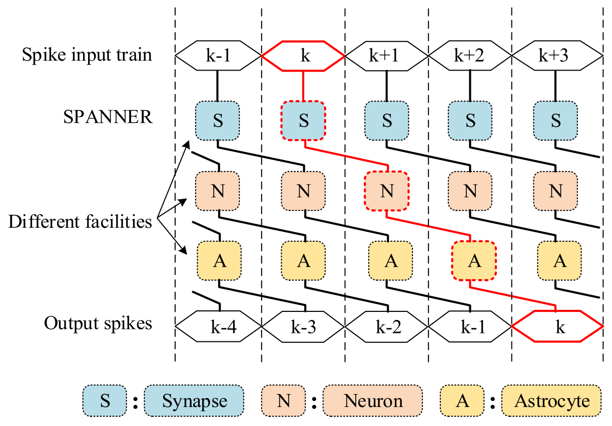


Fig. 6. Pipelining strategy at the system level in the SPANNER.

takes these two signals as inputs, then calculates the change of PR, i.e., ΔPR, and outputs it to the PR register component for the update. This process modulates the PRs and implements the self-repairing mechanism, e.g., when faults occur, the PR adjustor associated with e-SP will enhance the transmission PRs of the healthy synapses which can help the neuron maintain the target output. Extensive experiments are carried out in Section III to demonstrate the fault-tolerant capabilities of the proposed hardware SPANENR.

D. Optimized Design of the SPANNER

In this section, the optimization of the SPANNER hardware at system and facility levels are discussed. First, the input spike processing rate (SPR) is improved using a pipeline strategy at the system level. A high input SPR is beneficial as the system processing capability is improved. In the SPANNER hardware system each facility depends on a set of data as inputs where they are only available after the computing of the previous facility calculations. Consequently, there is a data dependence between the three facilities, e.g., the astrocyte facility depends on the neuron facility, which requires the output data from the synapse facility. Assume that the processing times of the synapse, neuron, and astrocyte facilities are t_s, t_n, and t_a, respectively; then the input SPR before pipelining is given by

$$\text{SPR} = 1/(t_s + t_n + t_a) \tag{19}$$

where SPR is equal to the reciprocal of the total processing time of three facilities. For each spike input train data, the total processing time directly governs the input SPR. A pipeline strategy is employed in this approach, which can break the data dependencies and permit a set of operations to be grouped together, and therefore allow different facilities to execute in parallel and improve the processing rate of the input spike train. Fig. 6 shows the pipelining strategy used at the system level. It can be seen that three facilities are running in parallel, e.g., the synapse facility is processing the kth input spike, the neuron and astrocyte facilities are processing the $(k-1)$th and $(k-2)$th input data, respectively. Thus, SPANNER hardware can process input spike data more quickly. The input SPR after pipelining (SPR') is given by

$$\text{SPR}' = 1/\max(t_s t_n t_a) \tag{20}$$

where $\max(t_s t_n t_a)$ denotes the maximum value of the processing time among the three facilities. It can be seen that $\text{SPR}' > \text{SPR}$ which indicates the pipelining strategy at system level achieves a faster SPR.

Second, the SPANNER hardware can be further optimized at the facility level to reduce the total process time (i.e., latency). Section II-C presents the architectures of these facilities. It shows that the astrocyte facility has several operations that can be executed in parallel (e.g., J_{leak}, J_{pump}, and J_{chan} calculators etc.); thus it can also be optimized using the pipelining strategy. Therefore, the pipelining strategy is also applied to the astrocyte facility to increase the concurrence and reduce the total processing time. The detailed results using the pipelining strategies at the system and facility levels are provided in Section III.

III. Experimental Results

The test bench setup is presented first in this section, and then the detailed experimental results are provided for the proposed SPANNER hardware system. In the experiments, the self-repair mechanism is evaluated and verified in the real-time FPGA hardware platform. The dynamic behaviors of synapses under different level faulty scenarios are discussed. This section also gives the results of hardware performance analysis of the proposed SPANNER, including the FPGA hardware and software simulation performance comparison, required computing time of different solutions, and resource cost analysis.

A. Testbench Setup

An example SPANNER FPGA implementation was created using the neuron, astrocyte and synapse facilities shown in Fig. 7. In this spiking neuron network, it has one astrocyte and two neurons where each neuron is associated with ten synapses. The initial PRs for all the synapses are 0.5, and the average frequencies for the input spike trains are 10 Hz. As shown in the synapse facility, every synapse has a fault injection enable input. A fault is injected into the synapse when the enable signal is set high. When a neuron spikes, 2-AG is generated which initiates the creation of DSE (from the neuron facility) and e-SP (from the astrocyte facility). These signals are fed back to the synapses and influence the preterminal transmission PR values. All the components are modularized and parameterized, and implemented on FPGA hardware as illustrated in Fig. 7.

The proposed SPANNER system was developed using VHSIC Hardware Description Language and based on a Xilinx Virtex-7 FPGA VC707 evaluation kit, which includes a Virtex-7 XC7VX485T-2FFG1761C FPGA device. For all experiments the following setup was used: 200 MHz system frequency, the user interface in the evaluation board (e.g., rotary switch, buttons, and so on) was used for injecting faults of different types (e.g., permanent/temporary faults). The failure model in [43] is employed for this approach. If the synapses are faulty, the failures result in permanent malfunctions of the circuit blocks, i.e., they fail to respond to the spikes completely and stop generating any activations. Thus in this approach, if a

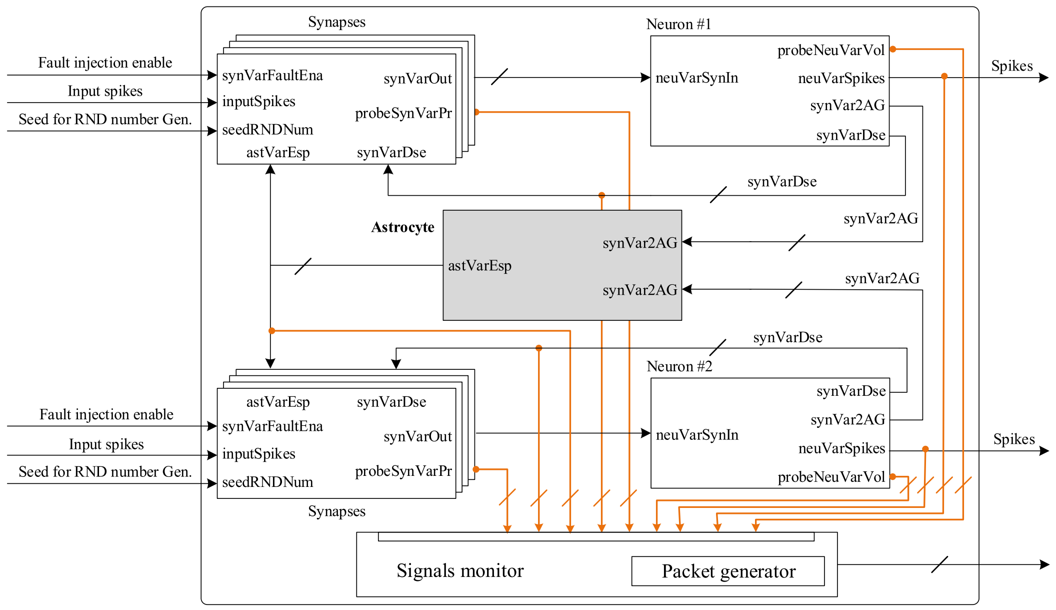


Fig. 7. SPANNER FPGA hardware with one astrocyte and two neurons.

synapse is faulty, it is modeled as a dead/broken synapse and the corresponding PR is set to zero. The SPANNER hardware implementation used the forward Euler method of integration to solve the equations of our model and the IEEE 754 floating-point data format with double-precision (i.e., 64 b) is used for the numerical representation. This permits the same numerical accuracy as the software simulations of the computer program.

In addition, a similar monitoring framework from previous work in [44] was employed to capture data during runtime execution on the FPGA. For example, a signal monitor component was designed which can probe the signals between the facilities (e.g., neuron, astrocyte, and synapse) and collect the data in real-time. The signal lines shown in Fig. 7 (colored yellow) were monitored during hardware execution. The values of these signals are collected and uploaded to the desktop computer in real-time for performance analysis.

Based on the SPANNER system in Fig. 7, the experimental results are given in the next sections, which demonstrate dynamic behaviors of the SPANNER and how self-repair can occur at the synapse facilities. Note that the experimental data was collected using the monitoring mechanism, however, the data was analyzed and visualized using the MATLAB R2014a software.

B. Results With no Faults, 40% and 80% Partial Faults

In the SPANNER system of Fig. 7, ten synapses are connected to each of the two neurons, which is similar to the diagram shown in Fig. 2. In this approach, the fault density denotes the percentage of faulty synapses in a synapse group, e.g., a fault density of 80% means eight synapses are faulty in a ten-synapse group. When the synapse is faulty, its PR is set to zero. In the experiments, faults occur to the synapses associated with neuron #2. The fault densities of 0%, 40%, and 80% were used to test the self-repairing capabilities of the SPANNER system. The hardware signal monitor in Fig. 7 is

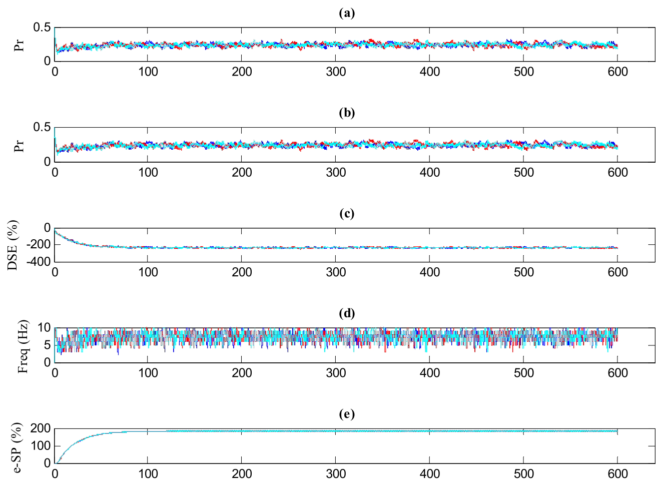


Fig. 8. Neuron #1 in the SPANNER. Color codes represent the results of neuron #1 when the fault densities of neuron #2 is: no fault (dark blue), 40% (red), and 80% (light blue).

used to observe all the signals inside the FPGA device and transmits the results to the computer. The experiment runs for a period of 600s which permits the repair process to complete as outlined in the approach of [22].

In first experiment, the fault density is 0%, i.e., no fault occurs. The results of neuron #1 is shown in Fig. 8 (dark blue). Note that Fig. 8 also shows the results of neuron #1 under other fault conditions. In the current experiment (e.g., the fault density is 0%), the results are shown in dark blue. The results of neuron #2 are not given due to the limited space, but the profiles are similar to neuron #1. When the presynaptic stimuli presents, both neurons fire. It occurs while coupling with astrocyte process via 2-AG signal pathway. Two synapses from neuron #1 (i.e., the first and tenth) are chosen to illustrate the transmission PR values. From Fig. 8, it can be seen that when the fault density is 0%, the first and tenth synapse have similar

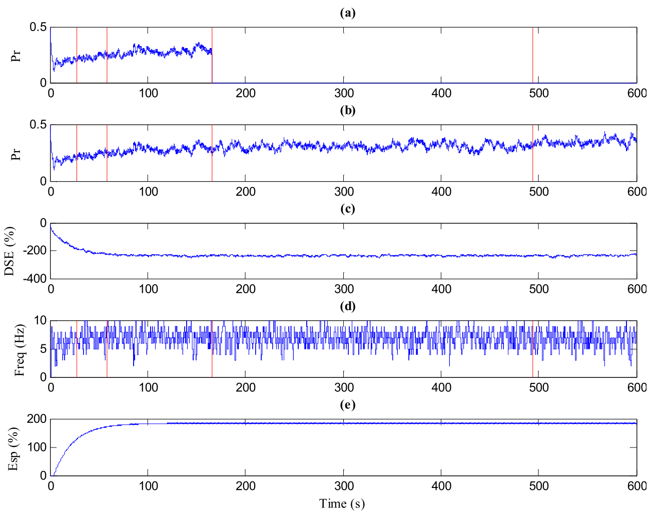


Fig. 9. Neuron #2 under 40% fault density.

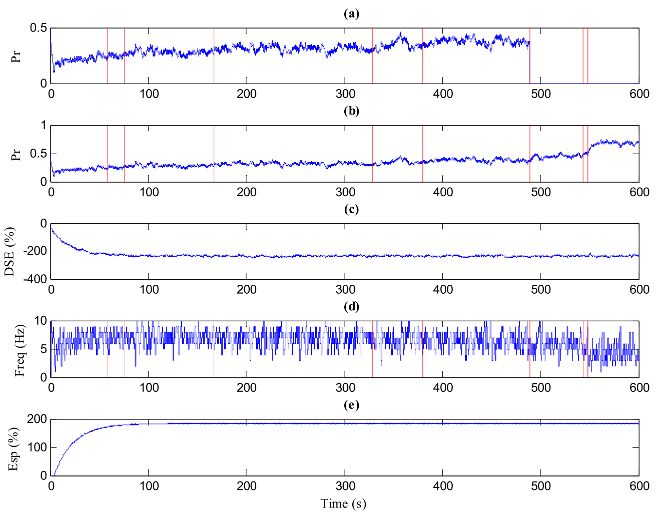


Fig. 10. Neuron #2 under 80% fault density.

profiles for the PRs, i.e., both the initial and average values are ~0.5 and ~0.25, respectively. During the initial stage, after the e-SP and DSE are generated and modulate the synapse transmission, the PR values reduce from ~0.5 to ~0.25. The DSE values are not exactly the same for individual neurons. However, the e-SP, as a global signal for both neurons, is the same for neuron #1 and #2. Their output frequencies (i.e., the average firing rate of neurons) are also included, which are similar (~7 Hz) and correctly match the results achieved from SPANNER software model [22].

For next experiment, the fault density is increased to 40%, i.e., 40% synapses of neuron #2 (i.e., four synapses) are set to fault conditions where the associated PR values are reduced to zero. The plots in red in Fig. 8 present the results for neuron #1. It can be seen that neuron #1 and associate synapses, as expected, are unaffected. However, the behavior of neuron #2 is different. For a fault density of 40% we set four of the ten synapses of neuron #2 to have a zero PR value; e.g., the first 4 synapses are faulty and synapses 5–10 are healthy. The time to damage the synapses are distributed randomly, e.g., in this experiment the first to fourth synapses are damaged at 166, 277, 58, and 494 s, respectively. Fig. 9 plots the results for neuron #2 and its first and tenth synapses. The red vertical lines in the Fig. 9(a), (b), and (d) depicts the fault injection time, e.g., in Fig. 9(a) the first synapse is faulty after 166 s and the PR is zero after that time point. The PR of the third synapse drop to zero from 58 s, i.e., injecting faults. And from 166, 277, and 494 s, other three synapses (i.e., first, second, and fourth synapses) are damaged sequentially. It can be seen that from 58 s, the faulty synapse lead to a reduced firing rate of neuron #2 depicted in Fig. 9(d) which means the output frequency of neuron #2 experiences a slight decrease. This zero PR on the third faulty synapse reduces the associated DSE signal [see Fig. 9(c)]; then causes an imbalance in neuron cell dynamic and e-SP enhances the healthy synapses (5–10) associated with neuron #2; therefore the PR values of the healthy synapses are increased [e.g., tenth synapse is only shown in Fig. 9(b) due to space], in order to maintain the dynamic balance or output of the neuron. As a result, the average output frequency of neuron #2 gradually returns back to its original prefault value after a period of time. The same process applies to other faulty synapses. Therefore, the PR of healthy synapses increased after the synapse is faulty and the output of neuron #2 is maintained. In summary, when faults occur in some synapses, the process to maintain the neuron average firing rate through redistribution of PRs across all the synapses, is considered as the self-repairing mechanism. This experiment demonstrates this capability via the reestablishing of the average firing rate. The advantage of this self-repairing mechanism is further demonstrated through a more critical condition when a higher fault density of 80% is used.

With a fault density of 80% only two healthy synapses associated with neuron #2 (synapse 9 and 10) remain. The runtime execution in hardware is depicted in Fig. 8 (light blue) for neuron #1 and in Fig. 10 for neuron #2. Fig. 8 shows that neuron #1 is still unaffected when neuron #2 experiences the failures. However, for neuron #2, the first 8 synapses are damaged with a random time-distributed fault injection strategy which is shown by red vertical lines in Fig. 10. It can be seen that the more synapses which are damaged, the greater the increase in the strength of the remaining healthy synapses. The PR values of healthy synapses increase significantly [e.g., tenth synapse in Fig. 10(b) is only shown] after final fault injection time (i.e., 548 s), which are much larger than the PR values when the fault density is 40%. When the fault density is 40%, the enhanced PR value of healthy synapse is ~0.3; however, when the fault density is 80%, the enhanced PR value is ~0.7. The increasing action of the PR values demonstrates the repair process by the astrocyte where the PRs of healthy synapses are increased to compensate for the loss. Fig. 10 also shows that the average firing rate of neuron #2 has a slight decrease after final fault injection time (i.e., 548 s). However, this cannot be seen as a disadvantage of the proposed self-repairing mechanism. The healthy synapses are enhanced enough to maintain the system performance; however, as the majority of synapses (80%) associated with neuron #2 are

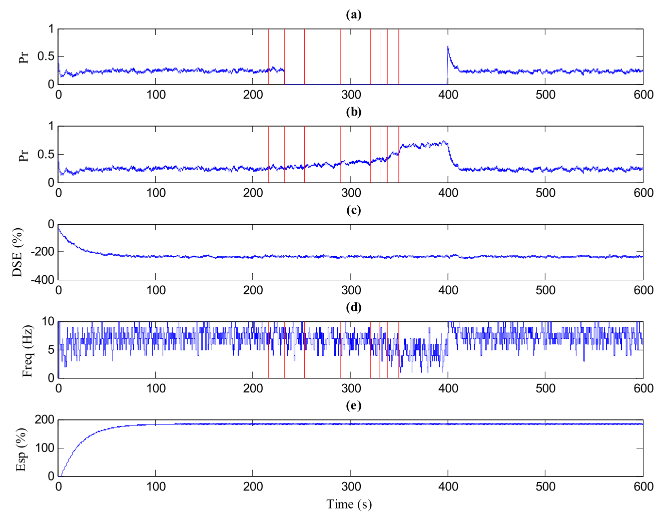


Fig. 11. Neuron #2 under 80% fault density with temporary faults.

faulty, there is a marginal performance degradation which is a reasonable expectation given the significant high fault density. The analysis of performance degradation is discussed further in Section III-D.

C. Temporary Faults

The previous section presented the results under various fault densities that were injected to the system permanently, i.e., exist continuously after they occur. Besides the permanent faults, temporary faults are also very common in electronic systems [4], [11], where they only exist for a period of time and disappear again. In the following experiment, the faults are injected temporarily between time 200 and 400 s into synapses 1–8 associated with neuron #2 (e.g., fault density is 80%). The fault injection time for the eight synapses are random, but all the fault injections are stopped at time 400 s (i.e., after that the PRs are set to initial values). Fig. 11 depicts the results from neuron #2. The first synapse, shown by Fig. 11(a), is one of the faulty synapse. Its PR is zero between time 233 s (i.e., fault injection time) and 400 s, which simulates a temporary fault. After the faults disappear at 400 s, the PRs of faulty synapses return to their prefault values. It can be seen that when the temporary faults occur, the PRs of healthy synapses, e.g., tenth synapse of neuron #2, are increased via the e-SP feedback from the astrocyte facility; and the neuron output frequency returns to 7 Hz gradually. Thus the self-repair process causes these PRs to increase with the aim to maintain the neuron target output as close as possible. In the time period of faults occur, the average firing rate of neuron #2, shown by Fig. 11(c), decreases slightly especially when all 8 synapses are damaged; however, this is due to the significant level of faults injected. In this experiment, the results clearly show that the proposed SPANNER hardware system can self-adopt and self-repair under various number of synapses that are faulty at different times. Therefore the results in the Sections of III-B and III-C demonstrate that the SPANNER has an efficient self-repairing capability for different fault levels and conditions.

TABLE I

AVERAGE FREQUENCIES OF DIFFERENT PLATFORMS UNDER VARIOUS FAULT DENSITIES (UNIT: Hz)

Fault density	Platform	Average output frequency	
		Neuron 1	Neuron 2
0%	Simulation	7.19	7.20
	Hardware	**7.28**	**7.27**
40%	Simulation	7.38	6.81
	Hardware	**7.37**	**6.88**
80%	Simulation	7.38	5.68
	Hardware	**7.37**	**5.75**

D. Performance Comparison Between Hardware System and Software Simulation

This section provides a comparison between the hardware system and software simulation which aims to give a fair evaluation of the proposed SPANNER hardware system. The hardware results are based on a Xilinx FPGA, see Section III-A for detail. The software simulation is running MATLAB R2014a on the desktop computer with an i7-2600 3.4-GHz processor and 4-GB memory. The neuron average frequencies are the outputs of the entire system and therefore reflect the performance of the proposed self-repairing mechanism. They are used for the comparison between the hardware system and computer software simulations. The average frequency is the mean value of neuron firing. All the average frequencies are calculated based on the hardware and software data reported in Section III-B.

Table I gives the average frequencies of different platforms, i.e., software simulation approach [22] and hardware implementation in this approach, under various fault densities. It can be seen that the software simulation and hardware system are consistent; e.g., between 0.01-and 0.09-Hz difference. It verifies that the proposed SPANNER hardware implementation achieves a good accuracy with the software simulation model of [22]. Table I also provides a quantitative analysis of the proposed self-repairing mechanism under different fault densities. It can be seen that the average frequency of neuron #1 maintains the same level (7 Hz) no matter what the fault density. However, for neuron #2, the average frequency has a slight degradation when the fault density is high, e.g., having a 5% and 20% degradation for 40% and 80% fault densities, respectively. When the fault density is 40%, the degradation of average frequency is only 5% which means the proposed SPANNER hardware can repair and reach an acceptable system performance. When the fault density increases to 80%, the degradation is 20%. However, it should be noted that with such a high fault density, this amount of frequency degradation is expected, as the conventional fault-tolerant mechanism for electronic systems normally fail completely when the fault density is greater than ~35% [3], [4]. SPANNER hardware can provide graceful degradation.

E. Performance Analysis of the SPANNER System

The performance analysis in terms of SPR, required computing time and resource cost are reported in this section. In the SPANNER hardware a pipelining strategy was implemented

TABLE II
REQUIRED COMPUTING TIME OF DIFFERENT SOFTWARE AND HARDWARE IMPLEMENTATIONS

Platform/Implementation	Time required [second]
Biological neural network	600
Software simulation	64
SPANNER **without** optimized astrocyte facility	58
SPANNER **with** optimized astrocyte facility	7

TABLE III
HARDWARE UTILIZATION OF DIFFERENT COMPONENTS

	Astrocyte	Synapse & Neuron	DSE generator
Slice LUTs	11394	9865 [12.8%]*	4353
Slice Registers	11666	10383 [17.4%]	4688
Slice	3552	3120 [15.4%]	1394
DSPs	42	45 [125%]	14

* The percentage denotes how many additional resources are required for the synapse & neuron component to implement the self-repairing mechanism.

at the system level to increase the input SPR. When we compare the hardware implementation with and without the system level pipelining strategy, the SPR and SPR′ are 39 and 58 MHz, respectively. Thus the pipelining strategy at the system level achieves a ∼49% higher SPR. The pipelining strategy was also employed at the astrocyte facility to optimize reduction of the total SPANNER processing time. The results in Table II are the required computing times of SPANNER with/without the optimized astrocyte facilities. It can be seen that if simulating 600 s of the biological spiking neural network, the SPANNER hardware with and without the optimized astrocyte facility exhibits 58 and 7 s runtime, respectively. Therefore, the optimized astrocyte design achieves a ∼88% hardware speed improvement.

The required simulation time is also given in Table II which is 64 s. Comparing the SPANNER software execution time with the hardware demonstrates that the hardware is approximately nine times faster. It should be noted that the main contribution of this approach lies in the self-repairing capability in hardware using the astrocytes cells merged within SNNs. However, as the SPANNER hardware is a parallel neuron computing system, it can achieve a faster computing speed than a software simulation. This accelerated computing capability has the potential, with further developments, for simulating a large-scale astrocyte-neuron system in real time [45], [46].

Hardware resource occupied by different facilities in the SPANNER is given by Table III. The results include the astrocyte facility, the synapse and neuron facilities, and a DSE generator component. The results show that the largest component in SPANNER is the astrocyte facility, which is expected as its model is more complex than others. The resource occupied by synapse and neuron facilities is less than astrocyte, and the resource cost of the DSE generator is the lowest. Table III also gives the percentage of additional resources for the synapse & neuron component to implement the proposed self-repairing capability, e.g., the required numbers of the slice LUTs for the synapse & neuron component with/without self-repair are 11 128, and 9865, respectively, thus the percentage of additional required resources is 12.8% (11 128-9865/9865 = 12.8%). The overhead for the slice, slice LUTs/Registers is less than 18% which is generally acceptable for the fault-tolerant mechanism [4]. The overhead for the DSPs is 125% due to the PR modulation process in (18) and this can be reduced by the further hardware optimization. As the aim of this paper is to demonstrate the principle of biological self-repair of astrocyte-neuron network in hardware devices, the hardware optimization is out of scope. However, a possible solution is discussed in our research work of [47], which simplifies the calcium dynamics within the astrocyte cells and achieves a compact hardware area for the astrocyte-neuron network.

IV. DISCUSSION

The results in Section III demonstrated that if the synapses associated with neuron #2 are damaged with various fault level densities (e.g., 40% and 80%), the average frequency of the neuron can be maintained. Furthermore, an experiment evaluated the self-repair capability of the proposed system if faults occur simultaneously for the input synapses of both neurons, i.e., the synapses of both neuron #1 and #2 are damaged. In this experiment, the synapses associated with both neurons are damaged with 80% fault densities. The results showed that the average frequencies of both neurons decrease from ∼7 to ∼5 Hz in a similar profile as Fig. 10. It demonstrated that even when the inputs of both neurons are faulty, the proposed SPANNER still has the self-repair capability due to the PR enhancements of the healthy synapses of both neurons which are modulated by the astrocyte cell.

For a large-scale SNN, using astrocyte cells to enhance the system fault-tolerant capability is more efficient than traditional fault tolerance techniques, which is analyzed as follows.

1) *The Astrocyte-Neuron Network Is an Online Self-Detecting and Self-Repairing System:* The traditional techniques such as scan chain or built-in self-test only test for faults before power-up. This is a drawback of such offline testing, especially for mission critical electronic systems [11]. Unlike these techniques, SPANNER is an online fault detection mechanism and the faults can be detected while the system is running. The regular application is not interrupted for fault testing, i.e., the astrocyte cell does not introduce an intrusion for the fault detection process. This inherent fault-tolerant capability (e.g., the online fault testing with nonintrusion) provides an alternative solution to enhance the resilience of electronic systems. In addition, this testing and repair capability is fault-tolerant itself. For example, glia cells communicate with other glia cells, and therefore provide a distributed detection capability [21]. There is no one central controller which can be compromised due to faults, unlike Built-In Self-Test approaches.
2) *Scalability Analysis:* The area overhead of the astrocyte cell in SPANNER is used to analyze the efficiency regarding the occupied resource of the astrocyte cell. As mentioned in Section II, the astrocyte cell generally enwraps ∼8 neurons and ∼10^5 synapses. Therefore

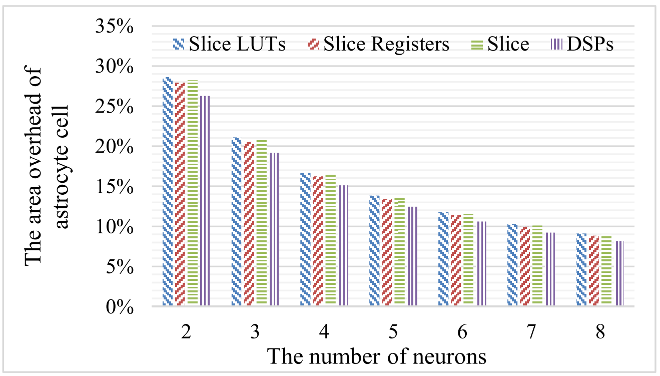


Fig. 12. Relative area overhead of astrocyte cell for the SPANNER with different number of neurons.

Fig. 12 gives the results regarding the area overheads of the astrocyte cell for the system with different numbers of neurons (up to 8), e.g., if two neurons are included in the system, the percentage of slice LUTs occupied by astrocyte cell is 28.6%. Fig. 12 shows that when the number of neurons increases from 2 to 8, the area relative overhead of the astrocyte cell decreases from ∼28% to ∼9%. The area overhead of astrocyte cell (less than 10%) for one neuron group (eight neurons) is acceptable for the online fault-tolerant computing system [4]. A typical SNN includes large numbers of neurons and synapses, e.g., the TrueNorth chip has 1 million digital neurons that communicate with each other via 256 million synapses [48]. For these large-scale neuron networks, it is impossible to apply the traditional techniques (e.g., triple modular redundancy) to tolerant the faults, as large area overhead is introduced due to the replication of the same modules. However, the area required by the astrocyte cells in SPANNER increases linearly with the number of neuron groups (e.g., eight neurons for each group). Note that for each neuron, only ten synapses are connected for the calculations in Fig. 12. If the number of synapses is much larger, the relative area overhead of the astrocyte cell is further reduced. This work used a small neural network as early exploration to study the self-repairing mechanism. Increasing the number of neurons can give more scope to provide high levels of reliability as there are more healthy synapses and neurons in the system which allow repair to occur.

In addition, reliability is also a critical issue during the learning process in the neural network. Future work will investigate how to model the astrocyte-neuron network with self-learning and self-adapting capabilities, and apply it to real-world applications, e.g., pattern recognition tasks [49], [50], robotic applications [51]. Our previous SWAT training algorithm [52] and B-STDP learning rule [21] in the SNN will be employed to explore this aim.

V. Conclusion

A self-repairing spiking neural network hardware architecture, i.e., SPANNER, has been proposed in this paper which emulates the fault-tolerant capability of the brain. It includes three key components of neuron, synapse and astrocyte facilities. Based on these components, an astrocyte-neuron hardware system has been implemented. Extensive experiments were designed to evaluate and verify the system performance, e.g., under various fault densities and different fault types. The hardware verification results and resource costs were also provided. They demonstrated that compared with conventional fault-tolerant approaches, the proposed SPANNER exhibited the fine-grained self-adopt and self-repair capabilities for the hardware electronic devices.

This research work can be used to aid in the development of a large scale bio-inspired computing platform [39], [53], [54]. The results in this paper are promising and provide a new research direction for the brain-inspired self-repairing mechanism using SNNs. Although defining the SPANNER architecture is important, the efficient interconnection for large-scale astrocyte-neuron systems should be also considered and achieved. Thus future work will consider the efficient interconnection of astrocyte-neurons coupling hardware system, and the hardware optimizations.

References

[1] T. Karnik, P. Hazucha, and J. Patel, "Characterization of soft errors caused by single event upsets in CMOS processes," *IEEE Trans. Depend. Sec. Comput.*, vol. 1, no. 2, pp. 128–143, Jun. 2004.

[2] H. Chang and S. S. Sapatnekar, "Statistical timing analysis under spatial correlations," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 24, no. 9, pp. 1467–1482, Sep. 2005.

[3] 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 Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 21, no. 6, pp. 1053–1066, Jun. 2013.

[4] J. Liu, J. Harkin, Y. Li, and L. P. Maguire, "Fault-tolerant networks-on-chip routing with coarse and fine-grained look-ahead," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 35, no. 2, pp. 260–273, Feb. 2016.

[5] B. Pratt, M. Caffrey, J. F. Carroll, P. Graham, K. Morgan, and M. Wirthlin, "Fine-grain SEU mitigation for FPGAs using partial TMR," *IEEE Trans. Nucl. Sci.*, vol. 55, no. 4, pp. 2274–2280, Aug. 2008.

[6] K. S. Morgan, D. L. McMurtrey, B. H. Pratt, and M. J. Wirthlin, "A comparison of TMR with alternative fault-tolerant design techniques for FPGAs," *IEEE Trans. Nucl. Sci.*, vol. 54, no. 6, pp. 2065–2072, Dec. 2007.

[7] K. Zhang, G. Bedette, and R. F. Demara, "Triple modular redundancy with standby (TMRSB) supporting dynamic resource reconfiguration," in *Proc. IEEE Autotestcon*, Sep. 2006, pp. 690–696.

[8] J. Johnson *et al.*, "Using duplication with compare for on-line error detection in FPGA-based designs," in *Proc. IEEE Aerosp. Conf.*, Mar. 2008, pp. 1–11.

[9] S. Mitra, W. J. Huang, N. R. Saxena, S. Y. Yu, and E. J. McCluskey, "Reconfigurable architecture for autonomous self-repair," *IEEE Design Test Comput.*, vol. 21, no. 3, pp. 228–240, May 2004.

[10] K. Reick *et al.*, "Fault-tolerant design of the IBM power6 microprocessor," *IEEE Micro*, vol. 28, no. 2, pp. 30–38, Mar. 2008.

[11] J. Liu, J. Harkin, Y. Li, and L. Maguire, "Online traffic-aware fault detection for Networks-on-Chip," *J. Parallel Distrib. Comput.*, vol. 74, no. 1, pp. 1984–1993, 2014.

[12] W. Barker, D. M. Halliday, Y. Thoma, E. Sanchez, G. Tempesti, and A. M. Tyrrell, "Fault tolerance using dynamic reconfiguration on the POEtic tissue," *IEEE Trans. Evol. Comput.*, vol. 11, no. 5, pp. 666–684, Oct. 2007.

[13] M. G. Negoita and S. Hintea, "Bio-inspired technologies for the hardware of adaptive systems," in *Studies in Computational Intelligence*. Springer, 2009, pp. 1–168.

[14] S. Ghosh-dastidar and H. Adeli, "Spiking neural networks," *Int. J. Neural Syst.*, vol. 19, no. 4, pp. 295–308, 2009.

[15] Q. Yu, R. Yan, H. Tang, K. C. Tan, and H. Li, "A spiking neural network system for robust sequence recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 3, pp. 621–635, Mar. 2016.

[16] H. Shayani, P. Bentley, and A. M. Tyrrell, "A cellular structure for online routing of FPGAs," in *Evolvable Systems: From Biology to Hardware*. Springer, 2008, pp. 273–284.

[17] S. Roy, A. Banerjee, and A. Basu, "Liquid state machine with dendritically enhanced readout for low-power, neuromorphic VLSI implementations," *IEEE Trans. Biomed. Circuits Syst.*, vol. 8, no. 5, pp. 681–695, Oct. 2014.

[18] L. E. Clarke and B. A. Barres, "Emerging roles of astrocytes in neural circuit development," *Nature Rev. Neurosci.*, vol. 14, no. 5, pp. 311–321, 2013.

[19] M. de Pittà , N. Brunel, and A. Volterra, "Astrocytes: Orchestrating synaptic plasticity?" *Neuroscience*, vol. 323, pp. 1–19, May 2015.

[20] B. Stevens, "Neuron-astrocyte signaling in the development and plasticity of neural circuits," *Neurosignals*, vol. 16, no. 4, pp. 278–288, 2008.

[21] 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 Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 10, pp. 2370–2380, Oct. 2015.

[22] 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 Comput. Neurosci.*, vol. 6, no. 76, pp. 1–12, Jan. 2012.

[23] 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, Sep. 2015.

[24] M. Hayati, M. Nouri, S. Haghiri, and D. Abbott, "A digital realization of astrocyte and neural glial interactions," *IEEE Trans. Biomed. Circuits Syst.*, vol. 10, no. 2, pp. 518–529, Apr. 2016.

[25] E. M. Izhikevich, "Simple model of spiking neurons," *IEEE Trans. Neural Netw.*, vol. 14, no. 6, pp. 1569–1572, Nov. 2003.

[26] R. FitzHugh, "Impulses and physiological states in theoretical models of nerve membrane," *Biophys. J.*, vol. 1, no. 6, pp. 445–466, 1961.

[27] 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," *J. Biol. Phys.*, vol. 35, no. 4, pp. 425–445, 2009.

[28] H. Soleimani, M. Bavandpour, A. Ahmadi, and D. Abbott, "Digital implementation of a biological astrocyte model and its application," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 1, pp. 127–139, Jan. 2015.

[29] 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.

[30] S. Nazari, K. Faez, M. Amiri, and E. Karami, "A digital implementation of neuron–astrocyte interaction for neuromorphic applications," *Neural Netw.*, vol. 66, pp. 79–90, Jun. 2015.

[31] J. Liu, J. Harkin, L. Maguire, L. McDaid, J. Wade, and M. McElholm, "Self-repairing hardware with astrocyte-neuron networks," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2016, pp. 1350–1353.

[32] A. Araque, V. Parpura, R. P. Sanzgiri, and P. G. Haydon, "Tripartite synapses: Glia, the unacknowledged partner," *Trends Neurosci.*, vol. 22, no. 5, pp. 208–215, 1999.

[33] Y.-X. Li and J. Rinzel, "Equations for InsP3 receptor-mediated calcium oscillations derived from a detailed kinetic model: A Hodgkin-Huxley like formalism," *J. Theor. Biol.*, vol. 166, no. 4, pp. 461–473, 1994.

[34] M. de Pittà, V. Volman, H. Levine, and E. Ben-Jacob, "Multimodal encoding in a simplified model of intracellular calcium signaling," *Cognit. Process.*, vol. 10, p. 55, Feb. 2009.

[35] W. Gerstner and W. M. Kistler, *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge, U.K.: Cambridge Univ. Press, 2002.

[36] M. Navarrete and A. Araque, "Endocannabinoids potentiate synaptic transmission through stimulation of astrocytes," *Neuron*, vol. 68, no. 1, pp. 113–126, 2010.

[37] W. Gerstner and R. Naud, "How good are neuron models?" *Science*, vol. 326, no. 5951, pp. 379–380, 2009.

[38] F. Akopyan *et al.*, "TrueNorth: Design and tool flow of a 65 mW 1 million neuron programmable neurosynaptic chip," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 34, no. 10, pp. 1537–1557, Oct. 2015.

[39] S. Carrillo *et al.*, "Scalable hierarchical Network-on-Chip architecture for spiking neural network hardware implementations," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 12, pp. 2451–2461, Dec. 2013.

[40] M. M. Halassa, T. Fellin, H. Takano, J.-H. Dong, and P. G. Haydon, "Synaptic islands defined by the territory of a single astrocyte," *J. Neurosci.*, vol. 27, no. 24, pp. 6473–6477, 2007.

[41] J. Liu, J. Harkin, L. P. Maguire, L. J. McDaid, J. J. Wade, and G. Martin, "Scalable Networks-on-Chip interconnected architecture for astrocyte-neuron networks," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 63, no. 12, pp. 2290–2303, Dec. 2016.

[42] S. K. Park and K. W. Miller, "Random number generators: Good ones are hard to find," *Commun. ACM*, vol. 31, pp. 1192–1201, Oct. 1988.

[43] A. Hashmi, H. Berry, O. Temam, and M. Lipasti, "Automatic abstraction and fault tolerance in cortical microachitectures," in *Proc. 38th Annu. Int. Symp. Comput. Archit. (ISCA)*, Jun. 2011, pp. 1–10.

[44] J. Liu, J. Harkin, Y. Li, L. Maguire, and A. Linares-Barranco, "Low overhead monitor mechanism for fault-tolerant analysis of NoC," in *Proc. IEEE 8th Int. Symp. Embedded Multicore/Many-Core Syst.*, Sep. 2014, pp. 189–196.

[45] S. B. Furber *et al.*, "Overview of the SpiNNaker system architecture," *IEEE Trans. Comput.*, vol. 62, no. 12, pp. 2454–2467, Dec. 2013.

[46] B. V. Benjamin *et al.*, "Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations," *Proc. IEEE*, vol. 102, no. 5, pp. 699–716, May 2014.

[47] A. P. Johnson *et al.*, "An FPGA-based hardware-efficient fault-tolerant astrocyte-neuron network," in *Proc. IEEE Symp. Ser. Comput. Intell.*, Dec. 2016, pp. 1–8.

[48] R. F. Service, "The brain chip," *Science*, vol. 345, no. 6197, pp. 614–616, Aug. 2014.

[49] Q. Yu, H. Tang, K. C. Tan, and H. Li, "Rapid feedforward computation by temporal encoding and learning with spiking neurons," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 10, pp. 1539–1552, Oct. 2013.

[50] J. Hu, H. Tang, K. C. Tan, H. Li, and L. Shi, "A spike-timing-based integrated model for pattern recognition," *Neural Comput.*, vol. 25, no. 2, pp. 450–472, 2013.

[51] J. Liu, J. Harkin, L. Mcdaid, D. M. Halliday, A. M. Tyrrell, and J. Timmis, "Self-repairing mobile robotic car using astrocyte-neuron networks," in *Proc. Int. Joint Conf. Neural Netw.*, Dec. 2016, pp. 1–8.

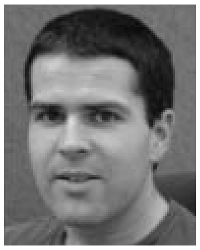
[52] J. J. Wade, L. J. McDaid, J. A. Santos, and H. M. Sayers, "SWAT: A spiking neural network training algorithm for classification problems," *IEEE Trans. Neural Netw.*, vol. 21, no. 11, pp. 1817–1830, Nov. 2010.

[53] S. Carrillo *et al.*, "Advancing interconnect density for spiking neural network hardware implementations using traffic-aware adaptive network-on-chip routers," *Neural Netw.*, vol. 33, no. 9, pp. 42–57, 2012.

[54] S. Pande *et al.*, "Modular neural tile architecture for compact embedded hardware spiking neural network," *Neural Process. Lett.*, vol. 38, no. 2, pp. 131–153, Jan. 2013.

Junxiu Liu (M'–) received the Ph.D. degree from Ulster University, Coleraine, U.K., with the support of a Vice-Chancellor's Research Scholarship.

His current research interests include neural glial system, and system-on-chip/network-on-chip embedded systems.



Jim Harkin received the B.Tech. degree in electronic engineering, the M.Sc. degree in electronics and signal processing, and the Ph.D. degree from Ulster University, Coleraine, U.K., in 1996, 1997, and 2001, respectively.

He is currently a Reader with the School of Computing and Intelligent Systems, Ulster University, Derry, U.K. He has authored over 80 articles in peer-reviewed journals and conferences. His current research interests include the design of intelligent embedded systems to support self-repairing capabilities and the hardware/software implementation of spiking neural networks.

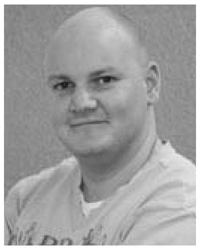
Liam P. Maguire received the M.Eng. and Ph.D. degrees in electrical and electronic engineering from the Queen's University of Belfast, Belfast, U.K., in 1988 and 1991, respectively.

He is currently the Dean of the Faculty of Computing and Engineering and also the Director of the Intelligent Systems Research Center, Ulster University, Coleraine, U.K. He has an established track record of securing research funding and has also supervised 15 Ph.D. and three M.Phil. students to completion. He has authored for over 200 research papers including 70 journal papers. His current research interests include fundamental research in bio-inspired intelligent systems and the application of existing intelligent techniques in different domains.

Liam J. McDaid received the B.Eng. (Hons.) degree in electrical and electronics engineering and the Ph.D. degree in solid-state devices from the University of Liverpool, Liverpool, U.K., in 1985 and 1989, respectively.

He is currently a Professor of Computational Neuroscience with Ulster University, Coleraine, U.K., and leads the Computational Neuroscience and Neural Engineering Research Team. He has co-authored over 120 publications in his career to date. His current research interests include modeling 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.

Dr. McDaid received several research grants in this domain and is currently a Collaborator on an HFSP and EPSRC funded project.



John J. Wade received the B.Eng. degree in electronics and computing, the M.Sc. degree in computing and intelligent systems, and the Ph.D. degree in computational neural systems from Ulster University, Coleraine, U.K., in 2004, 2005, and 2010, respectively.

He is currently a Researcher with the Intelligent Systems Research Center, Magee College, Ulster University, Derry, U.K., where he is with the Computational Neuroscience and Neural Engineering Research Team. His current research interests include developing computational models of neural and glial systems to aid understanding of how the brain functions and learns.