Assessing Self-repair on FPGAs with Biologically realistic Astrocyte-neuron Networks

Shvan Karim, Jim Harkin, Liam McDaid, Bryan Gardiner and Junxiu Liu
School of Computing and Intelligent Systems,
University of Ulster, Magee Campus,
Derry, Northern Ireland, UK, BT48 7JL
{haji_karim-s, jg.harkin, lj.mcdaid,b.gardiner, j.liu1}@ulster.ac.uk

David M. Halliday, Andy M. Tyrrell, Jon Timmis, Alan Millard and Anju Johnson
Department of Electronic Engineering,
University of York, Heslington,
York, UK, YO10 5DD
{david.halliday, andy.tyrrell, jon.timmis, alan.millard, anju.johnson}@york.ac.uk

***Abstract*— This paper presents a hardware based implementation of a biologically-faithful astrocyte-based self-repairing mechanism for Spiking Neural Networks. Spiking Astrocyte-neuron Networks (SANNs) are a new computing paradigm which capture the key mechanisms of how the human brain performs repairs. Using SANN in hardware affords the potential for realizing computing architecture that can self-repair. This paper demonstrates that Spiking Astrocyte Neural Network (SANN) in hardware have a resilience to significant levels of faults. The key novelty of the paper resides in implementing an SANN on FPGAs using fixed-point representation and demonstrating graceful performance degradation to different levels of injected faults via its self-repair capability. A fixed-point implementation of astrocyte, neurons and tripartite synapses are presented and compared against previous hardware floating-point and Matlab software implementations of SANN. All results are obtained from the SANN FPGA implementation and show how the reduced fixed-point representation can maintain the biologically-realistic repair capability.**

Keywords—Self-repair; Astrocytes; Spiking neural networks; FPGA; Bio-inspired computing

I. INTRODUCTION

Bio-inspired computing is seen as a potential replacement for traditional Von Neumann computer architectures [1]. This new trend in research is motivated by the fact that biology, in particular the human brain, is able to perform computations more efficiently in terms of power consumption and in a massively parallel manner. Contemporary, Spiking Neural Networks (SNN)s have been seen as a modern way to create such brain-like computing systems [2]. The ability of the human brain to perform self-repair is another significant feature that electronic systems designers are trying to mimic in the next generation of computer systems. It has been recently proposed that a specific type of glial cell, namely the astrocyte, is a key player in self-repair [3] as it is now believed they have an important role in modulating synaptic activity by mediating direct and indirect feedback to presynaptic and postsynaptic neurons [3], [4]. The interactions between astrocyte and neuron cells facilitate a distributed self-repair capability. Astrocytes enwrap groups of tripartite synapses and therefore are able to communicate with presynaptic and postsynaptic neurons at synaptic junctions [3], [5]–[7] which provides the pathways to realize repairs. The authors have previously demonstrated this repair capability of SANNs [3, 5]. In addition, prior work by the authors has demonstrated the repair principle in hardware using floating-point implementations [22]. Implementing such SANNs in hardware allows for future computing tasks to self-repair in hardware; thereby providing high degrees of reliability. This paper focuses on the feasibility of implementing such mechanism on FPGAs using reduced fixed-point representation, where biologically realistic accuracy can still be maintained. The paper is organized as following; Section II provides a review of current software and hardware implementations of SANNs. In Section III the software astrocyte model that is used for self-repair is discussed and a hardware equivalent is presented. Section IV provides results on a fixed-point SANN FPGA implementation using the astrocyte model and section V provides a conclusion with future work.

II. BACKGROUND

In this section, previous Astrocyte hardware and software implementations will be discussed.

A. Software implementations

An artificial neuron-glial network (NGN) has been explored for solving classification problems [8]. The astrocyte in this model is stimulated when the associated synaptic junction experiences neural activity for a minimum of *x* out of *y* iterations (4 out of 6, for example). When the astrocyte is active the synaptic weights increase by 25%, otherwise they decrease by 50%. The authors report that the NGN performance is significantly higher than that of traditional neural networks. Furthermore, astrocytes have been used in artificial NGNs to automate the generation of the values that are usually considered as constant parameters in neuron-glial models. Each neuron is connected to an astrocyte in this model and improvements have been reported over traditional Artificial Neural Networks (ANN)s and other NGNs in terms of solving classification problems [9]. Astrocytes have also been used to increase firing rates at tripartite synapses [10]. Here the internal calcium pool of the astrocyte is excited by two mechanisms. The first mechanism is a fast mechanism that is initiated by depolarization in postsynaptic neurons while the second one is a slow mechanism which is initiated by presynaptic diffusion. These values add to calcium volumes inside the astrocyte and once a threshold is reached, an astrocyte mediator is emitted that affects the synaptic terminal and increases the possibility of firing in the postsynaptic terminal. Finally, a Matlab software astrocyte model has been explored for providing traditional SNNs with a

978-1-5090-6762-6/17/$31.00 ©2017 IEEE

self-repairing capability [3]. The work presented in this paper proposes an implementation of the model in [3] using a fixed point representation and will act as the basis for designing an FPGA accelerator for SANNs of various network sizes.

B. Hardware implementations

HANA [11] is a hardware SANN in which the astrocytes have successfully been integrated into a hierarchical Network on Chip (HNoC) [1] architecture. In this network, the astrocytes are able to communicate with each other and with neurons. In [12] and [13] an NGN was implemented on FPGAs to demonstrate that digital circuits, especially FPGAs, can accurately reproduce NGN responses similar to software models in terms of accuracy and performance. Additionally, the authors claim to have approximated the neurons and the astrocyte models they use through piecewise linear approximation while simpler components such as shift registers and adders have been used to replace more area and power consuming IPs such as multipliers. In other works [14], [15], an astrocyte model is proposed that breaks the synchronization of two "Hopf" oscillators. The circuit is realized on FPGAs and the authors report a successful asynchronous operation of the "Hopf" oscillators. SPANNER [22] is a High Level Synthesis (HLS) implementation of the astrocyte model that is used to perform self-repair in astrocyte-neural networks [3]. In addition to digital implementations, analogue realization of astrocytes has also been explored [16]–[19]. Generally, analogue circuits are more accurate because of the absence of quantization errors than digital implementations. However, analogue circuits are more prone to noise and require more memory. Moreover, the existing Electronics Design Automation (EDA) tools cannot infer analogue circuit from HDL codes.

In summary, these approaches have all focused on basic astrocyte behavior such as synchronization of neuron firing rates and not on the mechanism of repair. Models which realize the repair capability are richer in biological detail, i.e. more complex due to increases in the types of interactions between astrocytes, neurons, and synapses.

The novelty of the proposed work in this paper is implementing a SANN, e.g. astrocyte-neuron network with the repair capability, using fixed-point representation. In addition, analysis of the fixed-point SANN's tolerance to different fault ratios is assessed.

III. AN ASTROCYTE MODEL FOR SELF-REPAIR

Research at the Intelligent Systems Research Centre (ISRC) at Ulster University has led to an astrocyte model specific to performing self-repair in SNNs. The self-repair mechanism focuses on the interactions between astrocyte and neurons. The model for astrocytes operates within a feedback loop where the activity of neurons is relayed back to their associated tripartite synapses via astrocytes [3], [5], [20]. When an action potential occurs at the presynaptic neuron, Glutamate is released into the synaptic cleft and binds to the mGlu Receptors (mGluR) on the postsynaptic terminal. Significant depolarization of the postsynaptic neuron results in movement of calcium into its dendrite via NMDA receptors, leading to the synthesis of the feedback messenger endocannabinoid (2-AG) which is subsequently released into the extracellular space. 2-AG is a type of endocannabinoid and acts as a signaling pathway to the presynaptic terminal in two manners, namely directly and indirectly.

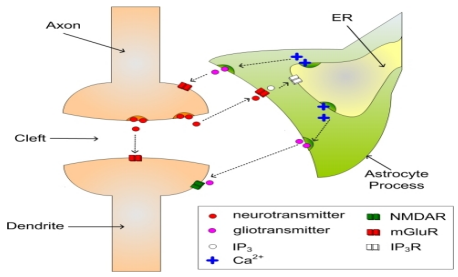


Fig. 1. Tripartite synapses [3]

1- Directly: 2-AG crosses the synaptic cleft to attach to type 1 Cannabinoid Receptors (CB1Rs) on the presynaptic axon. This lowers the probability of release (PR), a process called Depolarization Induced Suppression of Excitation (DSE).

2- Indirectly: 2-AG also binds to CB1R located on the astrocyte process. The upshot of this is IP3 formulation in the astrocyte that subsequently causes the release of intracellular calcium from Endoplasmic Reticulum (ER). Consequently, Glutamate in vesicles are released into the extracellular space and subsequently bind to mGluRs at the presynaptic terminal, enhancing PR. This action is termed e-SP.

This self-repair mechanism is illustrated in Fig. 1. When a synapse fails to release neurotransmitter, the associated neural activity falls and consequently the level of 2-AG decreases. The absence of the 2-AG signal, which is a retrograde feedback messenger from active postsynaptic neurons, causes an overall increase in PR at all tripartite synapses. This is because the direct feedback of 2-AG to the presynaptic terminal DSE has diminished leaving the indirect feedback signal from the astrocyte e-SP to cause a sudden increase in PR. It is important to note that one or more nearby active neurons will be sufficient to maintain the astrocyte in an excited state to produce enough e-SP. Therefore, the repair capability can be viewed as distributed whereby the state of neurons is continually monitored locally by the nearby astrocyte. Also because the astrocyte coverage is up to about 10^5 synapses then local in biological network terms can be anything between 4 to 12 neurons [21].

A probabilistic based tripartite synapse model is used where a random number between 0 and 1 is generated by a uniformly distributed pseudorandom number generator when the

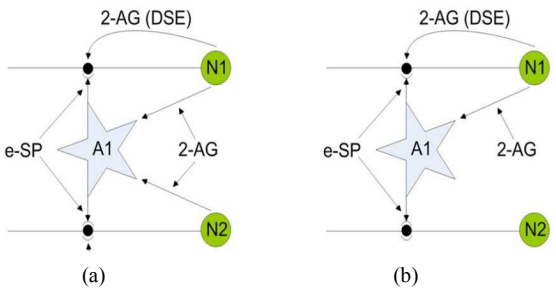


Fig. 2. Interactions between the astrocyte and the two neurons between and after the fault is injected [3].

presynaptic terminal injects a spike into the tripartite synapse I_{syn}. If this value is larger than PR, no current will be injected. Otherwise a fixed current of 5000pA will be inserted into the Leaky Integrate and Fire (LIF) neuron.

$$I^{i}_{syn}(t)=\begin{cases} Iinj & rand \leq PR \\ 0 & rand > PR \end{cases} \quad (1)$$

When the network functions as usual with no faults, the PR relating to each synapse is presented as:

$$PR_{(t)} = \frac{PR_{(t0)}}{100} * DSE_{(t)} + \frac{PR_{(t0)}}{100} * eSP_{(t)} \quad (2)$$

Here, $PR_{(t0)}$ is the initial PR of each corresponding synapse. In this case each synapse activity is depressed by ~50% overall. When a major fault is present at one or multiple synapse that prevents neuron (N2) from firing (Fig. 2), the DSE variable in (2) will no longer be relevant, since it will only be present if the postsynaptic neuron fires, hence the equation can be rewritten as below:

$$PR \rightarrow (\frac{PR_{(t0)}}{100} * eSP_{(t)}) \quad (3)$$

With the absence of DSE, the PR experiences an increase of 200% at the associated healthy synapse site. This is due to the fact that the indirect feedback pathway from N1 through the astrocyte (A1) still exists and acts as a catalyst for self-repair. In short, it means that even if N2 is silent because of one or more faulty synapses, other healthy synapses or the recoverable faulty synapses still can bring N2 back to activity due to the indirect signalling from N1 via the astrocyte pathway.

A. Astrocyte hardware implementation

In this work, the astrocyte block diagram in Fig. 3 has been implemented with a fixed-point representation on a Stratix IV FPGA and verified using Altera SignalTap II. This astrocyte model is biologically faithful and gives accurate results when compared to a MATLAB software version that uses the same equations as the hardware version. The mathematical equations, parameters of the astrocyte, input values and expected output results for both the software and the HDL versions of the astrocyte are obtained from the original self-repair model [3]; e.g. this is a block based design with each block representing a parameter of the astrocyte mentioned in the original model [3].

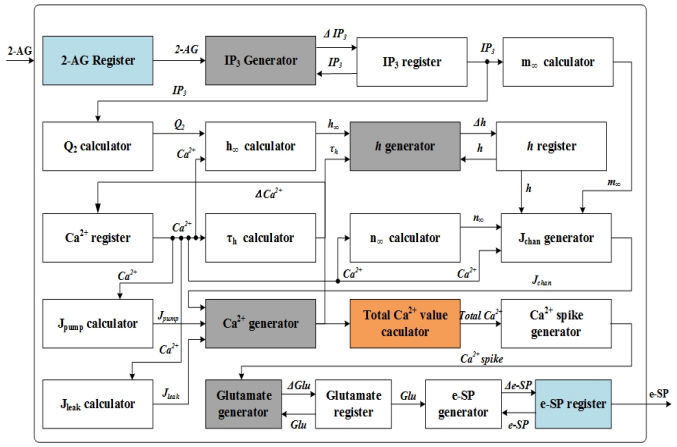
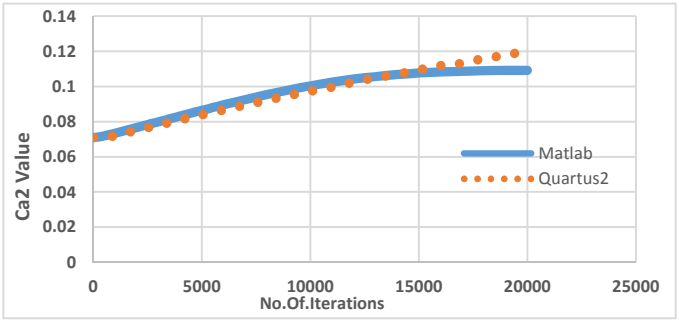


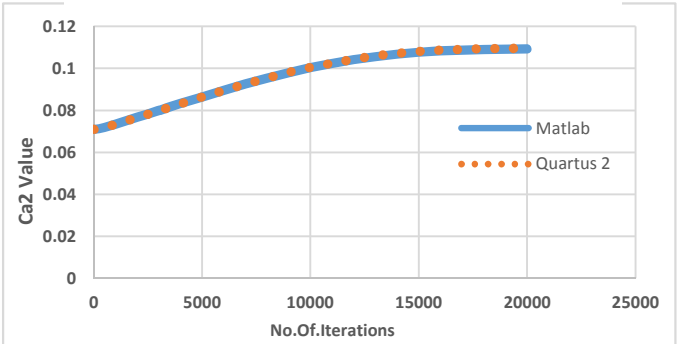
Fig. 3 Astrocyte hardware block diagram

B. Reducing astrocyte bit precision

In software, using a large number of bits to represent the values of the parameters and the variables of the astrocyte model presents little performance issues since modern computers can compute using double floating precision without much effort. In hardware, however, this can prove challenging for the designer because using larger resolutions can significantly increase the design resource consumption and physical blueprint, while using a small resolution can affect the accuracy of the results significantly. For efficient implementation fixed point arithmetic is used. We start from a resolution of 24 bits, 16 bit for the fraction part and the remaining 8 bits for the real part. These values were chosen because the astrocyte values include significant fraction arithmetic elements. The precision of the results is important since inaccurate results from any of the astrocyte sub-blocks in Fig. 3 will propagate through the chain and affect the output result of the complete block. Fig. 4 shows a comparison between results from the floating-point MATLAB and fixed-point VHDL astrocyte models. The *x*-axis is the number of iterations while the *y*-axis represents the calcium value which is obtained from "Ca^{2+} generator" block in Fig. 3. The calcium value has been chosen for accuracy comparison because the blocks that follow are threshold based. Each 1,000 iterations represent one second of biological time, as the Euler method is used for solving the differential equations representing various components of the SANN [3] with a time constant of 1ms. Altera Quartus Prime software is used for synthesizing the VHDL code onto a Startix IV FPGA and Altera SignalTap II is used for verifying the results. As can be seen from Fig. 4(a), for the 24-bit precision, the difference between the two trajectories is non-linear and fluctuates between iterations 10,000 and 15,000 before picking up steadily until it reaches just over 10% at 20,000. We believe that for a



(a) FPGA- Matlab- 24-bit resolution comparison



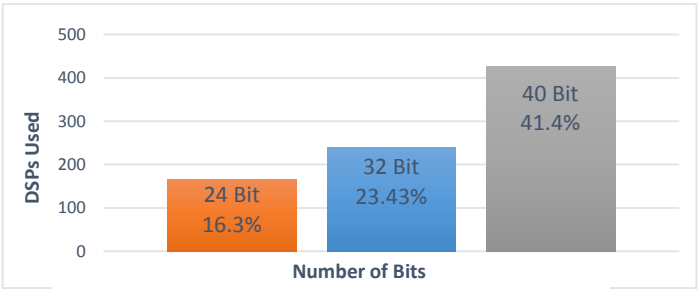
(b) Matlab - 32-bit resolution comparison

Fig. 4. Comparison between the double-float Matlab astrocyte model and its fixed-point VHDL hardware implementation

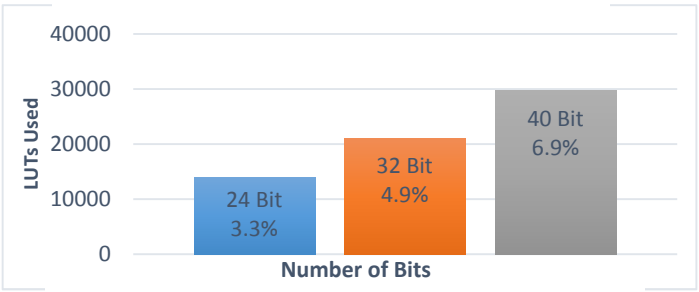
biologically faithful astrocyte, loss in accuracy of 10% is non-trivial. This led to increasing the resolution from 24 bits to 32 bits with all additional 8 bits assigned to the fraction part since that is where the error occurs. Fig. 4(b) illustrates enhancement in accuracy of the hardware model after the resolution was increased as the trajectory of the hardware model is closely aligned with the Matlab software model; this indicates a better fit between the values of the two models. For the 32-bit resolution, the difference between the two plots in Fig. 4(b) is 0.04% after 20,000 iterations. This error is small compared to over 10% for the 24-bit model. On average, for over 20,000 iterations, the percentage difference is 2.7312% for the 24-bit implementation while it is 0.1667% for the 32-bit design. Hardware resource usage for 24-bit, 32-bit and 40-bit resolutions are shown in Fig. 5. A 32-bit resolution was chosen after examining resource utilization and accuracy error; in comparison with the Matlab software model, the error rate of the 24-bit hardware design is unacceptable and the 40-bit resolution doesn't significantly enhance the accuracy over the 32-bit design. In addition, the 40-bit design significantly increases resource usage. A neuron in 32-bit resolution requires 1840 LUTs (< 1% of device resources) and 60 DSP blocks (6% of device resources). It is worth noting that the resoure utilization results are obtained using Altera Quartus tools. It is possible to command the synthesis tool to further optimize the resourse usage or to use less DSP at the cost of using more LUTs. However, we show these figures for the sake of comparison between the designs for different bit resolutions. Balancing out hardware resource utilization for optimisation will be explored in future work.

IV. SANN HARDWARE IMPLEMENTATION AND RESULTS

In this section, a SANN containing one astrocyte and two surrounding neurons with ten synapses each was implemented on an Altera DE4 development board with the Altera EP4SGX530KH40C2 device. The astrocyte (A1) monitors the activity at two group of synaptic junctions, each group containing 10 synapses that are associated with the two neurons (N1 and N2). Fig. 2 shows the topology of this network. The LIF model was used for describing the neurons and the tripartite synapses are probabilistic and based on the model developed in [3]. The inputs to the tripartite synapses are stimulated by means of spike generators. The advantages of the SANN implemented in this work are reproducing biologically faithful behaviors, resilience to high fault ratios and maintaining functionality when the bit precision is reduced. The limitation of this design is the necessity of using multiple FPGAs for a network with a large number of astrocytes, neurons and tripartite synapses.



(a) DSP count



(b) LUT count

Fig. 5. Comparision of resource utilization between the different astrocye implementations.

A. FPGA Self-repair with Reduced Bit Precision

To demonstrate the self-repair characteristics of the SANN in hardware, an experiment was conducted where the network suffers a catastrophic failure, with 80% of synapses stop functioning, but still able to maintain more than 55 % of its functionality. This is the worst-case scenario; the network will retain a higher amount of it is frequency when the fault rate is less significant as is shown in Section IV-b.

Faults were injected into hardware by means of physically reducing the PR value of synapses to zero (classed as "severe" fault), or to a value of 0.1 ("partial" fault). That means the PR values of the synapses were physically changed on the FPGA. This method allowed synapses to be selected as faulty via forcing the PR value to zero or 0.1.

The SANN of Fig.2 was implemented on FPGA with the 32-bit astrocyte model. Simple shift registers were used to act as spike generators. The outputs of the spike generators were fed into the presynaptic connections at the tripartite junctions. Average frequencies that were read from neuron outputs over a window of 1,000 iterations were used to assess the network. Average frequencies were used rather than instant frequencies because the output frequencies in SNNs oscillate over time. The average output frequencies of the network were observed from the two neurons (N1 and N2), as shown in Fig. 2. Initially, the synaptic junctions were all operating normally, then a severe fault was injected simultaneously into 8 out of the 10 synapses connected to N2. The average output frequency of N1 and N2 were then observed on the FPGA. Fig. 6 shows the average output frequency of the two neurons before and after the faults were injected.

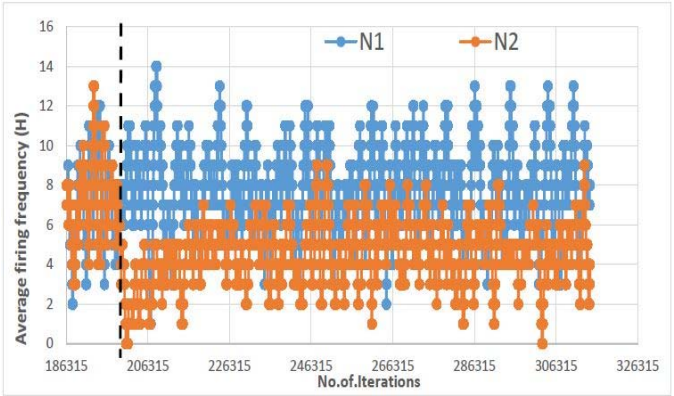


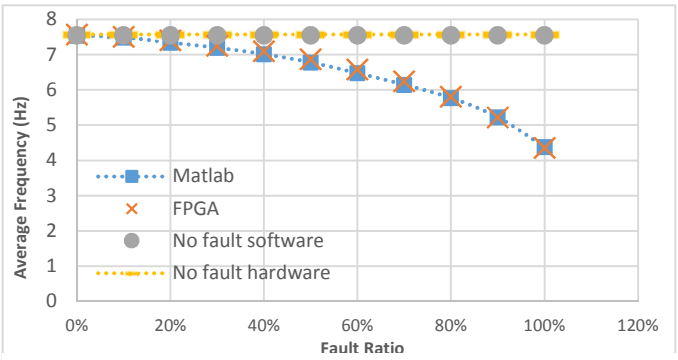
Fig. 6: Average output frequencies of N1 and N2

The values present on the *X*-axis represent the number of iterations. The fault is injected after 200K iterations which is indicated by a dotted black line in Fig. 6. As shown in the graph, the average output frequency of N2 is reduced to zero steeply at first, before increasing again due to the activation of the astrocyte self-repair mechanism. As the number of synapses (8 out 10 for N2, e.g. 80% failure) and the severity of the fault (PR forced to zero) the N2 average frequency does not fully recover to its previous values again. Since no fault is introduced to the inputs of N1, the average frequency graph fluctuates as normal.

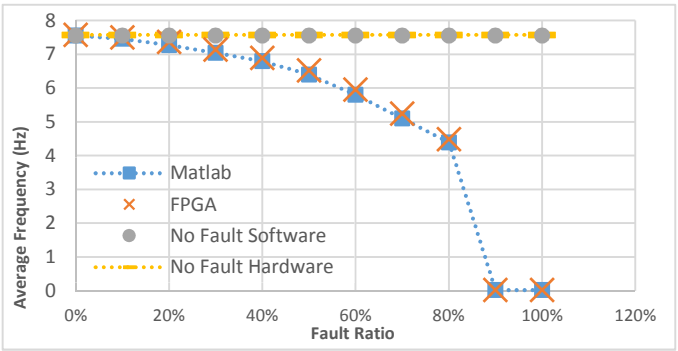
B. Assessing Self-repair over varied fault rates

In this section, similar tests were repeated for different fault rates e.g. 10 % , 20 %…etc. A 10 % fault rate means that one of the 10 synapses is damaged. This is the first study in assessing the graceful degradation of SANNs in hardware. In essence, this enables a threshold to be identified where resilience of the SANN network starts to diminish as it deviates significantly from its original target average output frequency. 'Severe' and 'partial' types of faults will be evaluated.

A severe fault means a synapse's probability of release, $PR_{(t0)}$ in (2), is reduced from an initial value of 0.5 to zero, while partial fault means the probability of release is reduced to 0.1. Faults of different percentages were injected to the SANN running on the FPGA and the average output frequencies were recorded. Fig. 7. illustrates the average output frequency of the faulty neuron (as depicted in neuron N2 of Fig. 2) in response to different fault ratios and compares these values to a situation when no fault exists. Average frequency in this figure's context means the main average of the average output frequencies of the neuron over 130,000 iterations, i.e. the average of the entire simulation duration in Fig. 6. This value arises from the fact that Altera SignalTap II can save results for a maximum of 128*1024 clock cycles. Fig. 7(a) demonstrates the response of the neuron with faulty synapse(s) when the fault reduces the probability of release to 0.1. According to (3), the tripartite synapse is still able to generate $PR_{(t)}$ in the subsequent iterations. Although this value is less when compared to a situation when no fault exists, it still can cause the synaptic junction to pass spikes to its associated neuron when the value of the random probability generator is less.



(a) Average neuron frequency in response to different rates of a partial fault



(b) Average neuron frequency in response to different rates of a severe fault

Fig. 7: Average neuron frequencies in response to different rates of injected faults.

It is evident from Fig. 7 that even with 40% damage the output of N2 can still reach a value of over 7Hz which is close to the target 7.55 Hz (7.57 Hz for the FPGA implementation) and one with which the SANNs paradigm can still function. This degradation is minimized by the self-repair mechanism; e.g. the astrocyte re-strengthens the remaining healthy and partially synapses to encourage the neuron N2 to maintain its firing freq. Note: neural networks are insensitive to slight variation in output frequencies. When all 10 synapses are deemed 'partial' faulty (100%), the neuron degrades more than 55% from its target firing frequency. However, this is a graceful degradation as shown in Fig. 7(a). In Fig. 7(b), severe faults are injected to the synapses and the frequency degradation is more significant since a severe fault reduces $PR_{(t)}$ to zero and the faulty synapse will not pass spikes regardless of the value of the random probability generator. In this case, at 30% damage the output of N2 can still reach a value of over 7Hz which is acceptable to the target. This is clearly evident in Fig. 7(b). In addition, the average frequency projection falls more steeply and N2 becomes inactive when the fault rate is more than 80%. Fig. 7(b) also shows that the self-repair mechanism helps the neuron retain more than 50% of its original average frequency even at 80% fault rate. Fig. 7 also illustrates the Matlab model results for the same network under the same testing condition. From the figure it can be seen that the fixed-point hardware model provides significant accuracy when benchmarked against the double-float Matlab implementation [3]. The key point to note is the capability of the SANN hardware to still maintain operation when over 30% of faults in the synapses are experienced. This provides a high degree of tolerance to failure via self-repair, even when the core astrocyte model's precision is reduced to minimize hardware resources.

V. Future Work and Conclusion

The SANN model presented in this paper will be used as the basis for developing an FPGA accelerator for SANNs for various parameters and dimensions. Furthermore, it will be necessary to develop necessary software and hardware for injecting simulation and configuration data along with monitoring the results. Since the envisioned accelerator platform is expected to span multiple FPGAs, appropriate communication strategies will be required to ensure integral data exchange between the FPGAs. Table. 1 shows the early results of the improved performance achieved as a result of implementing the SANN model on dedicated FPGA. The acceleration factor in Table. 1 shows the advantage the FPGA

implementation presented in this paper has over previous SANN implementations in terms of simulation speedup.
As can be seen, the FPGA accelerator platform implemented in this work is 1,067 times faster than an equivalent software implementation. This is a significant speedup since it implies that SANN simulations times can be greatly reduced using such a FPGA accelerator platform.

Table 1 Speedup results using different astrocyte implementations

Platform implementation	Running time (seconds)	Acceleration factor
Biological neural network	600	N/A
Software simulation	64	1,067
SPANNER	7	116.7
FPGA Accelerator platform	0.06	1

In summary, in this paper, a fixed-point hardware model of an astrocyte has been implemented on FPGA hardware, and results comparing the astrocyte hardware with the software model.
Based on the FPGA resource utilization and the accuracy of the hardware astrocyte model, a resolution of 32-bits was identified and this model was used in a SANN network to evaluate the self-repairing capability in FPGAs. Results obtained directly from the FPGA using Altera SignalTap II demonstrates this self-repair capability together with the accuracy of the fixed-point hardware implementation when benchmarked against the double floating point software model [3]. This provides scope for implementing much larger SANNs on FPGAs and integrating with existing Networks-on-Chip [11].

ACKNOWLEDGEMENTS

The authors would like to acknowledge the EPSRC funding council grant (EP/N00714X/1 & EP/N007050/1) and Ulster University for supporting this research.

REFERENCES

[1] 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, 2013.

[2] S. Carrillo *et al.*, "Advancing interconnect density for spiking neural network hardware implementations using traffic-aware adaptive network-on-chip routers," *Neural Networks*, vol. 33, pp. 42–57, 2012.

[3] 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.," *Front. Comput. Neurosci.*, vol. 6, no. September, p. 76, 2012.

[4] S. Y. Gordleeva, S. V Stasenko, A. V Semyanov, A. E. Dityatev, and V. B. Kazantsev, "Bi-directional astrocytic regulation of neuronal activity within a network.," *Front. Comput. Neurosci.*, vol. 6, no. November, p. 92, 2012.

[5] 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 Networks Learn. Syst.*, vol. 26, no. 10, pp. 2370–2380, 2015.

[6] 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.

[7] 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," *Int. J. Reconfigurable Comput.*, vol. 2009, pp. 1–13, 2009.

[8] A. B. Porto-Pazos *et al.*, "Artificial astrocytes improve neural network performance," *PLoS One*, vol. 6, no. 4, pp. 1–8, 2011.

[9] P. Mesejo, O. Ibáñez, E. Fernández-Blanco, F. Cedrón, A. Pazos, and A. B. Porto-Pazos, "Artificial Neuron–Glia Networks Learning Approach Based on Cooperative Coevolution," *Int. J. Neural Syst.*, vol. 25, no. 4, p. 1550012, 2015.

[10] B. A. Abed, A. Ismail, and N. A. Aziz, "Real time astrocyte in spiking neural network," *SAI Intell. Syst. Conf. 2015*, pp. 565–570, 2015.

[11] 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 Regul. Pap.*, vol. 63, no. 12, pp. 2290–2303, Dec. 2016.

[12] M. Hayati, M. Nouri, S. Haghiri, and D. Abbott, "A Digital Realization of Astrocyte and Neural Glial Interactions," *IEEE Trans. Biomed. Circuits Syst.*, pp. 1–12, 2015.

[13] 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.

[14] S. Nazari, M. Amiri, K. Faez, and E. Karami, "A novel digital circuit for astrocyte-inspired stimulator to desynchronize two coupled oscillators," in *2014 21st Iranian Conference on Biomedical Engineering, ICBME 2014*, 2015, no. Icbme, pp. 80–85.

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

[16] M. Ranjbar and M. Amiri, "On the role of astrocyte analog circuit in neural frequency adaptation," *Neural Comput. Appl.*, pp. 1–13, 2015.

[17] M. Ranjbar and M. Amiri, "An analog astrocyte–neuron interaction circuit for neuromorphic applications," *J. Comput. Electron.*, vol. 14, no. 3, pp. 694–706, 2015.

[18] S. Barzegarjalali and A. C. Parker, "A neuromorphic circuit mimicking biological short-term memory," in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2016, pp. 1401–1404.

[19] Y. Irizarry-Valle and A. C. Parker, "An Astrocyte Neuromorphic Circuit That Influences Neuronal Phase Synchrony," *IEEE Trans. Biomed. Circuits Syst.*, vol. 9, no. 2, pp. 175–187, Apr. 2015.

[20] S. Nadkarni and P. Jung, "Modeling synaptic transmission of the tripartite synapse.," *Phys. Biol.*, vol. 4, no. 1, pp. 1–9, 2007.

[21] 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. 6473–7, Jun. 2007.

[22] J. Liu, J. Harkin, L. P. Maguire, L. J. Mcdaid, and J. J. Wade, "SPANNER : A Self - Repairing Spiking Neural Network Hardware Architecture.", *IEEE International Symposium on Circuits and Systems (ISCAS)*, Montreal, QC, pp. 1350-1353, 2016.