
Proceedings Paper:
Johnson, Anju Pulikkakudi orcid.org/0000-0002-7017-1644, Liu, Junxiu, Millard, Alan Gregory orcid.org/0000-0002-4424-5953 et al. (6 more authors) (2019) Time-multiplexed System-on-Chip using Fault-tolerant Astrocyte-Neuron Networks. In: 2018 IEEE Symposium Series on Computational Intelligence (SSCI). 2018 SYMPOSIUM SERIES ON COMPUTATIONAL INTELLIGENCE, 18-21 Nov 2018 , IND

https://doi.org/10.1109/SSCI.2018.8628710

Time-multiplexed System-on-Chip using Fault-tolerant Astrocyte-Neuron Networks

Anju P. Johnson†, Junxiu Liu*, Alan G. Millard†, Shvan Karim*, Andy M. Tyrrell†,
Jim Harkin*, Jon Timmis†, Liam McDaid* and David M. Halliday†
†Department of Electronic Engineering, University of York, York YO10 5DD, UK
∗School of Computing and Intelligent Systems, Ulster University, Derry BT48 7JL, UK
Email: {anju.johnson, david.halliday, alan.millard, andy.tyrrell, jon.timmis}@york.ac.uk
Email: {j.liu1@, haji_karim-s@email., jg.harkin@, lj.mcdaid@}ulster.ac.uk

***Abstract*—Spike-based brain-inspired systems have shown an immense capability to achieve internal stability, widely referred to as homeostasis. This ability enrols them as the best candidate for next-generation computational neuroscience as they bridge the gap between neuroscience and machine learning. Spiking Neural Networks (SNN), a third generation Artificial Neural Network (ANN), which operates using discrete events of spikes, contributes to a category of biologically-realistic models of neurons to carry out computations. Spiking Astrocyte-Neuron Networks (SANN) have a characteristic attribute homologous to brain self-repair. Although SNNs are more powerful in theory than 2nd generation ANNs, they are not widely in use as their implementations on normal hardware are computationally-intensive. On the contrary, due to the capability of modern hardware such as FPGAs, which operates in MHz and GHz range, facilitates real-time and faster-than-real-time simulations of SNNs. In this work, we overcome the computational overhead of the SNNs using the benefits of real-time hardware computations, utilizing time-multiplexing to design a Self-rePairing spiking Astrocyte Neural NEtwoRk (SPANNER) chip, generic to users' choice of task, emphasizing fault-tolerance, targeting safety-critical applications. We demonstrate the proposed methodology on a SANN system implemented on Xilinx Artix-7 FPGA. The proposed architecture has minimal hardware footprints, power dissipation profile and real-time computational capability, enhancing its usability in constrained applications.**

Keywords*—Spiking Neural Networks, Astrocytes, Self-Repair, Fault Tolerance, Time Multiplexing, FPGA, Bio-inspired Engineering, Neuromorphic Computing.

I. Introduction

Self-rePairing spiking Astrocyte Neural NEtwoRk (SPANNER) is a novel chip- based on Field Programmable Gate Arrays (FPGAs) designed to mimic bio-inspired systems consisting of Spiking Astrocyte Neural Network (SANN) targeting real-time safety-critical applications. The system works on accelerated biological time scales and uses the principles of time-multiplexing to achieve a compact architecture which is highly scalable to support real-world tasks. This work collaborates with the researches in Neuroscience with Electronics in designing a brain-inspired spiking neural network node generic to the users choice of application. The work acknowledges the contribution of a special cell in brain termed the Astrocyte in regulating neurotransmitters [1] and unsupervised biological learning methods [2], [3] in establishing fault-tolerance in the system and hence is a brain-inspired work.

The human brain includes two major cells- Neurons and Glia. Glia can be of different types including Astrocytes, Oligodendrocytes, and Microglia. These cells work together to create electrical activity in the neuron. It is this electrical activity that underpins the ability of human intelligence. Brain cells use spikes for communication. A time-series of electrical signals recorded from individual neurons in the brain is termed a spike-train. Spike trains are action potentials that occurs when the membrane potential of a neuron exceeds a threshold value and rapidly falls. Hence the human nervous system is a spike-driven system. Spiking Neural Networks (SNN) mimic spiking mechanisms of neurons as they perform computations using biologically-realistic models. SNNs aims to reduce the disparity between neuroscience and Machine Learning (ML) and are promising as the next generation ML [4], [5].

The SPANNER architecture is built around the question: Can the damaged brain repair itself? The human brain is remarkable in its ability to self-repair, for example following injury or stroke. The human brain is capable of self-repair resulting from a range of distributed and fine-grained mechanisms which act in synchrony to ensure that the neurons in the system continue to function in as close to a normal state as possible. This opens up two concepts in neuroscience. One deals with brain plasticity, whereas the second investigates how new brain cells replace damaged ones (Neurogenesis and Synaptogenesis). Brain plasticity, also called neuro-plasticity is the brain's ability to change its own structure and function for adjusting to new situations within the body or in the external environment [6], [7]. This work emphasizes the concept of brain-plasticity using two concepts. Firstly, we utilize the astrocytes capacity in regulating neurotransmitters leading to selective transmission of neural information in the SNN. Secondly, we include unsupervised biological learning methods in the architecture to produce reliable neural activity irrespective of failures in the system.

Previous papers [8], [9] discuss how Astrocytes and brain-plasticity can co-exist to produce a useful application. One problem faced with the above implementation is scalability. In this paper, we work on a universal architecture which is scalable and implementable in reduce hardware resources achieving real-time computations. The system is run on Field Programmable Gate Arrays (FPGAs) with a clock speed of $100MHz$, establishing homeostasis in milliseconds. In this work, we define homeostasis as the ability of the neurons in the system in achieving a constant firing activity irrespective of faults in the system. The faults we consider include

irregular spiking activity in presynaptic neuron or failures in the interconnection between the pre and postsynaptic neurons (synapses). Although the potential of SNNs is widely recognized, their capabilities are not fully exploited due to their computationally-intensive calculations of differential equations. Conversely, modern hardware such as FPGAs, which operates at MHz and GHz range helps in achieving this goal. Recent research [10], [11] showed that FPGAs can outperform GPUs in complex tasks using Deep layer Neural Network (DNN), AI, big data or ML. Also, FPGAs provide superior energy efficiency compared to GPUs. Compared to Application Specific Integrated Circuits (ASICs), FPGAs are flexible and make it easy to implement changes on the same chip. In this work, we overcome the computational overhead of the SNNs using the benefits of real-time hardware computations on FPGAs, utilizing time-multiplexing to design a SPANNER chip.

Time Multiplexing is a method of reusing a common signal pathway so that each activity uses the path only for a fraction of time [12]. In this model, we extract common operations in a SANN system and share these resource using multiple signals arriving through this path at specified time intervals. Sparse nature of neuronal spiking behavior helps to utilize time multiplexing without much loss of spiking information (dense nodes may loss spikes during time-slots of other nodes). The nature of FPGA in achieving high calculation speed enables real-time interactions with the environment to be easily latched into the SANN system without loss. The SPANNER chip is generic to users' choice of task, emphasizing fault-tolerance, targeting safety-critical applications and is a promise to next-generation cognitive or neuromorphic computing applications.

The rest of the paper is organized as follows. Section II describes the contribution of astrocytes in regulating neural transmitters, emphasizing its role in selectively propagating spiking information down the multi-layer neural network. Section III introduces the role of learning mechanisms in a SANN system in achieving a constant firing activity in a system. In Section IV, the design of novel SPANNER Universal Function Generator derived by using time-multiplexing and solutions for implementing complex biological processes using approximations for a hardware efficient scalable design is discussed. This section incorporates the main contribution of the work. Experimental results establishing the effectiveness of the proposed scheme by analyzing the learning/repair abilities are presented in Section V. Finally, conclusions and future work are discussed in Section VI.

II. Astrocyte-GABA Interactions

Synapses in the human brain are surrounded by astrocytes, with each astrocyte encloses over 100000 synapses [13]. This introduces a notion of a tripartite synapse [14], which consists of presynaptic terminal, a postsynaptic terminal, and an astrocyte terminal. The role of astrocyte as an active element in the brain facilitating neurotransmitter regulation, brain homeostasis, and normal cognitive function has been acknowledged [15]. Recent evidence shows that interneurons play a vital role as key regulators in neural network functioning and can activate Ca^{2+} signaling in astrocytes [16]. There are two classes of interneurons: excitatory-utilizes the neurotransmitter glutamine, and inhibitory-uses gamma-aminobutyric acid

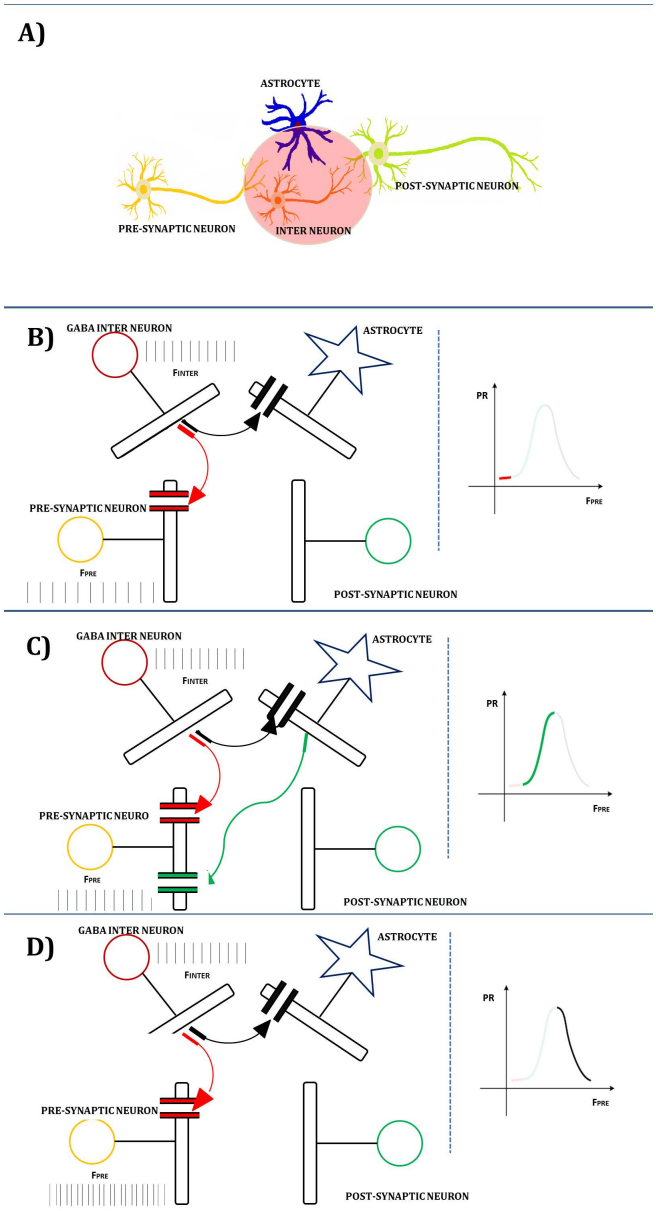

Fig. 1: **Regulatory phenomena mediated by GABAergic signaling**: (A) Tripartite synapse with a paired interneuron (B) represents the signaling pathways between a GABA interneuron and the tripartite synapse at a low presynaptic frequency (C) as presynaptic frequency increases (D) and at the high presynaptic frequency. The conventional tripartite synapse consists of a presynaptic terminal, a postsynaptic terminal, and an astrocyte terminal. Transmission probability of a tripartite synapse coupled to a GABA inter-neuron follows a Gaussian distribution. Transmission probability reduces under low or high presynaptic input.

(GABA) and/or glycine as their neurotransmitter. The balance between the firing of excitatory and inhibitory interneurons is critical in the maintenance of normal sensory function. In this work, we now consider the signaling pathways between a tripartite synapse and a GABA interneuron terminal as shown in Fig. 1-A.

Let assume that the presynaptic spiking frequency (f_{pre}) is same as that of the GABA neuron frequency [16]. At low presynaptic firing activity GABA binds to GABA-A receptors on the presynaptic terminal and to GABA-B receptors on the astrocyte. Low input spike rate produces low 2-Arachidonoylglycerol (2-AG), a type of endocannabinoid and retrograde messenger. This leads to degradation of inositol 1, 4, 5- trisphosphate (IP3) within the astrocyte as per equation (1).

$$\frac{d(IP_3)}{dt} = \frac{IP_3^* - IP_3}{\tau_{IP_3}} + \gamma_{IP_3} AG \tag{1}$$

Where IP_3 is the quantity within the astrocyte's cytoplasm, IP_3^* is the baseline of IP_3 when the cell is in a steady state and receiving no input, τ_{IP_3} and γ_{IP_3} is the decay and production rate of IP_3 respectively. Because of an insufficient amount of IP_3 is released to open calcium stores in the astrocyte at low f_{pre} and therefore there is no gliotransmitter release from the astrocyte. Also, because GABA is biding to GABA-A receptors at the presynaptic terminal under low f_{pre}, this phenomenon dominates and causes an inhibitory effect represented a low transmission probability (PR) value. This is depicted in Fig. 1-B.

If f_{pre} is increased further, a point is reached whereby a sufficient amount of IP_3 exist within the astrocyte cytoplasm to trigger the opening of the calcium store, leading to the production of calcium spikes and the opening of group I metabotropic Glutamate Receptors (mGluR). This signaling results in an increase of synaptic transmission probability through indirect signaling, which overrides the inhibitory effect as shown in Fig. 1-C.

However, if f_{pre} continuous to increase, the $IP3$ within the astrocyte increase and achieves saturation, whereby the calcium oscillations stop leading to the ceasing of gliotransmitter released by the astrocyte and the PR falls rapidly. This process is depicted in Fig. 1-D.

Based on the above three observation, we represent transmission probability using a Gaussian function as presented in equation (2). This can be viewed as the capability of the system to selectively pass information from one neuron to another in a multi-layered network [1].

$$PR = exp\left(\frac{(f_{pre} - f_s)^2}{2\sigma^2}\right) \tag{2}$$

Where f_{pre} is the frequency of presynaptic neuron spikes, f_s is the center frequency and σ is the width of the Gaussian passband.

III. Learning in SANN

Figure 2 shows a two-layer network with 3 presynaptic neurons for the output neuron N_O. Each pre-post neural coupling has multiple synaptic pathways (8 in our experiments) of different path delays. The design of multiple pathways permits fault-tolerance and is a mechanism to build postsynaptic potential. The GABA-astrocyte interaction regulates transmission probability of each input to neuron N_O as illustrated in Fig. 2.

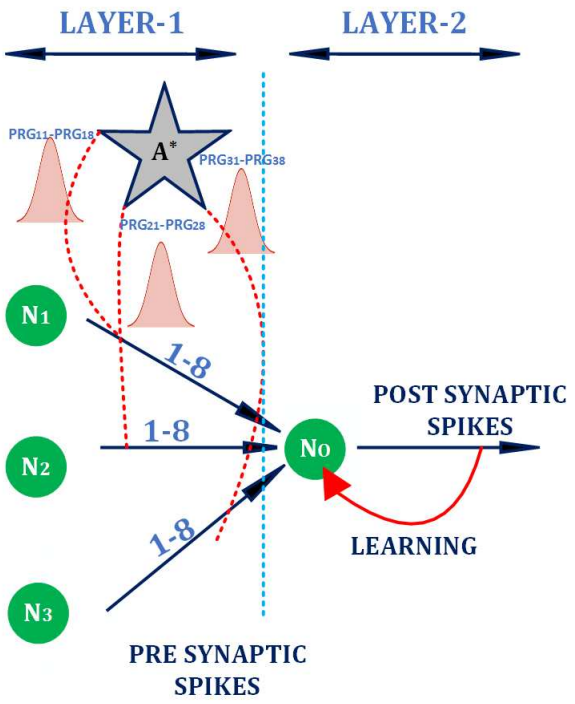

Fig. 2: **Basic unit for fault-tolerant learning mediated by an astrocyte.** Neurons N_1, N_2 and N_3 resides in layer 1 and N_O resides in layer 2 of the SANN. Astrocyte A^* regulates the transmission probability of inputs received by N_O from N_1, N_2 and N_3. There are 8 parallel paths between each of presynaptic and the postsynaptic neuron. A^* modulates PR to pass the selected pattern to the neuron N_O. Based on the input pattern, the output neuron N_O learns to achieve the required spike rate.

Let us assume that the neuron N_O has three presynaptic neurons N_1, N_2, N_3. Each input is encoded by a binary representation of spike train. For example $1 = 35Hz$ and $0 = 25Hz$. We would like to detect a pattern of spike train of frequency $35HZ$, $25Hz$ and $35HZ$ (equivalent to pattern 101)respectively from the presynaptic neurons. GABA-astrocyte coupling controls transmission probability of each connection between a pair of pre-post synaptic neuron (total $3 * 8$ interconnection). A minimum transmission probability PR_0 is required to permit a spike to reach the input of neuron. Let us consider the connection between N_1 and N_O. The transmission probability of each of the eight paths are calculated as per equation (2), with $f_s = 35Hz$, and if any of these paths has a transmission probability of PR_0 or greater, the spikes arriving through N_1 has the required spike rate. This calculation is repeated for other branches of neurons connected to N_O for their respective center frequency. If all the branches pass the test, the pattern is detected and the current equivalent to the spike inputs are fed to the neuron N_O. From the received inputs, the neuron N_O, and it learns to achieve a constant firing activity of $35MHz$.

To generate a constant firing activity for the SANN, we use a learning algorithm. In this approach, STDP [2], [3], together with BCM learning rule [17], [18] are combined to develop

the BCM-STDP rule. STDP uses the time difference between presynaptic and postsynaptic spikes to adjust the synaptic weights as is described by, Equation (3)

$$\delta w(\Delta t) = \begin{cases} +A_0.exp(\frac{\Delta t}{\tau_+}), & \Delta t \leq 0 \\ -A_0.exp(-\frac{\Delta t}{\tau_-}), & \Delta t > 0 \end{cases} \quad (3)$$

Where $\delta w(\Delta t)$ is the weight update, Δt is the time difference between presynaptic and postsynaptic spike events, A_0 is the height of STDP learning window controlling the maximum levels of weight potentiation and depression, τ_+ and τ_- control the decay rate of weight updating. In addition, the BCM learning rule modulates the height of the STDP plasticity window as a function of the neuron actual firing rate according to Equation (4).

$$A_0 = \frac{A}{1 + exp^{a(f-f_0)}} - A_0 \quad (4)$$

Where f and f_0 are the actual and target firing rate of the postsynaptic neuron, respectively, A is the maximum height of plasticity window and $A-$ is the maximum height of plasticity window for depression. The parameter a is constant which controls the opening/closing speed of plasticity window and is found experimentally to be 0.1.

IV. Design of SPANNER Universal system-on Chip

The universal SPANNER system-on Chip, built on the hardware consist of 3 layers - the input layer, the hidden layer, and the output layer. The astrocyte-GABA interaction permits flow of information between the neurons of two adjacent layers as descibed in Section II. Based on the inputs received, the neurons in each layer learns to achieve a constant spiking activity as described in Section III. The term 'universal' is used here to emphasize the fact that any 'n' input boolean function expressed in Sum of Product (SOP) form is implementable using an 'n' input SPANNER chip. The details of how to build a function using SPANNER architecture is described below.

A. Complete Architecture

An 'n' input SPANNER architecture consists of three layers of neurons as shown in Fig. 3. Neurons in the input layer reflect the number of inputs to the system. The neurons in the hidden layer perform the boolean operation of products of input signals. Considering 'n' inputs we have 2^n combinations and hence 2^n hidden layer neurons. For each combination of input, the corresponding spike trains need to be permitted to flow to the corresponding hidden layer neuron. Hence for each hidden layer patterns there exist an Astrocyte. This implies that a hidden layer consists of 2^n Astrocytes in total. Output layer of the network consists of 'k' neurons, each represents the number of functions to be generated. Output layer consists of 'k' number of astrocyte to connect the spike trains of hidden layer neurons with its respective functions.

We now describe how to implement boolean functions using SPANNER architecture. Let us consider a system with 3 inputs S_1, S_2 and S_3. This example corresponds to an obstacle detection network, where S_1, S_2, S_3 represents the sensor

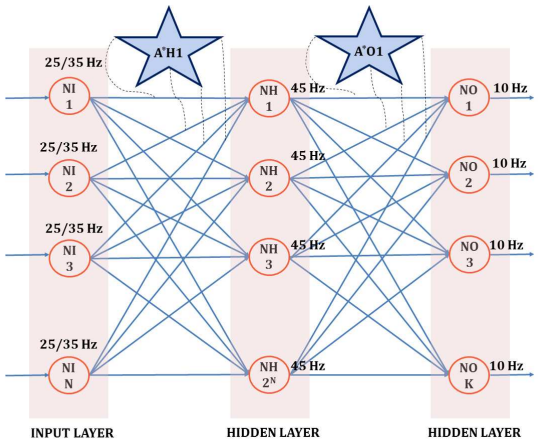

Fig. 3: **Complete Architecture for fault-tolerant learning SNN mediated by an astrocyte.** Neurons $NI_i, (0 < i \leq N)$ resides in input layer, neurons $NH_j, (0 < j \leq 2^N)$ resides in hidden layer and $NO_m (0 < m \leq K)$ resides in output layer of the SANN. There are 2^N astrocyte regulating transmission between input layer and hidden layer. A^*Hj is interpreted as astrocyte j of hidden layer. There are K astrocyte regulating transmission between hidden layer and output layer. A^*Om is interpreted as astrocyte m of hidden layer. Astrocytes modulates transmission probability between the layers to permit the selected pattern to the next layer. Based on the input pattern, the post-synaptic neurons learns to achieve the required spike rate.

distinguishing presence (1) or absence of an obstacle (0) to a robot from directions forward, right and left respectively. The function F corresponds to a decision to move forward, R corresponds to move to the right and L corresponds the decision to move left. The functions to be generated are $F = \overline{S_1}$, $R = S_1\overline{S_2}S_3 + S_1S_2S_3$ are $L = S_1S_2\overline{S_3}$. Let us encode the system with the spike frequency of $25Hz$ to represent a logic 0 and $35Hz$ to represent a logic 1. Since there are three inputs to the system, we require 8 (2^3) hidden layer neurons and the same number of astrocytes in the network hidden layer. Now consider a pair of neurons NI_1 of the input layer and NH_1 of the hidden layer. There are 8 parallel paths (synapses) between this pair of neurons as discussed in Section III. Now consider all input neurons connected to NH_1. The astrocyte-GABA interactions between these inputs neurons and hidden layer neuron NH_1, (A^*H_1) monitors all spikes arriving at NH_1 and selectively enables spike flow only if the pattern is matched. Since the circuit to be configured is in SOP form, the pattern to be detected in NH_1 is '000' ($25Hz$, $25Hz$ and $25Hz$: No obstacle in any direction). Spike propagation to NH_1 is permitted is any of the 8 interconnections between each pair of pre-post synaptic neuron has a transmission probability higher than PR_0 (a predetermined threshold). Similarly, every other hidden layer neuron detects respective patterns (presence/absence of obstacle in various directions) based on its corresponding astrocyte-GABA interaction.

Each hidden layer neuron receives different density of spikes through 8 different synapses. A completely healthy

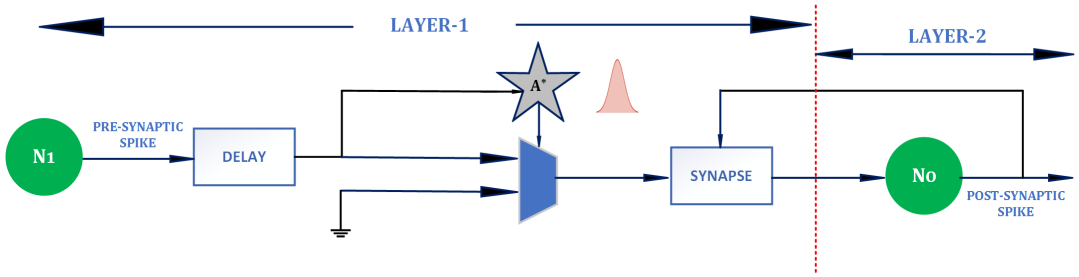

Fig. 4: **Basic unit for fault-tolerant learning mediated by an astrocyte.** Neurons N_1, corresponds to a pre-synaptic neuron and N_O represents a post-synaptic neuron in the SANN system. Astrocyte A^* regulates the transmission probability of inputs received by N_O from N_1. The basic unit contains only a single path between the presynaptic and postsynaptic neuron. A^* modulates PR so as to permit the selected pattern to the neuron N_O. If the pattern is matched, the spikes are permitted to flow to the next execution unit, otherwise spike flow is blocked using a multiplexer. Based on the input pattern, the output neuron N_O learns to achieve the required spike rate.

interconnection means all 8 inputs between a pair of neurons are non-faulty. This may not be the case in practice. Due to failures in the system (Noise from sensors or interconnection failures in hardware implementations), even if patterns are correct, the hidden layer neuron may receive different spike rates. The next stage of learning enables the hidden layer neurons to achieve a constant activity irrespective of variations in the presynaptic inputs. We assume that at least one healthy interconnection exists between a pair of neurons. When a pattern is matched, (at least one interconnection exists between input layer neurons (NI_1, NI_2, NI_3) and NO_1). Based on the received spikes, the synaptic interconnections of neuron NH_1 is updated using Equations (3) and (4) to achieve a constant activity of $45Hz$ (predetermined choice of frequency). when all the hidden layer patterns have been recognized, the Astrocyte in the output layer connects the pattern to the respective function. Here the purpose of the Astrocyte-GABA interconnection is to selectively permit hidden layer signals to the output layer neuron. The Astrocyte corresponding to function F (A^*O_1) permits spike flow to NO_1 if frequency of hidden layer neuron NH_1, NH_2, NH_3 and NH_4 are all $45Hz$. Similarly (A^*O_2) permits spike flow if NH_5, NH_6 and NH_8 are $45Hz$ and (A^*O_3) permits spike flow if NH_7 is $45Hz$. If the patterns are matched, the output layer neurons fire to the predetermined frequency of $10Hz$ using learning mechanism. The final decision of the obstacle detection system can be given priorities using inhibitory interneurons [19] to take final decision of movement of the robot ($Forward > Right > Left$).

B. TDM based Reduced Architecture on Hardware

In this paper, we use a Leaky Integrate-and-Fire (LIF) neuron model [20], due to its simplicity as they require low computing resources and minimal tuning parameters. The neurons are arranged in layers, where a basic spanner unit consist of two neurons and a single interconnection between them as shown in Fig. 4. This unit is then time multiplexed to generate the complete architecture in Fig. 5. The representation of a LIF neuron is described in Equation (5).

$$\tau_{mem}\frac{dv}{dt} = -v(t) + R_{mem}.I_{total} \tag{5}$$

Where τ_{mem}, R_{mem}, v and I_{total} are the time constant, membrane resistance, membrane potential and current injected to the neuron respectively. On reaching the threshold voltage, the membrane potential is brought back and held at 0V following a nominal refractory period (3 clock cycles). The expression is evaluated using Euler method of integration with a fixed time step. Now we explain how, the complete architecture is brought to a reduced form for hardware implementation using the concepts of time multiplexing and optimization. The current generated by 'i_{th}' synapse of presynaptic neuron 'j' (S_{ji}) is given by equation (6).

$$I_{inj_{ji}} = \eta.(W_{ji}) \tag{6}$$

Where (W_{ji} is the synaptic weight of S_{ji}, η is a constant used to modulate the synaptic weights. Based on the value of PR (determined by the astrocyte), the flow of current to the neuron is regulated as described in equation (7).

$$I_{ji}(t) = \begin{cases} I_{inj_{ji}}, rand \leq PR \\ 0, \quad otherwise \end{cases} \tag{7}$$

Where $I_{inj_i}(t)$ is the amount of current generated at time t by the synapse S_{ji}. $'rand'$ is a random function used to model the probabilistic synapse. $I_{j,i}(t)$ is a current released by the synapse on a successful probabilistic event described as in equation (7). The total current injected to a post-synaptic neuron (if a pattern is detected by GABA-Astrocyte interaction) is given by equation (8).

$$I_{total} = \sum_{j=1}^{p}\sum_{i=1}^{q} I_{inj_{j,i}}(t) \tag{8}$$

Where 'p' is the number of presynaptic neurons (also the width of a pattern), 'q' is the number of paths between a pair of pre-post synaptic neuron (8 in our experiments). Based on the input pattern, post-synaptic neuron learns to achieve the required spike rate. Learning is achieved using STDP and BCM rules, using equations (4) (3). If output frequency deviates from the required output frequency (f_o), the weights of synapses are updated by a certain amount. For hardware implementation, equation (4) is approximated using a straight line and equation (3) by powers of 2 (shift operations).

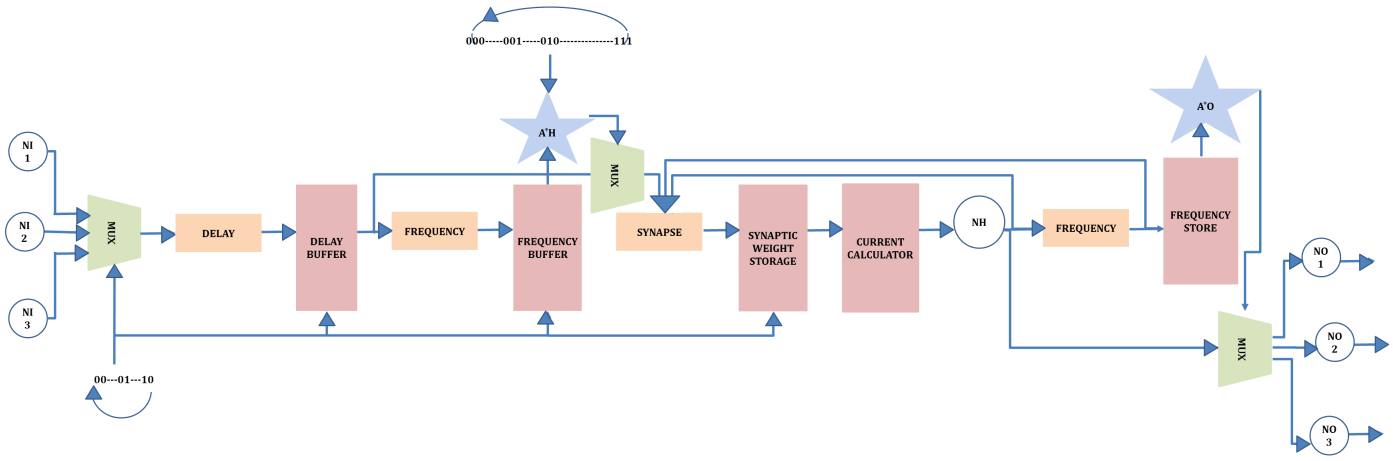

Fig. 5: **Time Multiplexed SPANNER Architecture.** Neurons NI_1, NI_2 and NI_3 resides in the input layer, the inputs are time multiplexed to the system, thus reflecting a presence of a single input neuron. and N_H resides in hidden layer of the SANN. This neuron is time multiplexed to perform the functionality of 8 hidden layer neurons in the complete architecture. Astrocyte A^* regulates the transmission probability of inputs received by N_H from the input layer. The pattern to be checked by the astrocyte changes periodically to incorporate detection of all patterns. The system uses memory elements to store responses in various time-multiplexed stages.

Transmission probability represented by a Gaussian filter in equation (2) is approximated to a rectangular bandpass filter.

We observe that for every pair of neurons there exist 8 parallel paths for the spike propagation. The operations performed in this passage include delay by a predetermined time, astrocyte determining if this route has the specified pattern frequency and updating the synaptic weight of this interconnection using STDP rule following the relation between the postsynaptic neuron and presynaptic neuron spike times. For every 8 interconnection, the same operation is performed, similarly, for every pair of pre-post synaptic neurons, the above-mentioned operation is repeated. Also, for every pattern identification (postsynaptic neuron), the same operation is performed. Hence we use the above procedures as a basic unit to be multiplexed for detecting a particular pattern from the inputs, this is depicted in Fig. 4.

The spanner architecture implemented in hardware consists of 3 input layer neuron, 8 hidden layer neuron and 3 output layer neuron. For implementing any 3 input function, the basic spanner unit is multiplexed three times. The values corresponding to each stage of multiplexing are stored in memory, particularly we store values of spike frequencies and synaptic weights. This is reused for further stages. The complete time multiplexed spanner architecture is shown in Fig. 5. In every three clock cycles, every input neuron (presynaptic neuron) is given a chance to propagate its spike through the basic unit. But for each of its turn, they are propagated with a different predetermined delay, to mimic the 8 parallel delayed paths in the complete architecture. Hence in a total of 24 (3*8), clock cycles are required to process all the input neurons corresponding to a hidden layer neuron. Then the next hidden layer neuron is processed, which permits a different pattern. This process is repeated for every hidden layer neuron. Hence in total, the unit requires 192 clock cycles $(3 * 8 * 8)$ to propagate evaluate spikes of all combinations of input-hidden layer neuron. The system requires 1920000 clock cycles to achieve a constant firing activity. The device operates at a clock speed of $10MHz$, which implies a $200ms$ to make a decision (generate a constant activity), which can be considered as real-time or faster than real-time for real-world applications. We target an application of mapping proximity sensors to wheels of a robot, similar to [8]. For safety-critical missions, the proposed scheme provides fault-tolerance and high scalability compared to [8].

Note that the sparse nature of spike trains helps to utilize time multiplexing without much loss of spiking information. Also, the nature of FPGA in achieving high calculation speed enables real-time interactions from the environment to be easily latched into the SANN system with minimal loss of spiking information thereby achieving a faster control response which is real-time and responsive.

V. Experimental Results

A. Functional equivalence between the two model

The proposed architecture of enhanced learning SANN is implemented in Xilinx Vivado 2018.1. The system is benchmarked against a fault-free SANN architecture implemented on the FPGA. We deliberately induced faults in the system to establish the concept of fault recovery. In presence of faults of various grades, the proposed system could successfully establish fault-free behavior using the proposed neuronal learning concept. The results are in agreement showing that the FPGA based approximation produces results comparable with the complete architecture in hardware and Matlab based software implementations.

B. Real-time Computations

The functions discussed in Section IV is designed using the complete architecture and reduced time-multiplexed architecture. The plot in Fig. 6 shows the response for the obstacle in the right direction. As expected both the complete (blue) and

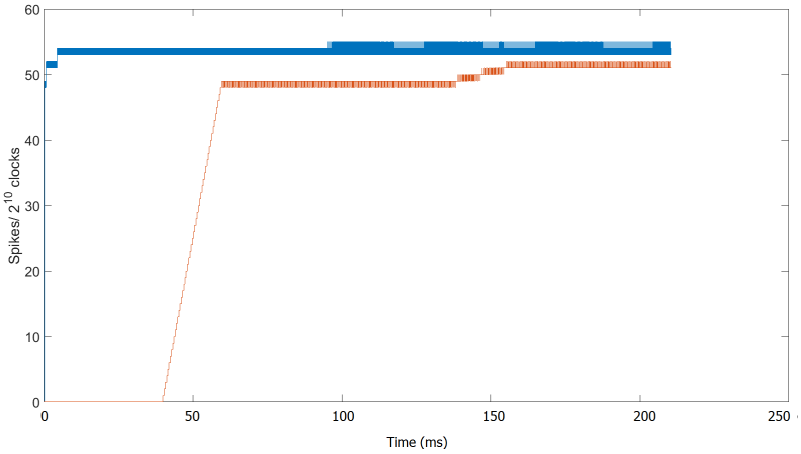

Fig. 6: **Real time computation capabilities of SPANNER Architecture**. Similar test conditions are applied to the complete architecture and the time-multiplexed spanner. The response is collected for the function R described in section IV. Both the complete architecture (blue) and the time-multiplexed architecture (red) produces real time response.

TABLE I: **Hardware overhead of components of SPANNER Architecture**

Components	Slice	Slice Reg	LUT	BRAM	DSP
LIF neurons	7	38	78	0	7
A_0 Generator	140	32	470	0	16
Moving Average calculator	11	21	21	2	0
PR	5	1	21	0	0
synapses	12	425	1138	0	16

the reduced (red) architecture produces real-time responses. The complete architecture achieves constant activity at 1ms, whereas the time-multiplexed architecture requires $200ms$ to achieve the same response. The reduced architecture takes more time to achieve a constant activity, which is due to the trade-off for hardware resources.

C. Hardware and Power Footprints

The proposed SPANNER is implemented on the Xilinx Artix-7 FPGA board. Recovery of firing rates in the proposed methodology, implemented on the FPGA is monitored using the *Integrated Logic Analyzer* (ILA). Power estimation of the circuits was carried out using *Xilinx Power Estimation and Analysis Tools* and delay estimation using *Timing Closure & Design Analysis*. Table II reports the hardware resources required for implementing a single unit of various components. Table I depicts the footprint of the complete architecture and time-multiplexed SPANNER architecture. There is a notable reduction in hardware footprints for the reduced implementation. The proposed reduced architecture is scalable to any number of inputs, the trade-off would be the real-time response rate of the system. The complete architecture was synthesized using Xilinx design tools but was large for realization in Xilinx Artix-7 or larger FPGAs such as Virtex-7. We have previously incorporated the concept of complete spanner architecture in a reduced form using a hybrid network (combination of spiking network and digital circuits) [8]. On the contrary, this work represents a complete spike-driven system (not a hybrid system). Estimated total on-chip power dissipation and the maximum operating frequency of the overall proposed architectures are $1.2W$ and $60MHz$ respectively. As evident from these reports, the proposed fault-tolerant learning mechanism in a SANN can be incorporated with reduced hardware overhead and power consumption, establishing its usability in constraint applications. The proposed system implemented on an FPGA achieves an acceleration of 10^4 compared to the software simulation, which guarantees the viability of this approach for real-time fault-tolerant implementations. For a complete architecture, hardware utilization increases with the number of inputs. The proposed TDM based SPANNER architecture is generic and universal for any number of inputs. The increase in hardware in proposed architecture is only in the memory requirement. We also suggest the use of cascaded SPANNER reduced architecture for building larger circuits.

VI. Conclusions

In this paper, we discussed how interactions between Astrocytes and GABA neuron can be utilized in implementing a fault tolerant system incorporating spike-based learning mechanisms. The SPANNER architecture combines the ability of GABA-astrocyte interactions in selectively transmitting pieces of information down a multi-layer network. The concept is implementable with reduced hardware resource, power dissipation and propagation delay leading to a SANN system scalable to a deep-layer neural network architecture. The SANN system is adaptive to the changes in the system and does not require any dedicated units for fault detection and correction (a fault-tolerant system). Even if some of the inputs are faulty, the system could regenerate the task by adjusting the synaptic weights of the fault-free interconnects. Secondly, the proposed idea is demonstrated on an FPGA system with a real-time computation (10^4 times faster than biological timescale($1ms$)). One reason is its ability to work in an accelerated biological timescale (clock speed of $10MHz$). The system could effectively establish a constant activity with a minimum of 1 interconnection (healthy synapse) between a pair of presynaptic and a postsynaptic neuron. Finally, the

TABLE II: **Hardware overhead of 3-input complete SPANNER Architecture and Time-multiplexed SPANNER Architecture (output layer neurons are not considered)**

Methodology	Complete Architecture						Reduced Architecture					
Components	**Number of Units**	**Slice**	**Slice Reg**	**LUT**	**BRAM**	**DSP**	**Number of Units**	**Slice**	**Slice Reg**	**LUT**	**BRAM**	**DSP**
LIF neurons	77	28	418	858	0	77	2	14	76	156	0	14
A_0 **Generator**	8	1120	256	3760	0	128	1	140	32	470	0	16
Moving Average calculator	32	352	168	168	16	0	2	22	42	42	4	0
PR	32	160	32	672	0	0	1	5	1	21	0	0
synapses	32	384	13628	36416	0	128	1	12	425	1138	0	16

reduced architecture is compared with a complete architecture in terms of area power and computational capacities. Future work will deal with the development of SPANNER architecture to effectively distinguish faults from sensor noise. The work is also open to applications targeting fault-tolerance, where safety is an important constraint. The work address fault-tolerance in electronic devices, and sensor failures in capturing environmental data, by combining bio-inspired fault-tolerance principles. The project aims at developing a bio-inspired SPANNER chip, for in order to achieve fault-resilient functional circuit designs. We particularly target applications of the proposed hardware SANN systems in safety-critical robotic missions for implementing a real-time responsive system establishing the fault-resilience.

VII. Acknowledgements

The work is part of the SPANNER project and is funded by EPSRC grant–(EP/N007050/1, EP/N00714X/1). Additionally, the authors would like to acknowledge the platform grant–(EP/K040820/1) funded by EPSRC.

References

[1] J. Liu, L. McDaid, J. Harkin, J. Wade, S. Karim, A. P. Johnson, A. G. Millard, D. M. Halliday, A. M. Tyrrell, and J. Timmis, "Self-Repairing Learning Rule for Spiking Astrocyte-Neuron Networks," in *Proceedings of the 9th International Conference on Neural Information Processing (ICONIP)*, 2017.

[2] L. F. Abbott and S. B. Nelson, "Synaptic plasticity: taming the beast," *Nature neuroscience*, vol. 3, pp. 1178–1183, 2000.

[3] S. Song, K. D. Miller, and L. F. Abbott, "Competitive Hebbian Learning through Spike-Timing-Dependent Synaptic Plasticity," *Nature neuroscience*, vol. 3, no. 9, pp. 919–926, 2000.

[4] J. H. Lee, T. Delbruck, and M. Pfeiffer, "Training deep spiking neural networks using backpropagation," *Frontiers in neuroscience*, vol. 10, p. 508, 2016.

[5] W. Nicola and C. Clopath, "Supervised learning in spiking neural networks with force training," *Nature communications*, vol. 8, no. 1, p. 2208, 2017.

[6] K. Pozo and Y. Goda, "Unraveling Mechanisms of Homeostatic Synaptic Plasticity," *Neuron*, vol. 66, no. 3, pp. 337–351, 2010.

[7] G. G. Turrigiano, "The self-tuning neuron: synaptic scaling of excitatory synapses," *Cell*, vol. 135, no. 3, pp. 422–435, 2008.

[8] A. P. Johnson, J. Liu, A. G. Millard, S. Karim, A. M. Tyrrell, J. Harkin, J. Timmis, L. McDaid, and D. M. Halliday, "Homeostatic Fault Tolerance in Spiking Neural Networks utilizing Dynamic Partial Reconfiguration of FPGAs," in *31th International Conference on VLSI Design and 17th International Conference on Embedded Systems (VLSID)*, Jan. 2018.

[9] A. P. Johnson, D. M. Halliday, A. G. Millard, A. M. Tyrrell, J. Timmis, J. Liu, J. Harkin, L. McDaid, and S. Karim, "An FPGA-based Hardware-Efficient Fault-Tolerant Astrocyte-Neuron Network," in *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, Dec. 2016, pp. 1–8.

[10] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[11] J. Qiu, J. Wang, S. Yao, K. Guo, B. Li, E. Zhou, J. Yu, T. Tang, N. Xu, S. Song, Y. Wang, and H. Yang, "Going Deeper with Embedded FPGA Platform for Convolutional Neural Network," in *Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, ser. FPGA '16. New York, NY, USA: ACM, 2016, pp. 26–35.

[12] S. Haykin and B. Van Veen, *Signals and systems*. John Wiley & Sons, 2007.

[13] E. A. Bushong, M. E. Martone, Y. Z. Jones, and M. H. Ellisman, "Protoplasmic astrocytes in CA1 stratum radiatum occupy separate anatomical domains," *Journal of Neuroscience*, vol. 22, no. 1, pp. 183–192, 2002.

[14] J. J. Wade, L. J. McDaid, J. Harkin, V. Crunelli, and J. S. Kelso, "Bidirectional coupling between astrocytes and neurons mediates learning and dynamic coordination in the brain: a multiple modeling approach," *PloS one*, vol. 6, no. 12, p. e29445, 2011.

[15] S. Jäkel and L. Dimou, "Glial Cells and Their Function in the Adult Brain: A Journey through the History of Their Ablation," *Frontiers in Cellular Neuroscience*, vol. 11, no. 24, pp. 1662–5102, 2017.

[16] G. Perea, R. Gómez, S. Mederos, A. Covelo, J. J. Ballesteros, L. Schlosser, A. Hernndez-Vivanco, M. Martín-Fernndez, R. Quintana, A. Rayan, A. Díez, M. Fuenzalida, A. Agarwal, D. E. Bergles, B. Bettler, D. Manahan-Vaughan, E. D. Martín, F. Kirchhoff, and A. Araque, "Activity-dependent Switch of GABAergic Inhibition into Glutamatergic Excitation in Astrocyte-neuron Networks," *Elife*, vol. 5, pp. 1–26, Dec. 2016.

[17] E. L. Bienenstock, L. N. Cooper, and P. W. Munro, "Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex," *Journal of Neuroscience*, vol. 2, no. 1, pp. 32–48, 1982.

[18] M. Bear and F. Ebner, "A physiological basis for a theory of synapse modification," *WORLD SCIENTIFIC SERIES IN 20TH CENTURY PHYSICS*, vol. 10, pp. 121–130, 1995.

[19] A. P. Johnson, J. Liu, A. G. Millard, S. Karim, A. M. Tyrrell, J. Harkin, J. Timmis, L. J. McDaid, and D. M. Halliday, "Homeostatic Fault Tolerance in Spiking Neural Networks: A Dynamic Hardware Perspective," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. PP, no. 99, pp. 1–13, Jul. 2017.

[20] W. Gerstner and W. M. Kistler, *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge university press, 2002.