Optogenetics in Silicon: A Neural Processor for Predicting Optically Active Neural Networks

Junwen Luo, Konstantin Nikolic, Benjamin Evans, Na Dong, Xiaohan Sun, Peter Andras, Alex Yakovlev and Patrick Degenaar

Abstract—**We present a reconfigurable neural processor for real-time simulation and prediction of opto-neural behavior. We combined a detailed Hodgkin-Huxley CA3 neuron model with the four-state Channelrhodopsin (ChR2) into reconfigurable silicon hardware. Our architecture consists of a Field Programmable Gated Array (FPGA) with tailor designed computing data-path, a separate data management system and a memory approach based router. The advancement over the prior art is the inclusion short and long –term calcium-and light-dependent ion channels in reconfigurable hardware. Also, the developed processor is computationally efficiency, and requires 0.03ms processing time per sub-frame for a single neuron and 9.7ms for a fully connected network of 500 neurons with a given FPGA frequency 56.7 *Mhz*. It can therefore be utilized for exploration of closed loop processing and tuning of wet optogenetic circuitry.**

Index Terms— **Optogenetics, ChR2, Neural Processor, FPGA, Neuromorphic Circuits, Neuroprothesis, Hodgkin Huxley**

I. Introdu ction

OPTOGENETICS is the genetic modification of cells to become sensitive to light by incorporating light-gated cation channel Channelrhodopsin ChR2 [1] or anion channels [2]. It has attracted interest from multiple disciplines, particularly due to its ability to genetically target neural sub-circuits, paving the way for high spatial and temporal resolution with perhaps better biocompatibility than with electrical approaches[1]. Some promising translational neuroprosthetic therapies to date include pacemakers for epilepsy [4][5] and visual prosthesis [6].

The kinetics of the molecule, were previously explored from an engineering viewpoint by ourselves (Nikolic et al. [7] and Grossman et al. [8]) using data from optogenetically transfected hippocampal cells. However there are still challenges to be addressed, such as how to physically stimulate large numbers of neurons. More importantly, how can therapeutic or scientific network stimulation protocols be translated into a particular three dimensional light pattern? Such questions will be application specific and can be answered empirically or through modelling. The latter would require accurate software models. To explore further, perhaps bio-silicon hybrid networks could be used, with the potential for exploring both basic science and downstream translation.

A range of methodologies exist to simulate and predict the state of neural networks. These differ in their accuracy of mathematical representation as well as their scope and range of biological features. Abstract models such as integrate and fire [9], Izhikevich [10], and Hindemarsh-Rose [11] provide computational efficiency. This allows scaling to large network simulations (of many thousands of neurons) on commodity hardware. There is however a need for more moderate sizes of neural networks but with bio-realism and real-time operation. In particular, optogenetics can provide stimuli to relatively localised neuronal circuitry. This requires the combination of optogenetic models with spatially detailed Hodgkin Huxley models of neurons [12]. Such a system could potentially interpret recordings and command stimulation equipment in real time (through closed loop control), and could be very useful to both the in-vitro [13] and in-vivo communities [14].

Previously computer workstations have been used to achieve high speed computation of moderately complex neural networks. This is particularly the case when Graphics Processing Units (GPU's) are used for their parallel processing capability: Fidjeland used a GPU kernel to simulate 55,000 neurons with 1,000 connections per neuron under bio-plausible conditions [15]; Wang implemented a network with 1 million HH based neurons on a commodity GPU, achieving a 28x speed-up over CPU implementations [16], and Tadashi applied a cerebellum gain and timing control algorithm on a GPU for real-time processing. However, with this technique it is difficult to achieve accurately timed output states for stimulation in real time using computational systems with operating systems. Therefore further digital logic is required to provide buffering and timing accuracy in the stimulus. The motivation for this work is that it could be beneficial for timing accuracy to also put the neural network processing in this digital logic layer, and use the computer for updating variables associated with the neurons and network.

One of the most appealing solutions for creating such a digital implementation is via reconfigurable logic, and in particular with a Field Programmable Gated Array (FPGA). FPGA's consist of arrays of logic and memory elements which can be defined as particular digital elements and connected in highly parallelized forms. These allow for rapid bespoke prototyping of digital circuits and their relative connectivity. As they are reprogrammable, they can be re-tuned to whatever neural network configuration is required. The downside of FPGA's is that classically their relatively high power consumption means that their application is limited to the benchtop. This is still acceptable for in-vitro applications

Junwen Luo, Alex Yakovlev and Patrick Degenaar are with School of Electrical and Electronic Engineering, Newcastle University, NE1 7RU, UK. (e-mail: {jun-wen.luo; alex.yakovlev, Patrick.degenaar}@newcastle.ac.uk).

Konstantin.nikolic and Benjamin.evans are with the Institute of Biomedical Engineering, Imperial College London, SW7 2AZ, UK (e-mail: {k.nikolic, benjamin.evans} imperial.ac.uk).

Peter Andras is with School of Computing and Mathematics, Keele University, ST5 5BG, UK (e-mail: p.andras@keele.ac.uk).

Na Dong and Xiaohan Sun are with school of Electrical Engineering, Southeast University, China (e-mail: xhsun@seu.edu.cn, dongna2008@gmail.com)

however, and more recently non-volatile forms of FPGA's provide low power operation suitable for battery-based applications.

FPGA systems have already been used to implement the Hodgkin-Huxley (HH) model, albeit with only voltage-dependent ion channels: Smaragdos implemented an olivocerebellar 92-neuron network using a three-compartment HH model [17]; Weinstein et al. developed a system level design flow for implementing voltage-dependent ion channels [18]; and Graas et al. presented a timing multiplexing technique to process multi-neuron activities sequentially [19].

In this work, we have developed an FPGA based highly biologically plausible processor for real-time simulation of optogenetic neural networks. Fig. 1 depicts the opto-neural architecture. The neuron itself is modelled from the Traub mammalian interpretation of the HH model [20], but also incorporates calcium and calcium dependent ion channels [21]. The inserted opsin is modelled using our previous four-state Channelrhdodopsin-2 (ChR2) model [7].

The first key advancement of this work lies in how we implement in-silico a biologically realistic neuron model with our four-state ChR2 model[7]. In addition, we have incorporated calcium and calcium-dependent channel models from both Traub et al. [20] and Soto-Treviño et al. [21]. As Calcium is an important ion for neuronal adaptation (and often also imaging). We believe this to be useful and important.

Our model can be adapted to represent most forms of optogenetic channels by modifying the time-constants, reversal potential and conductance to capture the dynamics of other variants. Therefore, compared to the other FPGA based neural systems, the developed hardware architecture includes short and long-term calcium- and light-dependent ion channel, which are capable of replicating more advanced neural characters(*e.g.* light-to-spike processes and calcium-related adaption) in real-time.

The second key aspect to this system is its flexibility and computational efficiency. The data management system and configuration unit are separate to the computing data-path. Thus, the system application objectives can be easily updated by modifying corresponding implemented model characters (*e.g.* light irradiance, architecture, parameters and network sizes). For example, since each neuron is calculated sequentially, by pre-stored different level of light irradiance per neuron in data generation system, the system is able to simulate the effects of different light stimulation distribution over a population of neurons, which is quite useful for investigating multi-site light stimulation strategies for optogenetics.

Furthermore, our pipelined parallel processing requires only 0.03ms for a single neuron and 9.7ms for a fully connected 500-neuron network to calculate a simulation sub-frame. Thus the applicability of this system for either open or closed loop interaction with tissue is where the neuron count is in the hundreds of thousands rather than millions. Examples of this include active pixel sensor neural recording systems [22] and stimulation systems (e.g. by Wang et al. [23] and ourselves [24]).

It is also possible to directly translate the FPGA design into an Application Specific Integrated Circuit (ASIC) chip. In that instance, the chip would be sufficiently small, and low power for in-vivo applications.

II. Modelling the light-to-spike process

The optogenetic-neuron mathematical model has been adapted from previous work [7]. It combines a detailed Hodgkin-Huxley neuron model with parameters for a CA3 neuron [20], and integrates an additional ChR2 channel [7]. The structure is shown in Fig. 1, which consists of four compartments: the synapses, axon, dendrites and soma. In order to ensure hardware translation, we do not attempt to increase the number of compartments to reflect long neuronal arbors. Nevertheless, it is still significantly more accurate than for abstract point-neuron models.

A. Cell model: Soma and Dendrites

Our cell model is essentially a two-compartment neuron model: one compartment emulates the complete dendritic tree including synaptic inputs and the other compartment models the cell soma. Nominally there is a third compartment – the axon – but in our model it is treated as a simple communication contact, hence a separate compartment was not associated with it. The common ion channels for both the soma and dendrites are:

- The voltage-dependent ion channel: a sodium ion channel [Na^+], a calcium ion channel [Ca^{2+}], a delayed rectifier potassium ion channel [K^+(Dr)], and an A-type of transient potassium ion channel [K^+(A)].
- The calcium-dependent ion channel: a long duration Calcium-dependent potassium ion channel [K^+(AHP)], and a short duration Calcium-dependent potassium ion channel [K^+(C)].
- The light-dependent ion channel: [ChR2].

The light-dependent ion channels (ChR2) are assumed to be expressed only in the soma. We justify this as the surface area of the dendrites of any given cell is relatively small compared to the volume of tissue they inhabit, so optical stimulation is best targeted at the soma. We feel the computational cost is not justified by the small dendritic contribution of traditional ChR2, which has very low channel conductance. If however a high conductance opsin were to be used, these effects could be incorporated.

Synapses are assumed to be only in the dendrites. Similarly, this is to simplify the model computationally, but again, this can be easily changed if required.

The neuronal model is based upon the traditional HH differential equations [12] which treat individual channels as having an individual conductance with a specific reversal potential. The traditional model contains potassium, sodium and leakage ion channel components. We have also incorporated calcium and rhodopsin channels.

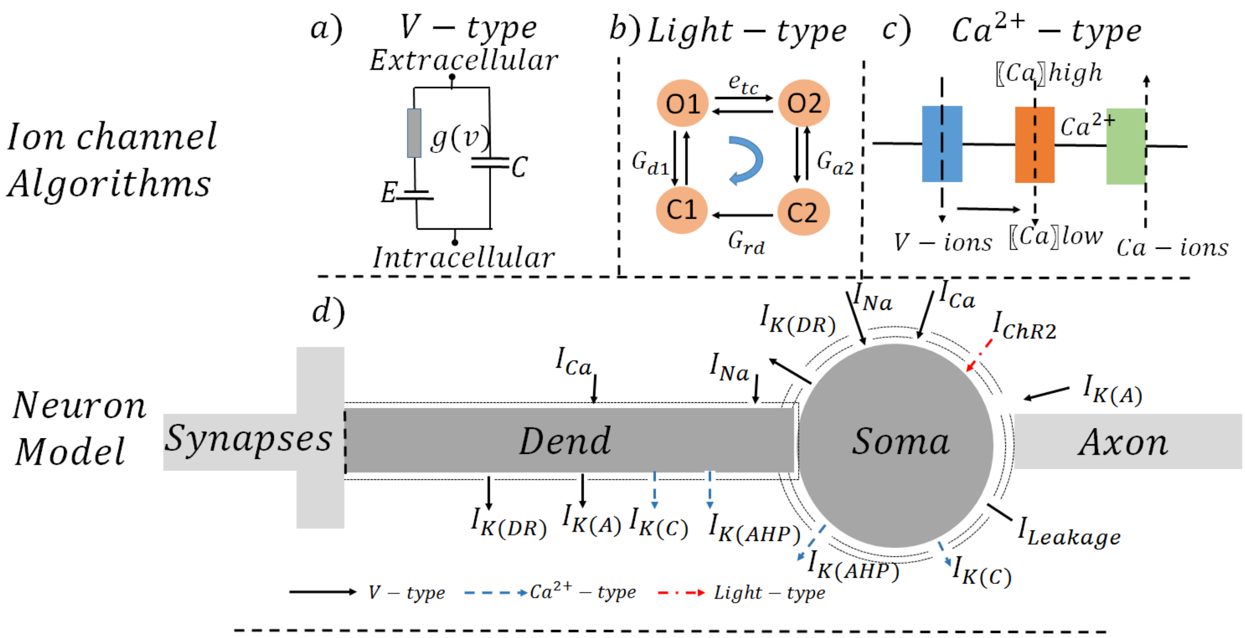


Fig. 1: An optogenetic-neuron architecture. The soma and dendrites contain three different types of ion channels: voltage-dependent ion channels, calcium dependent ion channels and light-dependent ion channels. The voltage-dependent ion channels are: a sodium ion channel I_{Na}, a calcium ion channel I_{Ca}, the delayed rectifier K^+ ion channel $I_{K(DR)}$, and the A-type of transient K^+ ion channel I_K(A) which are modelled using the HH equations. The calcium-dependent ion channels are a long duration Ca-dependent K ion channel $I_{K(AHP)}$, and a short duration Ca^{2+}-dependent K^+ ion channel $I_{K(C)}$. The Ca^{2+}-dependent ion channels depend on the current intracellular calcuim concentrations, typically calculated only in the cytoplasmic shell near the cell membrane as described in Traub et al. [20]. The light-dependent ion channel is ChR2, based on the four-state Markov process of Nikolic et al.[7]. The synapses receive synaptic currents from the other neurons and generated action potentials are transmitted along the axon

(1) and (2) describe the time evolution of the membrane potential for the soma compartment (dV_{soma}/dt) and for the dendritic tree compartment (dV_{dend}/dt) in terms of current flow through each channel:

$$C_{soma}\frac{dV_{soma}}{dt} = -(I_{syn} + I_{Na} + I_{Kdr} + I_{Ka} + I_{Kahp} + I_{Kc} + I_{Ca} + I_{ChR2} + I_{Leak} + \ g_c(V_{dend} - V_{soma})) \quad (1)$$

$$C_{dend}\frac{dV_{dend}}{dt} = -(I_{dNa} + I_{dKdr} + I_{dKa} + I_{dKahp} + I_{dKc} + I_{dCa} + I_{dLeak} + \ g_c(V_{soma} - V_{dend})) \quad (2)$$

The current terms are described in Table 1. The last term in both equations describes the current between the compartments. $g_c = 0.02\text{nS}/\mu\text{m}^2$ is the conductance between the soma and the dendrites (compartments), $C_{soma} = 0.01\ \text{pF}/\mu\text{m}^2$ is the membrane capacitance of the soma compartment, and $C_{dend} = 0.01\text{pF}/\mu\text{m}^2$ is the membrane capacitance of the dendritic compartment.

The mathematical equations for the current flow through the voltage dependent ion channels are given by (3) and (4)

$$I_i = g_i \times m_i^p h_i^q \times (v - E_i) \quad (3)$$

$$\frac{dm(h)}{dt} = m(h)_\infty \times \big(1 - m(h)\big) - \ m(h)_\tau \times m(h) \quad (4)$$

Where I_i is the ion channel current, g_i is the ion conductance, m and h are gate variables (where h has the same form as m) and $m(h)_\infty$ and $m(h)_\tau$ are the gate-variable steady-state and time constant values respectively. Finally, v is the reduced membrane potential ($v = V - V_{rest}$) and E_i is the reduced reversal potential.

An empirical equation for intracellular calcium concentration $[Ca^{2+}]$ was proposed by Traub et.al. [20] and shown here in (5):

$$\frac{d[Ca^{2+}]}{dt} = -FI_{Ca} - [Ca^{2+}]/\tau_{Ca} \quad (5)$$

Where $F = 3$ is the scaling constant, and $\tau_{Ca} = 13.33$ ms is the time constant for the decay of intracellular calcium concentration, due to the rapid action of ion pumps which extrude calcium. The corresponding parameters are shown in Table 1 and Table 2.

The four-state model of Channelrhodopsin-2 was previously described by ourselves (Nikolic et al.[7]), which we believe to be optimal in terms of the balance between accuracy and simplicity. The model describes ChR2 as having four states; two dark states and two activate (conductive) states, and is shown in Fig. 1b .

The retinal molecular core of the ChR2 rhodopsin complex absorbs a photon to switch from all-trans to 13-cis-retinal.

Table 1: Parameter values of voltage- and calcium-dependent ion channels

		Soma		Dendrites	
		$g(nS/\mu m^2)$	$E(mV)$	$g(nS/\mu m^2)$	$E(mV)$
Na+ channel	I_{Na}	0.3	115	0	115
Delayed rectifier K+ channel	I_{Kdr}	0.15	-15	0	-15
A-type K+ channel	I_{Ka}	0.05	-15	0	-15
Long-term Ca2+-dependent K+ channel	I_{Kah}	0.008	-15	0.008	-15
Short-term Ca2+-dependent K+ channel	I_{Kc}	0.1	-15	0.05	-15
Ca2+ channel	I_{Ca}	0.04	140	0.02	140
Non-specific membrane leakage	I_{Leak}	0.001	-12.5	0.001	-12.5

Table 2 Activation and inactivation variable rate functions

	$Forward(\alpha)$	$Backward(\beta)$
$I_{Na}(m)$	$\frac{0.32(13.1-v)}{\exp\left(\frac{13.1-v}{4}\right)-1}$	$\frac{0.28(v-40.1)}{\exp\left(\frac{v-40.1}{5}\right)-1}$
$I_{Na}(h)$	$0.128\exp\left(\frac{17-v}{18}\right)$	$\frac{4}{1+\exp\left(\frac{40-v}{5}\right)}$
$I_{Kdr}(m)$	$\frac{0.016(35.1-v)}{\exp\left(\frac{35.1-v}{5}\right)-1}$	$0.25\exp\left(\frac{20-v}{40}\right)$
$I_{Ka}(m)$	$\frac{0.02(13.1-v)}{\exp\left(\frac{13.1-v}{10}\right)-1}$	$\frac{0.0175(v-40.1)}{\exp\left(\frac{v-40.1}{5}\right)-1}$
$I_{Ka}(h)$	$0.0016\exp\left(\frac{-13-v}{18}\right)$	$\frac{0.05}{1+\exp\left(\frac{10.1-v}{5}\right)}$
$I_{Kahp}(m)$	$\min(0.2\times Ca^{2+}\times 10^{-4}, 0.01)$	0.001
$I_{Kc}(m)$	$\frac{\exp[\left(\frac{v-10}{11}\right)-\left(\frac{v-6.5}{27}\right)]}{18.975}$ $v\le 50;$ $2\exp\left(\frac{6.5-v}{27}\right), v>50;$	$2\exp\left(\frac{6.5-v}{27}\right)-\alpha_{kc}$, $v\le 50;$ $0,\ v>50;$
$I_{Ca}(m)$	$\frac{1.6}{1+\exp(-0.072(v-65))}$	$\frac{0.02(v-51.1)}{\exp\left(\frac{v-51.1}{5}\right)-1}$
$I_{Ca}(h)$	$0.005,\ v\le 0;$ $\frac{exp(-v/20)}{200}, v>0;$	$0,\ v\le 0;$ $0.005-\alpha_{Ca}$, $v>0;$

Table 3: The parameters of ChR2 model

$\tau_{ChRL}(ms)$	$\tau_{ChRD}(ms)$	$G_{rd}(ms^{-1})$	$e_{ct}(ms^{-1})$	$e_{tc}(ms^{-1})$
1.3	0.3	1/3000	0.053	0.023
$G_{d1}(ms^{-1})$	$G_{d2}(ms^{-1})$	$A_{ChR2}(\mu m^2)$	$v_0(mV)$	$v_1(mV)$
0.13	0.0025	5000	43	70
$\bar{g}_{ChR2}\ (nS/\ \mu m^2)$			$E_{ChR2}\ (mV)$	
0.0025			70	

This induces the channel to switch from a dark-adapted OFF state [C1] to a dark-adapted ON state [O1]. If illuminated in this ON state there is a chance of further photon absorption. This would transition the ChR2 from a dark-adapted ON state [O1] to a less conductive light-adapted ON state [O2]. From there it may thermally transition back to [O1] or decay to the light adapted OFF state [C2]. The [C2] state slowly reverts to the [C1] state (on the order of seconds) by thermal means. These relations can be described as four coupled differential equations:

$$\frac{dC1}{dt} = G_{rd}C2 + G_{d1}O1 - G_{a1}(t)C1 \tag{6}$$

$$\frac{\mathrm{d}O1}{\mathrm{dt}} = \mathrm{G_{a1}(t)C1 - (G_{d1} + e_{ct})O1 +\ e_{tc}O2} \tag{7}$$

$$\frac{dO2}{dt} = G_{a2}(t)C2 - (G_{d2} + e_{tc})O2 +\ e_{ct}O1 \tag{8}$$

$$\frac{dC2}{dt} = G_{d2}O2 - (G_{a2}(t) + G_{rd})C2 \tag{9}$$

$$G_a(t) = \begin{cases} \varepsilon F[1-\exp(-t/\tau_{ChR})], & t\le\ t_{light} \\ \varepsilon F\left[\exp\left((t-t_{light})/\tau_{ChR}\right)-\exp(-t/\tau_{ChR})\right], & >\ t_{light} \end{cases} \tag{10}$$

$$I_{ChR2} = (O1+\gamma O2)\times A_{ChR2}\times\bar{g}_{ChR2}\times(V-E_{ChR2})\times\frac{1-\exp(-\frac{V-E_{ChR2}}{v_0})}{(V-E_{ChR2})/v_1} \tag{11}$$

Where $O1$, $O2$, $C1$ and $C2$ are the proportions of ChR2 rhodopsins in the open states (1 and 2), and closed states (1 and 2), which are conserved to sum to one. G_{d1} and G_{d2} are the deactivation rates O1→C1 and O2→C2 respectively, and e_{tc} and e_{ct} are the rates of transition between O1 and O2 and vice versa and G_{rd} is the rate of thermal conversion of C2 to C1. Ga1 and Ga2 are the activation rates for C1 to O1 and C2 to O2 respectively (described in general terms in (10)), $\gamma = 0.05$ is the conductance ratio of O1 and O2. F is flux in photons per ChR2 per millisecond and ε is the quantum efficiency of the rhodopsin. V is the membrane potential of a neuron (in mV), v_0 and v_1 are empirical constants equal to 40 mV and 15 mV and E_{ChR2} is the channel reversal potential, equal to 0 mV. The ChR2 channel maximum conductance per unit area, $\bar{g}_{ChR2} = 2.5\,pS/\mu m^2$ is multiplied by the ChR2 expression area A_{ChR2} to find the total channel conductance for the cell. The corresponding rate parameters are given by Table 3.

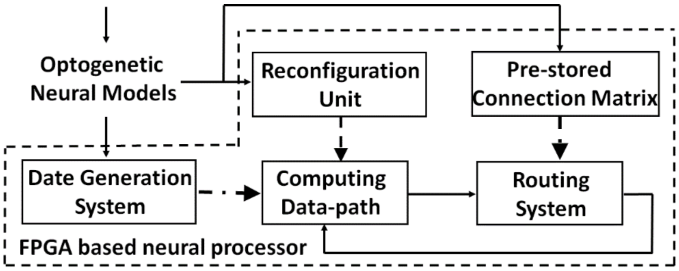

Optogenetic Neural Models
Reconfiguration Unit
Pre-stored Connection Matrix
Date Generation System
Computing Data-path
Routing System
FPGA based neural processor

Fig. 2: A conceptual architecture of an FPGA based neural processor. It consists of three main parts: data generation system, reconfiguration unit, computing data-path and routing system.

B. Synapses

The synapse model is described in (12):

$$I_{syn}^{i} = -\sum_{j=1}^{n} \delta_{ij}(t - t_j') \times \bar{g}_i \times \varepsilon \times (v_i - E_s) \qquad (12)$$

Where I_{syn}^{i} indicates the total synaptic currents received by the neuron, n is the number of presynaptic neurons, indexed by j, with their train of spike times represented by t_j', $\bar{g}_i$ is the maximum synaptic conductance of each postsynaptic neuron, and ε is the transmission efficiency. Spike events are represented by δ_{ij}, a Dirac-delta function, which is 1 at the time of a presynaptic spike (i.e. when $t - t_j' = 0$) or 0 otherwise. Our purpose here is to explore the network dynamics rather than learning processes, however they could be included later using synaptic potentiation/depression models from [25].

C. Axon

Cable theory such as described by Wilfrid [26] can be used to simulate axonal transmission. Its incorporation would allow for more detailed timing studies between synaptic connections *e.g.* spike correlated timing. However, the partial derivative calculations would increase the required FPGA resources. We believe that the cost outweighs the benefits.

As with other neuron network systems, we assume a transmission channel efficiency is 100%, i.e. no spike loss between soma and synapse. The transmission time is essential one simulation frame.

If transmission delays are important to study e.g. for rank [27] or phase coding [28], then they are best introduced as direct network delays. Our system can be reconfigured to interpret this behaviour, but at the cost of additional memory blocks, which would reduce the maximum implementable network size.

III. Neural processor architecture

The neural processor mainly contains three components: the computing data-path, the data generation and reconfiguration units, and the router.

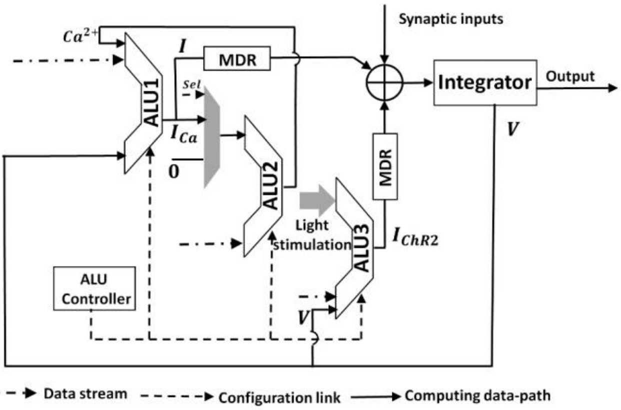


Fig. 3: The computing data-path component. It contains three different algorithm logic units (ALUs), one ALU controller and memory data registers.

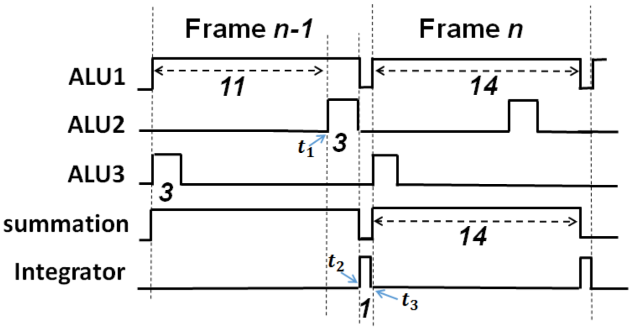


Fig. 4: The timing diagram of the developed neural processor. The algorithm logic units ALU1, ALU2 and ALU3 require 14, 3 and 3 clock clycles in each frame. The processes are integrated at the last clock cycle in each frame.

The processor architecture is shown at Fig. 2. The computing data-path is specifically designed for calculating the previously described mathematical equations (for details see Section 2.1), the data generation system aims to deliver all the required neuronal fixed model parameters to the different data-paths at the corresponding time, the reconfiguration unit is to modify the computing data-path based on the models, and the router is for implementing the network's synaptic connections.

The FPGA design utilizes 40-bit point numbering system and 22 bits available for fractions. Therefore, the parameter dynamic range can be calculated is from-361 to 361 with a resolution of 10^{-4}, and the output membrane potential v is from -50 to 150, which depends on the implemented LUT sizes. In addition, these parameter value are implemented by using flip flop and LUT slices.

A. Computing data-path

The computing data-path has three separate algorithm logic units (ALUs), which are shown in Fig. 3. Here ALU1 is for calculating voltage-dependent ion channel equations (3–4), ALU2 is for calculating calcium-dependent functions (5) and ALU3 is for calculating the ChR2 state variables (7–9). (N.B.

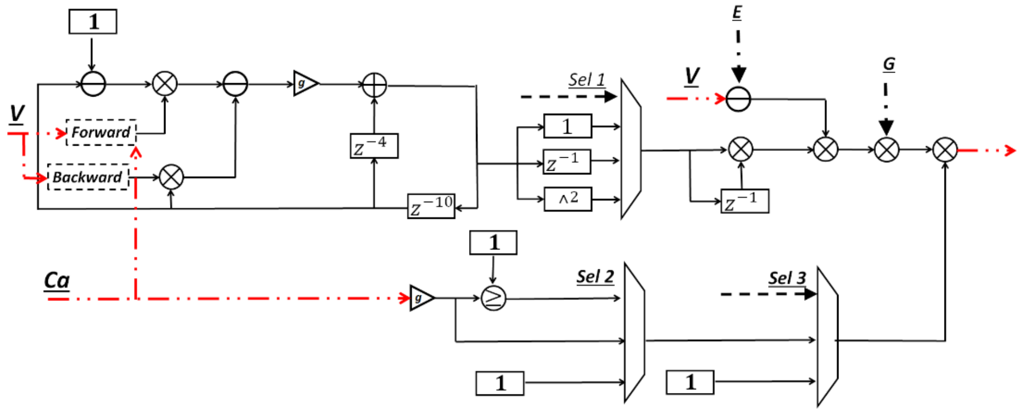


Fig. 5: The ALU1 hardware architecture. This unit aims to calculate voltage-dependent ion channel activities. The dashed arrows indicate system inputs and outputs and Sels represent configuration signals. The forward and backward slices are for calculating the activation and inactivation rates shown in Figure 1. E and G are ion channel reversal potential and maximum conductances, while V and Ca are neuron membrane potential and calcium concentrations used as inputs.

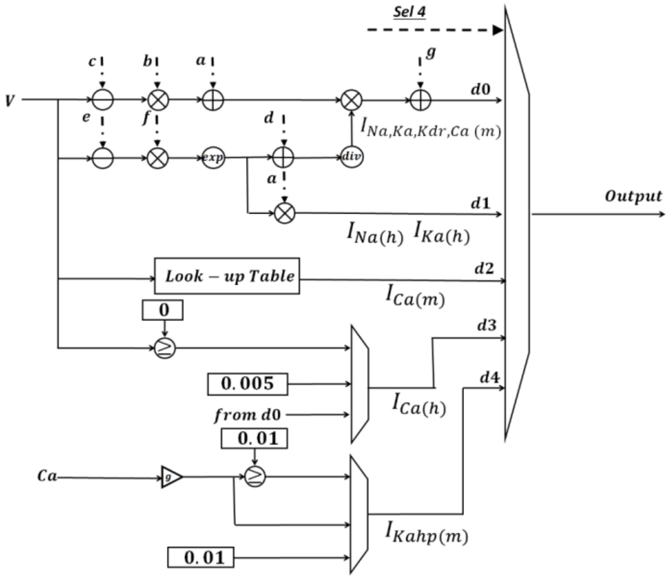


Fig. 6**:** The hardware architecture of the forward (backward) slice. Five different gate variable calculation styles can be selected in a system depending on Table 2. $a, b, \ldots, g$ are the ion channel gate parameters. $d0$ responses for calculating forward variables of $I_{Na,Ka,Kdr,Ca(m)}$, d1 is for $I_{Na,Ka(h)}$; d2 is d3 is for $I_{Ca(h)}$ and d4 is for $I_{Kahp(m)}$. Specifically, a Look-Up-Table block is employed for calculating short duration Ca-dependent ion channel gate variables.

For simplicity of implementation, the fourth differential equation for $C1'$ is eliminated by substitution, since by conservation of the states, it is equal to $C1 = 1 - O1 - O2 - C2$.) Each ALU receives two types of signal: the first are the data stream signals from the data generation systems, determined by the software model parameters. The second are the switch configuration link signals from the configuration unit, determined by the software model architecture and applications. The memory data register (MDR) is applied to maintain an equal latency for the different data-paths.

Since this architecture is pipelined, ion channels are calculated sequentially. These ALUs have to perform their calculations in a specific sequence to simulate the interactions between different types of ion channels. This timing diagram is shown at Fig. 4.

In this design, ALU1 calculates the voltage- and calcium-dependent ion channel activity in 14 clock cycles. When I_{Ca} results are released at time point t_1, ALU2 receives the values to calculate the calcium concentration, and the outputs at time point t_2 are feedback to ALU1 for computing calcium-dependent ion channels. In parallel, ALU3 calculates the ChR2 current based on the current membrane potential and light stimulation. At the 15th clock cycle at time point t_3, the integrator sums the outputs from ALU1 and ALU3 for the final output and the system performs the next frame calculation.

The ALU1 hardware architecture is shown at Fig. 5, which implements (3) and (4). During the process, the neural parameters (*e.g.* G, V) are released sequentially for calculation. A complete frame comprises of 14 clock cycles. There are also 3 switch control signals for the gate variable exponential (Sel1) and calcium calculation styles (Sel2 and Sel3). In the system the forward and backward slices are for calculating the activation and inactivation rate equations as shown in Table 1. As described in Fig. 6, five different gate variable calculation styles can be selected in a system depending on the select signals. Particularly, styles d0, d1 and d3 share the common data-paths. Overall ALU1 has 4 configurations and 10 data stream signals. Specifically, when I_{Ca} is calculated from the ALU1, the Sel signal in Fig. 3 will activate and send it to ALU2 for calcium computing. At the same time, the Sel signal will send ALU2 into an inactive state.

The data-path of ALU3 is shown at Fig. 7. The values G_{a1} and G_{a2} are pre-calculated and depend on the input light irradiance. The three coupled differential equations ((7)-(9)) are implemented to simulate the ChR2 four-state model's dynamic behaviour. The overall latency is optimized to 3 clock cycles, and the time-step for numerical integration is set

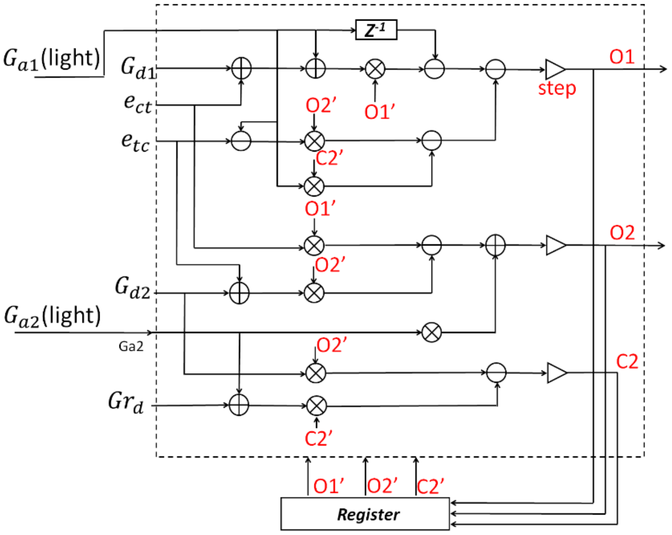


Fig. 7: Data-path of the ChR2 computing block. The mathematical descriptions are given in Equations 7–9. Where $\mathbf{O1}, \mathbf{O2}, \mathbf{C1}$ and $\mathbf{C2}$ are the numbers of ChR2 molecules in the current open states 1 and 2, and closed states 1 and 2. $\mathbf{G_{d1}}$ and $\mathbf{G_{d2}}$ are the transition rates for $\mathbf{O1} \rightarrow \mathbf{C1}$ and $\mathbf{O2} \rightarrow \mathbf{C2}$, $\mathbf{e_{tc}}$ and $\mathbf{e_{ct}}$ are the transition rates between $\mathbf{O1}$ and $\mathbf{O2}$ and vice versa. $\mathbf{G_{a1}}$ and $\mathbf{G_{a2}}$ are the activation rates of $\mathbf{C1} \rightarrow \mathbf{O1}$ and $\mathbf{C2} \rightarrow \mathbf{O2}$.

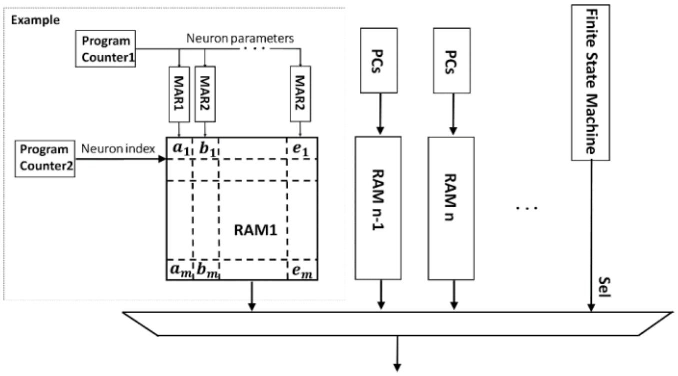


Fig. 8: Data generation system. The RAM is used for storing model parameters such as activation (inactivation) rate parameters *(e.g.* $\mathbf{a}, \mathbf{b}, \ldots, \mathbf{e}$) and ion channel conductance. The PCs are program counters where PC1 is an index of different parameters of a neuron, and PC2 is an index of different neurons in a network. A Finite State Machine (FSM) controls the system and the MARs are the memory address registers.

to 50μs. For each loop, the previous state values $O1'$, $O2'$ and $C2'$ are used with the current light stimulation levels to generate the ChR2 outputs. More importantly, ALU3 is only active for 1 clock cycle in a frame (14 clock cycles) due to the parallel implementation and computing pipeline.

B. *Data generation system*

The data generation system is shown in Fig. 8. As can be seen, it contains n individual units and a Finite State Machine (FSM). Each unit has one RAM cell and two program counters (PCs). The RAM is used for storing model parameters such as activation (inactivation) rate parameters (*e.g.* $a, b, \ldots, e$) and ion channel conductance. PC1 is an index of the different parameters of a neuron, and PC2 is an index of different neurons in a network. An FSM is employed as a control signal to select corresponding RAM states as output values. Specifically, the FSM decides frame and sub-frame control signals. In addition, memory address registers (MAR) are implemented based on the latency in the computing data-path. Since the system uses different sub-block RAM rather than an entire one, data management becomes more efficient and controllable. In a similar manner, the reconfiguration unit shares this technique with the data generation system.

Therefore, the simulator can handle biologically realistic situations which originate from uneven light distribution and/or ChR2 expression: the light simulation intensities for different neurons can be stored in these units as well, the PCs as the address index response for sending light-dependent parameters at the correcting timing, while the other PCs in the reconfiguration units able to turn on/off the ChR2 channel at specific time periods, which decided pulse width.

C. *Routing system*

The routing system is shown in Fig. 9. Spike events from three processors are sequentially sent into shift registers for processing, and the results are fed-back individually. The basic mechanism is as follows: when a neuron spike event (1 or 0) arrives, its corresponding post-neuron location (*e.g.* $1 \rightarrow 1, 1 \rightarrow 2, \ldots, 1 \rightarrow n$) will be addressed by the neuron index. By multiplying the synaptic strength pre-stored in the LUT and the spike event, the updated synaptic inputs are stored in the RAM block at the same location (*e.g.* $1 \rightarrow 1^t, 1 \rightarrow 2^t, \ldots, 1 \rightarrow n^t$). After calculating the states for all the neurons in the network, the accumulator adds all the received synaptic inputs per neuron for the next frame calculation (the process happens at the last sub-frame periods).

For example, for neuron index 1, the all synaptic currents $(1 \rightarrow 1^t, 2 \rightarrow 1^t, \ldots, n \rightarrow 1^t)$ will be accumulated and represented by $(:,1)^t$. Two memory data registers are implemented for storing the accumulator results. One is for sending previous frame synaptic inputs (e.g. $(:,1)^{t-1}$) to the calculated neuron, the other one is for storing currently summed synaptic inputs (*e.g.* $(:,1)^t$) for computing in the next frame. The frame period is a product of total neuron number and processing time per neuron.

IV. Results

A. *ChR2 ion channel*

The individual silicon ChR2 channel simulation results are shown in Fig. 10. Light pulses of seven durations are used in this experiment: 1, 2, 3, 5, 8, 10 and 20 ms.

The FPGA simulation indicate that the developed silicon ChR2-HH Neuron behaves similarly to its biological counterpart, on which the software model is based (data not shown but can be seen in [7]). There are some slight differences between the model and the FPGA implementation, especially at 2 and 3ms light pulses, due to the digital truncation errors and fixed step integration.

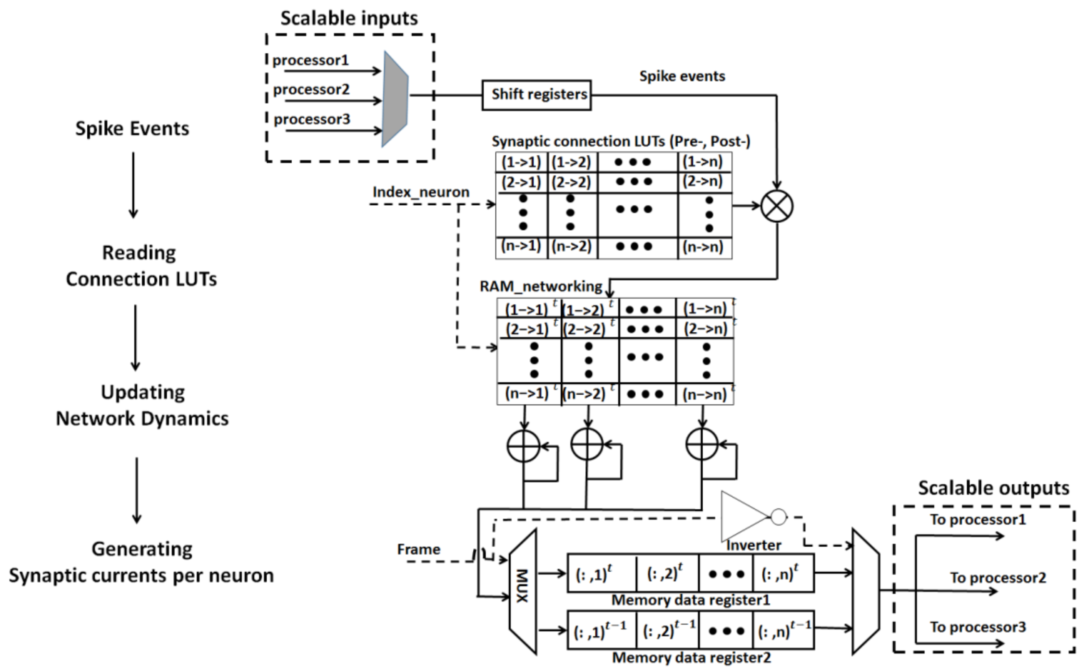


Fig. 9: Routing system. It contains a ROM based Look-up Table for storing network connectivity, RAM for updating synaptic events, and two registers for data management. In the synaptic connection LUT, location $(\mathbf{1}\rightarrow\mathbf{2}), (\mathbf{1}\rightarrow\mathbf{3})$ stores the maximum synaptic conductance of neuron index 1 to 2 and 1 to 3, and the RAM_networking LUT records the synaptic current values at time t for neuron index 1 to index 2 and index 3.

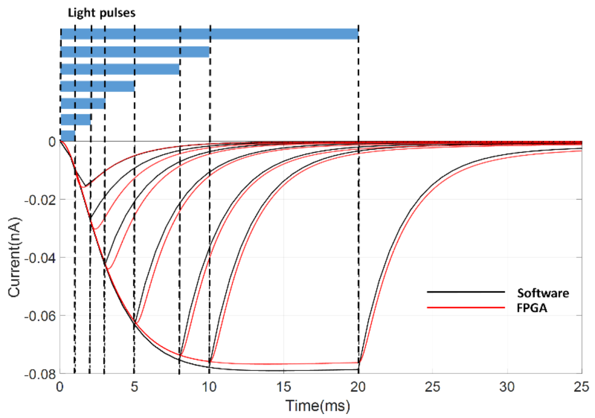


Fig. 10: The hardware simulation results of ChR2. At the left and right figure: the comparisons between the software model developed previously by Nikolic et al[7] and FPGA simulation results. The short light pulses are 1, 2, 3, 5, 8, 10 and 20 ms. The software fitting parameters used are $\boldsymbol{\tau}_{\mathbf{ChR}}$ = 1.3ms, $\boldsymbol{\gamma}$ = 0.1, $\mathbf{e_{ct}}$ = 0.01, $\mathbf{e_{tc}}$ = 0.02, Gd1 = 0.35 $\mathbf{ms^{-1}}$, Gd2 = 0.02 $\mathbf{ms^{-1}}$, $\mathbf{I_{max}}$ = 0.2 nA.

B. Hippocampal CA3 neurons

Voltage dependent and [ChR2-expressing+voltage-dependent] hippocampal CA3 neurons have been simulated for comparison. These are compared with data from previously published work [7] in Fig. 11. Fig. 11A and B, show oscilloscope readings of our neuron in response to constant and pulsed electrical stimulation (duty cycle = 50% with injected current 0.1 nA): the red line is the membrane potential and the blue line represents the electrical pulses. Fig. 11C and D, show oscilloscope readings of our neuron's response to constant and pulsed light stimulation (duty cycle = 50% with light irradiance $0.4mW/mm^2$): the red line is the action potential and green line is the ChR2 current. And the results are reasonable compare to the biological experiments[[29].

Furthermore, a comparison between software (simulated with Matlab) and hardware firing rates is shown in Fig. 12. For an electrical stimulus, as the stimulus strength increases, the firing rate increases accordingly. When the injected current exceeds 0.6nA, the CA3 neuron approaches its saturation and the firing rate collapses. For the light-based stimulus, the firing rate increases with light intensity (from 0.01 to 10 $\mathrm{mW/mm^2}$) and duty cycle (from 10% to 80%). In both conditions software and hardware systems show identical and biologically realistic results.

C. Optogenetically transfected neural network

We simulated a 25-neuron opto-neural network. Each neuron receives different light stimulation as shown in Fig. 13A. Each neuron randomly connects to 16-17 neurons on average with maximum synaptic conductance of 0.01 $\mathrm{nS/\mu m^2}$. The neural response in isolation to network connectivity in shown in Fig. 13B. This is, as would be expected, similar to the original optical irradiance light patterns: only five neurons with light-stimulation above threshold ($0.4\mathrm{mW/mm^2}$) have significantly firing rates, while the others remained silent.

The network dominating condition is shown in Fig. 13C. In this case, the synapses are all positive. i.e. no negative feedback. It can be seen the average firing rate is 45 Hz and

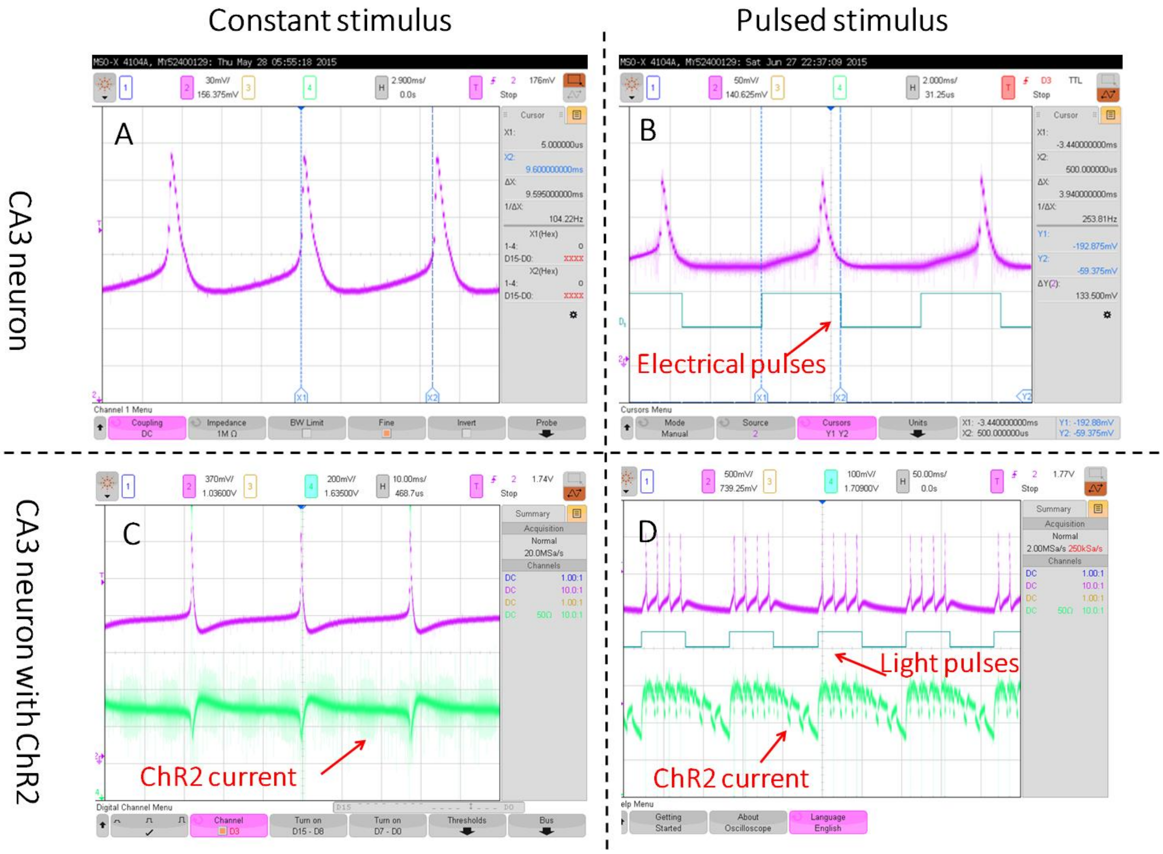


Fig. 11: The hardware experimental results of the neural processor. A: the results of a single neuron with constant stimulus (0.1 nA); B: the results of a single neuron with pulsed electrical stimulation (duty cycle = 50% with injected current 0.1 nA); C the results of a single optogenetic neuron with constant light stimulation ($0.4\boldsymbol{mW/mm^2}$); D the results of a single optogenetic neuron with pulsed light stimulation (duty cycle = 50% with light irradiance $0.4\boldsymbol{mW/mm^2}$). The FPGA digital signals are converted into analogue signals by using an external DAC based on a CY3214-PSoCEVALUSB PSoC1 development board.

the light pattern can no longer be seen. In this scenario, the radiant patter has an effect, but on the overall firing rate rather than a spatial pattern of activity, which is determined by the synaptic connections.

And Fig. 14 shows an interesting example of Neuron (1,1) where the two scenarios above are moderated such the firing behavior is supported by both surrounding network activity and optical stimulus. i.e. the optical stimulus on its own would not present significant firing activity.

V. Discussion

A. System scalability

We implemented different numbers of neuron on the FPGA processor to test the system's scalability by measuring the wall-time required for the system to generate a single spike (sub-frame). As shown at Fig. 15, the processor wall-time increases linearly with neuron number (blue line). This is because the calculations are sequential. In contrast, the router time depends exponentially on the number of neurons due to the memory based approach (where all the connections are pre-stored in the LUTs). At cross point B, the routing computing period exceeds that of the neural processing. At cross point C, the maximum number of neurons which can be implemented on the processor for real-time computing can be seen to be 500, for which the simulation time is 9.7ms (assuming the fastest biologically-realistic firing frequency is 100Hz).

Specifically, with fewer than 45 neurons, the network simulation time equals the processor time. This is because the processor and router compute in parallel in the hardware, and the routing period of a frame is less than a processor sub-frame period. However, with more than 45 neurons, indicated at cross point A, the system transitions from scaling linearly to non-linearly with the network size (neuron number). This is because the router requires more time for routing tasks compared to the processor's sub-frame periods at this stage, meaning that the processor has to wait until the router finishes its current frame tasks. Therefore, the system simulation performance will mainly depend on the router itself. Overall, the system performance exhibits a linear relationship to network size when it is below 45 neurons, and displays a non-linear relationship for more than 45 neurons (shown by the black line).

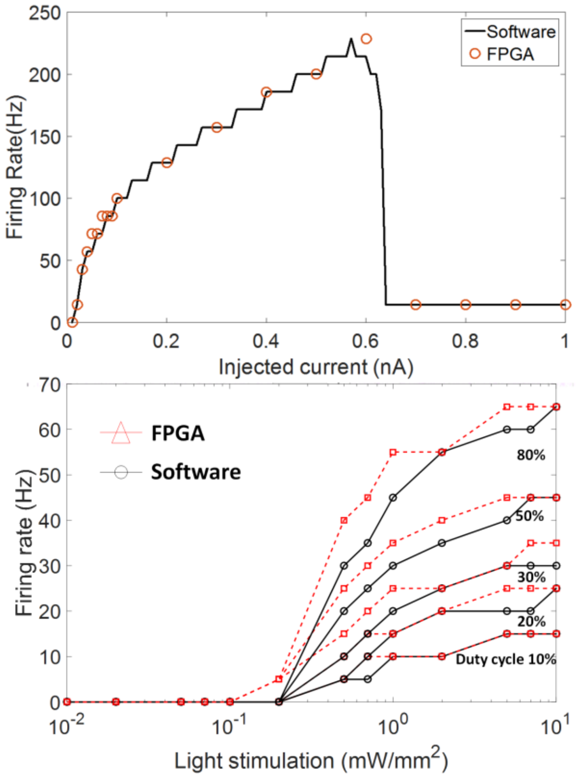


Fig. 12: A comparison of firing rates between (Matlab) software and hardware simulations. On the left, the injected currents are: 0.01 to 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9 and 1 (nA); and on the right, the light irradiances are 0.01, 0.02, 0.05, 0.07, 0.1, 0.2, 0.5, 0.7, 1, 2, 5, 7 and 10 (mW/mm^2). The duty period is 100 ms.

B. *Comparisons with the other techniques*

Comparisons between this work and previous FPGA neuron implementations are shown in Table 4. HH* indicates that a HH based model with three compartments, and HH+ represents our optogenetic-calcium enhanced model. Compared to the previous work, the major novelty of the presented work is that we include long- and short-term calcium- and light-dependent ion channels in the system. It is also more biologically realistic when compared to other abstract models.

The neuron model itself exerts a major influence on the hardware architecture design. General models with strong biophysical meaning have smaller time steps than mathematically abstract models: Izhikevich [30] and LIF [31] models have 1ms time step while HH [22][21] based models have time-steps ranging from 0.001 to 0.05ms. This is because complex neural model requires higher integration step resolution to compute the detailed ionic dynamics. As a result, the number of hardware operation in 1ms of biological time for bio-physical models is significantly larger than for the high-level phenomenological neuron models: LIF and Izhikevich based hardware implementations take only 30 and 13 operations to simulate 1ms of biological time, whereas the HH model with 3 compartments and our model require 22,200 and 11,880 operations respectively for the same period.

Another vital issue concerns the implementation of neural communication on the hardware. There are two major approaches for this: memory-based and routing-based. The memory-based approach uses on/off chip memory for pre-storing network connections. At each computing loop iteration, the neuron spike events will be sent to their postsynaptic-neuron targets according to their address packages (e.g. neuron and synaptic indices). Cheung12 [30] and our design follows this principle. It enjoys low latency and simple hardware design, but memory resource/bandwidth limits will be reached when the neuron number exceeds a certain threshold (dependent upon the resources of a particular FPGA). The other approach is to use a network-on-chip architecture; a tailor designed routing strategy implemented to deliver multi-core spike events in a system such as SpiNNaker platform [32]. Our previous work [33][34] employed this approach to implement cerebellum model [30] connections. It shows excellent system scalability but has more complex hardware design to ensure low latency. In addition, Randall07 [18], Andrew07 [31] and Georgious14 [17] implemented an all-to-all connection through their custom-designed techniques.

In prior work, different designs have used different methods to assess their relative computing performances. It is therefore hard to directly compare system speed and efficiency: Graas04 [19] proposed increasing FPGA clock frequency and the step size for the speed up of 40x real-time; Cheung12 [30] designed an event-driven and fully pipelined architecture for 2.48x real-time; Georgious14 [17] optimized their HLS C-code for 12.5x real-time. Fully pipelining and shortening the critical path are employed in our system speed optimizations.

There are also several different hardware platforms such as Spinnaker [32], Neurogrid [36], IFAT [37] and GPU [16] for neural modelling. Each system has strengths and weaknesses in particular areas. For example, Neurogrid and IFAT are mixed signal based architectures that are less reconfigurable but enjoy elegant design and efficient power consumption.

C. *Applications*

The developed hardware can serve as a multi-functional platform to investigate optogenetic related topics. Some of these potential applications are summarized in Table 5.

The first application is the investigation of optogenetic actuators such as channelrhodopsin, halorhodopsin [38] and archearhodopsin [39]. Depending on the required model, the ChR2 computing block can be easily re-configured to model other rhodopsins by updating its parameters and configuration signals. Also, since an optical–neural interface system [23] is hard to verify due to the complicated nature of the experiments, it would be useful to develop optogenetic hardware (*e.g.* optrode) functionality by using silicon networks at first. This will greatly speed-up development and improve the hardware success rates before investing time in biological experiments. Finally, as mentioned, society faces important challenges in fully realizing the potential of

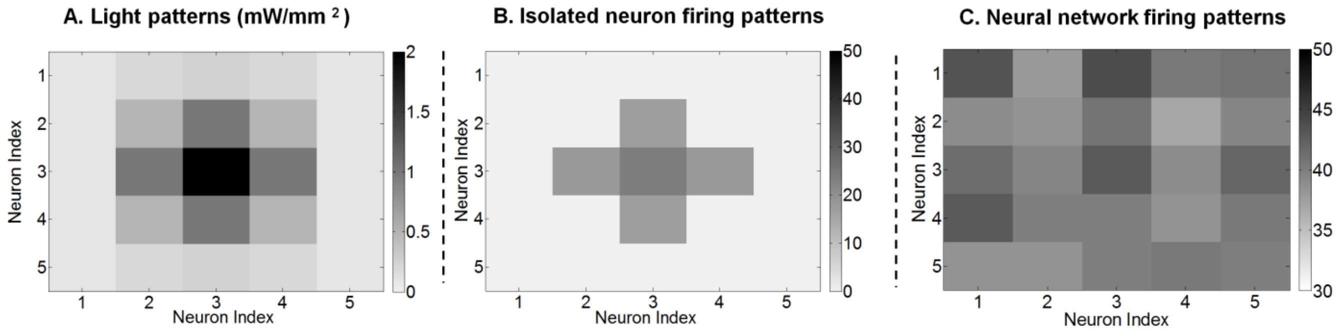


Fig. 13: Real-time simulation of an optogenetically transfected network of spiking neurons. A shows the light irradiance pattern for a network; B shows the network firing patterns without synaptic connections; C show the network firing patterns with synaptic connections;

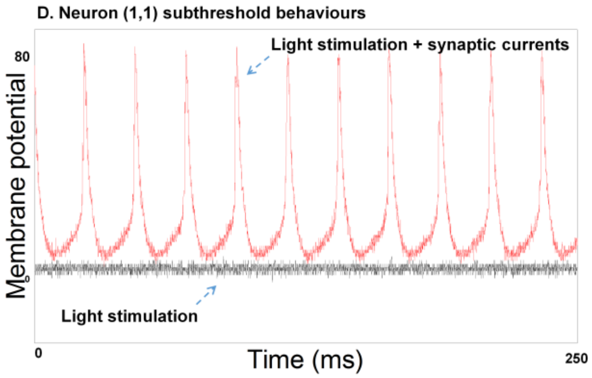


Fig. 14: An example of an opto-neuron sub-threshold firing behavours.

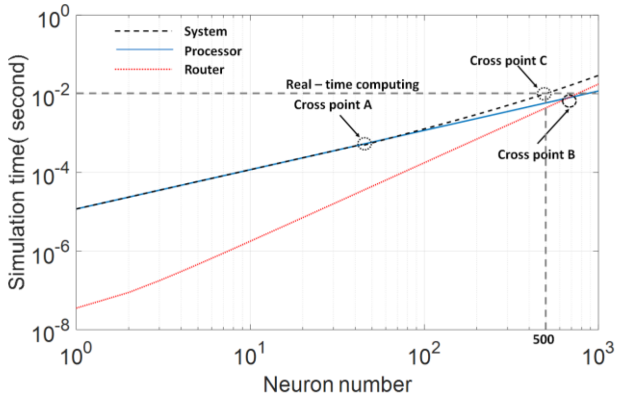


Fig. 15: The speed performance of the neural processor implementing different network sizes. At cross point A, the network performance shows non-linear behaviour rather than linear behaviour. At cross point B, the individual router processing periods will be longer than the processor's. At cross point C, the maximum neuron number that can be implemented on the processor for real-time computing is found to be 500 and takes 9.7ms (assuming fastest biologically-realistic firing frequency is 100Hz).

optogenetics as a method, such as how to translate therapeutic or scientific network stimulation into a particular three dimensional light pattern. We hope that the developed processor will prove to be a reliable tool with which to address those challenges.

D. *Future work*

One of the main areas for further development will be in developing new techniques for system optimization. For example, the natural communication in biological systems tends to be asynchronous and event driven. Therefore, an asynchronous communication protocol [40] coupled with an event driven approach [41] may potentially make the system more power efficient. Furthermore, sharing the common computing-path [42] (*e.g.* ALU1) and optimization of the neural network modularity [43][44] will result in utilizing less hardware resources. Finally, multi-core architectures [45] represent a promising way to scale the number of implemented neurons towards brain-scale sizes with real-time computation.

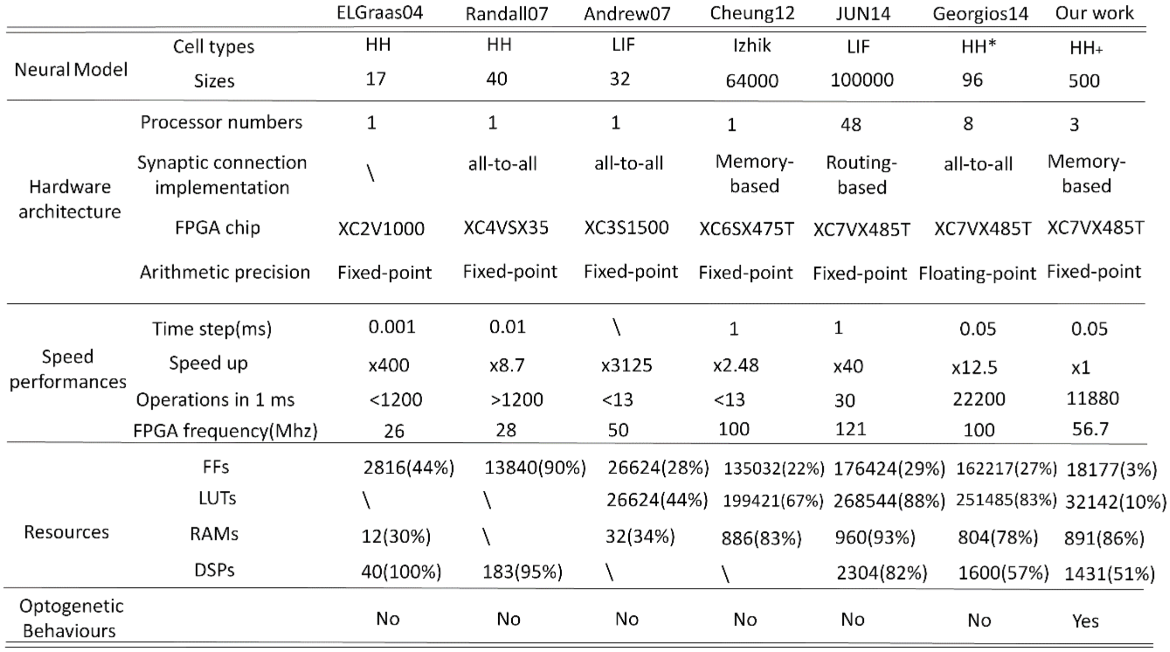
VI. Conclusion

In this work we have designed and implemented an FPGA based neural processor for real-time simulation of opto-neural behaviour. The developed neural processor can successfully reproduce the photo-kinetics of mammalian neurons expressing optically active ion channels [7] in a biologically realistic neural network model. It only requires 0.03ms for a single neuron and 9.7ms for a fully connected 500-neuron network to generate a spike. Therefore the system, with its real-time computing performance and highly biologically-realistic behaviour, can be applied in many ways as a powerful tool for multidisciplinary researchers in the field of optogenetics.

Acknowledgements

We would like to thank the Engineering Physical Sciences Research Council (eFutures program and Grant EP/N002474/1), Biotechnology and Biological Sciences Research Council (Grant BB/L018268/1) and Newcastle University for supporting this research.

Table 4: The summary of FPGA based neural modeling



		ELGraas04	Randall07	Andrew07	Cheung12	JUN14	Georgios14	Our work
Neural Model	Cell types	HH	HH	LIF	Izhik	LIF	HH*	HH+
	Sizes	17	40	32	64000	100000	96	500
Hardware architecture	Processor numbers	1	1	1	1	48	8	3
	Synaptic connection implementation	\	all-to-all	all-to-all	Memory-based	Routing-based	all-to-all	Memory-based
	FPGA chip	XC2V1000	XC4VSX35	XC3S1500	XC6SX475T	XC7VX485T	XC7VX485T	XC7VX485T
	Arithmetic precision	Fixed-point	Fixed-point	Fixed-point	Fixed-point	Fixed-point	Floating-point	Fixed-point
Speed performances	Time step(ms)	0.001	0.01	\	1	1	0.05	0.05
	Speed up	x400	x8.7	x3125	x2.48	x40	x12.5	x1
	Operations in 1 ms	<1200	>1200	<13	<13	30	22200	11880
	FPGA frequency(Mhz)	26	28	50	100	121	100	56.7
Resources	FFs	2816(44%)	13840(90%)	26624(28%)	135032(22%)	176424(29%)	162217(27%)	18177(3%)
	LUTs	\	\	26624(44%)	199421(67%)	268544(88%)	251485(83%)	32142(10%)
	RAMs	12(30%)	\	32(34%)	886(83%)	960(93%)	804(78%)	891(86%)
	DSPs	40(100%)	183(95%)	\	\	2304(82%)	1600(57%)	1431(51%)
Optogenetic Behaviours		No	No	No	No	No	No	Yes

Table 5: The summary of developed processor applications

Applications	Hardware requirements
Optogenetic actuators investigation	Updating parameters and configuration signals for ChR2 computational blocks
Verification of optical – neural interfaces	Updating all parameters in data generation system and configuration signals in control blocks
Multi-site light stimulation strategies investigation	Updating all parameters in light stimulation profiles

REFERENCES

[1] G. Nagel, T. Szellas, W. Huhn, S. Kateriya, N. Adeishvili, P. Berthold, D. Ollig, P. Hegemann, and E. Bamberg, "Channelrhodopsin-2, a directly light-gated cation-selective membrane channel.," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 100, no. 24, pp. 13940–5, Nov. 2003.

[2] H. Barbier-Brygoo, M. Vinauger, J. Colcombet, G. Ephritikhine, J.-M. Frachisse, and C. Maurel, "Anion channels in higher plants: functional characterization, molecular structure and physiological role," *Biochim. Biophys. Acta - Biomembr.*, vol. 1465, no. 1–2, pp. 199–218, May 2000.

[3] K. Deisseroth, "Optogenetics.," *Nat. Methods*, vol. 8, no. 1, pp. 26–9, Jan. 2011.

[4] J. T. Paz and J. R. Huguenard, "Optogenetics and epilepsy: past, present and future.," *Epilepsy Curr.*, vol. 15, no. 1, pp. 34–8, Jan. .

[5] E. Krook-Magnuson, C. Armstrong, M. Oijala, and I. Soltesz, "On-demand optogenetic control of spontaneous seizures in temporal lobe epilepsy.," *Nat. Commun.*, vol. 4, p. 1376, Jan. 2013.

[6] J. M. Barrett, R. Berlinguer-Palmini, and P. Degenaar, "Optogenetic approaches to retinal prosthesis.," *Vis. Neurosci.*, vol. 31, no. 4–5, pp. 345–54, Sep. 2014.

[7] K. Nikolic, N. Grossman, M. S. Grubb, J. Burrone, C. Toumazou, and P. Degenaar, "Photocycles of channelrhodopsin-2.," *Photochem. Photobiol.*, vol. 85, no. 1, pp. 400–11, Jan. .

[8] N. Grossman, K. Nikolic, C. Toumazou, and P. Degenaar, "Modeling study of the light stimulation of a neuron cell with channelrhodopsin-2 mutants.," *IEEE Trans. Biomed. Eng.*, vol. 58, no. 6, pp. 1742–51, Jun. 2011.

[9] L. . Abbott, "Lapicque's introduction of the integrate-and-fire model neuron (1907)," *Brain Res. Bull.*, vol. 50, no. 5–6, pp. 303–304, Nov. 1999.

[10] E. M. Izhikevich, "Simple model of spiking neurons.," *IEEE Trans. Neural Netw.*, vol. 14, no. 6, pp. 1569–72, Jan. 2003.

[11] J. L. Hindmarsh and R. M. Rose, "A model of neuronal bursting using three coupled first order differential equations.," *Proc. R. Soc. Lond. B. Biol. Sci.*, vol. 221, no. 1222, pp. 87–102, Mar. 1984.

[12] A. L. HODGKIN and A. F. HUXLEY, "A quantitative description of membrane current and its application to conduction and excitation in nerve.," *J. Physiol.*, vol. 117, no. 4, pp. 500–44, Aug. 1952.

[13] T. Bruegmann, D. Malan, M. Hesse, T. Beiert, C. J. Fuegemann, B. K. Fleischmann, and P. Sasse, "Optogenetic control of heart muscle in vitro and in vivo.," *Nat. Methods*, vol. 7, no. 11, pp. 897–900, Nov. 2010.

[14] A. Gerits, R. Farivar, B. R. Rosen, L. L. Wald, E. S. Boyden, and W. Vanduffel, "Optogenetically induced behavioral and functional network changes in primates.," *Curr. Biol.*, vol. 22, no. 18, pp. 1722–6, Sep. 2012.

[15] A. K. Fidjeland and M. P. Shanahan, "Accelerated simulation of spiking neural networks using GPUs," in *The 2010 International Joint Conference on Neural Networks (IJCNN)*, 2010, pp. 1–8.

[16] M. Wang, B. Yan, J. Hu, and P. Li, "Simulation of large neuronal networks with biophysically accurate models on graphics processors," in *The 2011 International Joint Conference on Neural Networks*, 2011, pp. 3184–3193.

[17] G. Smaragdos, S. Isaza, M. F. van Eijk, I. Sourdis, and C. Strydis, "FPGA-based biophysically-meaningful modeling of olivocerebellar neurons," in *Proceedings of the 2014 ACM/SIGDA international symposium on Field-programmable gate arrays - FPGA '14*, 2014, pp. 89–98.

[18] R. K. Weinstein, M. S. Reid, and R. H. Lee, "Methodology and design flow for assisted neural-model implementations in FPGAs.,"

IEEE Trans. Neural Syst. Rehabil. Eng., vol. 15, no. 1, pp. 83–93, Mar. 2007.

[19] E. L. Graas, E. A. Brown, and R. H. Lee, “An FPGA-based approach to high-speed simulation of conductance-based neuron models.,” *Neuroinformatics*, vol. 2, no. 4, pp. 417–36, Jan. 2004.

[20] R. D. Traub, R. K. Wong, R. Miles, and H. Michelson, “A model of a CA3 hippocampal pyramidal neuron incorporating voltage-clamp data on intrinsic conductances.,” *J. Neurophysiol.*, vol. 66, no. 2, pp. 635–50, Aug. 1991.

[21] C. Soto-Treviño, P. Rabbah, E. Marder, and F. Nadim, “Computational model of electrically coupled, intrinsically distinct pacemaker neurons.,” *J. Neurophysiol.*, vol. 94, no. 1, pp. 590–604, Jul. 2005.

[22] L. Berdondini, K. Imfeld, A. Maccione, M. Tedesco, S. Neukom, M. Koudelka-Hep, and S. Martinoia, “Active pixel sensor array for high spatio-temporal resolution electrophysiological recordings from single cell to large scale neuronal networks.,” *Lab Chip*, vol. 9, no. 18, pp. 2644–51, Sep. 2009.

[23] J. Wang, F. Wagner, D. A. Borton, J. Zhang, I. Ozden, R. D. Burwell, A. V Nurmikko, R. van Wagenen, I. Diester, and K. Deisseroth, “Integrated device for combined optical neuromodulation and electrical recording for chronic in vivo applications.,” *J. Neural Eng.*, vol. 9, no. 1, p. 016001, Feb. 2012.

[24] B. McGovern, R. B. Palmini, N. Grossman, E. Drakakis, V. Poher, M. A. A. Neil, and P. Degenaar, “A New Individually Addressable Micro-LED Array for Photogenetic Neural Stimulation.,” *IEEE Trans. Biomed. Circuits Syst.*, vol. 4, no. 6, pp. 469–76, Dec. 2010.

[25] H. D. I. Abarbanel, L. Gibb, R. Huerta, and M. I. Rabinovich, “Biophysical model of synaptic plasticity dynamics.,” *Biol. Cybern.*, vol. 89, no. 3, pp. 214–26, Sep. 2003.

[26] “Core Conductor Theory and Cable Properties of Neurons - Comprehensive Physiology.” [Online]. Available: http://www.comprehensivephysiology.com/WileyCDA/CompPhysArticle/refId-cp010103.html. [Accessed: 05-Oct-2015].

[27] B. Clark and M. Häusser, “Neural coding: hybrid analog and digital signalling in axons.,” *Curr. Biol.*, vol. 16, no. 15, pp. R585–8, Aug. 2006.

[28] S. A. Weiss, T. Preuss, and D. S. Faber, “Phase encoding in the Mauthner system: implications in left-right sound source discrimination.,” *J. Neurosci.*, vol. 29, no. 11, pp. 3431–41, Mar. 2009.

[29] J. Mattis, K. M. Tye, E. A. Ferenczi, C. Ramakrishnan, D. J. O’Shea, R. Prakash, L. A. Gunaydin, M. Hyun, L. E. Fenno, V. Gradinaru, O. Yizhar, and K. Deisseroth, “Principles for applying optogenetic tools derived from direct comparative analysis of microbial opsins.,” *Nat. Methods*, vol. 9, no. 2, pp. 159–72, Feb. 2012.

[30] K. Cheung, S. R. Schultz, and W. Luk, *Artificial Neural Networks and Machine Learning – ICANN 2012*, vol. 7552. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

[31] A. Cassidy, S. Denham, P. Kanold, and A. Andreou, “FPGA Based Silicon Spiking Neural Array,” in *2007 IEEE Biomedical Circuits and Systems Conference*, 2007, pp. 75–78.

[32] S. B. Furber, D. R. Lester, L. A. Plana, J. D. Garside, E. Painkras, S. Temple, and A. D. Brown, “Overview of the SpiNNaker System Architecture,” *IEEE Trans. Comput.*, vol. 62, no. 12, pp. 2454–2467, Dec. 2013.

[33] J. Luo, G. Coapes, T. Mak, T. Yamazaki, C. Tin, and P. Degenaar, “A scalable FPGA-based cerebellum for passage-of-time representation.,” *Conf. Proc. ... Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. IEEE Eng. Med. Biol. Soc. Annu. Conf.*, vol. 2014, pp. 3102–5, Jan. 2014.

[34] J. Luo, G. Coapes, T. Mak, T. Yamazaki, C. Tin, and P. Degenaar, “Real-Time Simulation of Passage-of-Time Encoding in Cerebellum Using a Scalable FPGA-Based System.,” *IEEE Trans. Biomed. Circuits Syst.*, Oct. 2015.

[35] T. Yamazaki and S. Tanaka, “A spiking network model for passage-of-time representation in the cerebellum.,” *Eur. J. Neurosci.*, vol. 26, no. 8, pp. 2279–2292, Oct. 2007.

[36] B. V. Benjamin, Peiran Gao, E. McQuinn, S. Choudhary, A. Chandrasekaran, J.-M. Bussat, R. Alvarez-Icaza, J. V. Arthur, P. . Merolla, and K. Boahen, “Neurogrid: A Mixed-Analog-Digital Multichip System for Large-Scale Neural Simulations,” *Proc. IEEE*, vol. 102, no. 5, pp. 699–716, May 2014.

[37] T. Yu, S. Joshi, C. Maier, and G. Cauwenberghs, “65k-neuron integrate-and-fire array transceiver with address-event reconfigurable synaptic routing,” in *2012 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, 2012, pp. 21–24.

[38] M. Neumüller and F. Jähnig, “Modeling of halorhodopsin and rhodopsin based on bacteriorhodopsin.,” *Proteins*, vol. 26, no. 2, pp. 146–56, Oct. 1996.

[39] H. Pinardo, “Archaerhodopsin Expression in New Hippocampal Neurons.” 01-Apr-2015.

[40] N. Imam, K. Wecker, J. Tse, R. Karmazin, and R. Manohar, “Neural spiking dynamics in asynchronous digital circuits,” in *The 2013 International Joint Conference on Neural Networks (IJCNN)*, 2013, pp. 1–8.

[41] D. Neil and S.-C. Liu, “Minitaur, an Event-Driven FPGA-Based Spiking Network Accelerator,” *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 22, no. 12, pp. 2621–2628, Dec. 2014.

[42] Q. Liu, G. A. Constantinides, K. Masselos, and P. Y. K. Cheung, “Data-reuse exploration under an on-chip memory constraint for low-power FPGA-based systems,” *IET Comput. Digit. Tech.*, vol. 3, no. 3, p. 235, 2009.

[43] A. Cassidy, A. G. Andreou, and J. Georgiou, “Design of a one million neuron single FPGA neuromorphic system for real-time multimodal scene analysis,” in *2011 45th Annual Conference on Information Sciences and Systems*, 2011, pp. 1–6.

[44] A. S. Cassidy, J. Georgiou, and A. G. Andreou, “Design of silicon brains in the nano-CMOS era: spiking neurons, learning synapses and neural architecture optimization.,” *Neural Netw.*, vol. 45, pp. 4–26, Sep. 2013.

[45] P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, Y. Nakamura, B. Brezzo, I. Vo, S. K. Esser, R. Appuswamy, B. Taba, A. Amir, M. D. Flickner, W. P. Risk, R. Manohar, and D. S. Modha, “A million spiking-neuron integrated circuit with a scalable communication network and interface,” *Science (80-.).*, vol. 345, no. 6197, pp. 668–673, Aug. 2014.