https://doi.org/10.1038/s44335-026-00066-4

Advanced neuronal logic circuit designs using spiking models: a framework for sequential biocomputation

Giulio Basso[1,2], Reinhold Scherer[3] & Michael Taynnan Barros[3] ✉

With conventional silicon-based computing approaching practical limits, biocomputing is being explored as a promising complement. Neuronal biocomputing is investigated for potential gains in energy efficiency, on-chip learning, and integration with biological systems. We explore logic gates and sequential circuits in spiking neuronal models that mimic motifs from conventional computer architectures. We propose a design framework that combines biophysically inspired spiking models, optimisation, and simulation for neuronal logic circuits. We demonstrate, in silico, NAND gates, SR latches, and D flip-flops implemented with spiking neurons. We configure synaptic conductances, introduce neuronal buffers for synchronisation, and specify network topologies. We encode binary information in spiking patterns and mitigate synchronisation issues using neuronal buffers and inhibitory control. Our results indicate effective and scalable neuronal logic circuits and, showing that they maintain a stable metabolic burden even in complex data storage configurations. Overall, this work provides a reproducible basis for logic and storage in spiking networks and lays the groundwork for future biological and neuromorphic implementations.

Conventional silicon-based computer chips are facing practical limits. Processing speed and circuit complexity are limited by factors such as power consumption, heat dissipation, the degree of miniaturisation, and the manufacturing processes and materials used[1]. To meet our growing computing needs, research is focusing on improving performance through alternative materials, superconductivity, and, more recently, a shift toward resource-efficient solutions like biological computing (biocomputing).

Deoxyribonucleic acid (DNA), fungi, bacteria, astrocytes, and neurons are examples of biomaterials used to deploy biocomputing[2]. The selection of appropriate computational biomaterial depends on the intended application[3]. Biomaterials types based on molecules or cells as machines are currently the most mature choices, with the latter being more reliable, faster, and resource-efficient[4]. For example, 3D cultured biological neurons networks already have demonstrated capabilities to recognise speech[5] or to interact with a digital computer system to play computer games[6,7]. Such complex functionality can only be achieved and maintained by adapting the information processing in cells, the cell-cell communications and the structure of the cellular network. Controlling these properties motivates computational frameworks for circuit design in neuronal substrates. We present an in silico spiking neural network framework that implements candidate logic gates and sequential memory elements through controlled spiking dynamics. While all results are in silico, the design principles are intended to inform future implementation in biological or bioengineered neural substrates and deployment on neuromorphic hardware.

Neuronal biocomputing models hold significant potential to serve as a viable complement to silicon-based computers[8,9], however a deeper understanding of how these models can be programmed or trained to perform complex, controllable computing tasks is needed[10,11], including digital data storage. While biological systems encode multilevel information, digital data processing is far removed from this characteristic creating new opportunities for rethinking it. Even though, reliability is a major challenge in any solution for data storage using biomaterials[12]. Moreover, one must overcome the challenge of building modular or scalable biocomputing solutions that are also reliable and realisable in-vitro. Biocomputing should draw inspiration from silicon-based digital systems, incorporating Boolean logic and advanced principles of circuit design and synthesis. This approach enables the combination of elementary logic gates into logic circuits capable of performing complex, controllable, and scalable calculations. A corresponding framework is not yet available in biocomputing[13]. In addition, the relationship between computational complexity and biological resource usage (e.g., metabolic burden, energy expenditure) remains under explored. Considering this, researchers would be able to understand the limits of biocomputing, its limitations and its applications towards improved reproducibility and reliability. We thus investigate whether spiking

[1]Department of Mechanical and Aerospace Engineering, Torino, Italy. [2]Department of Electrical Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands. [3]School of Computer Science and Electronic Engineering, University of Essex, Colchester, UK. ✉e-mail: m.barros@essex.ac.uk

neuronal circuits can reliably implement logic gates and digital signal storage using standard logic design principles, and we consider the implications such operations may have for biological resource use (e.g., metabolic burden) in future implementations.

One of the earliest demonstrations of logic gate behavior in biological neuronal systems was presented by Goldental et al.[14], who introduced a computational paradigm based on dynamic logic-gates embedded in cortical cultures. Their experimental framework showed that logic operations could be modulated in real-time using tailored stimulation protocols. Exploratory work has shown that neuronal models can compute outputs using simple logic systems based on AND and OR gates[13,15]. Achieving more complex computing circuits beyond cascaded solutions continues to be a major challenge since existing solutions often lack the necessary scalability and efficiency. Current approaches are limited in their ability to integrate multiple logic functions seamlessly. The state-of-the-art neuronal biocomputing circuit includes combinatorial circuits and cascaded circuits of AND and OR gating[16,17]. In this paper, we start with neuronal logic gates, and we explore the design of networks composed of these modules to implement synchronous sequential logic, which forms the foundation of silicon-based computer architectures.

We address the challenges posed by simple logic operations and the unscalability and unreliability of neuron-based biocomputing by focusing on the design and engineering of synchronized neuronal networks with specific topologies that implement advanced logic gates (NOT, AND NOT, NAND) in sequential circuits (SR latch, gated SR latch, flip-flop, and D flip-flop). We show how neurons can be used not only for binary data processing and storage by mimicking the same logic circuits used in traditional digital technology to store data. We use optimisation and digital-circuit principles to design sequential gates by configuring synaptic interactions and network topology. We specify synaptic configurations, network topology, and synchronisation via neuronal buffers to achieve digital-like information flow, processing, and storage. We successfully validated the operation of all gates and circuits in-silico. Additionally, we quantify the energetic footprint of these networks by coupling an energy-aware neuron model to spiking activity. This design framework offers a reproducible basis for composing logic and storage in spiking networks. Our framework supports the basis for powerful biocomputing technologies such as biological finite-state machines and more, giving resource-efficient computing a beyond-silicon new ally: neurons.

Results

For our experiments, we used validated models that account for spike events, the information carriers in neuron membranes, and synapses, which transmit signals between neurons. We chose Izhikevich neurons with conductance-based excitatory bipartite synapses and considered both excitatory and inhibitory neurons to regulate spike transmission. Binary information was encoded in spiking ("1") and non-spiking ("0") activities. Our results validated the logic structures through their Truth Tables, demonstrating correct input-output relationships. In-silico experiments, detailed in the Materials and Methods section, highlighted the precise engineering of neuron communication networks to achieve the desired gate behavior.

Neuronal AND NOT-based logic gates

The AND gate is the first neuronal logic gate, with its operational validation is shown in Fig. 1a–d. We fine-tuned the neuronal communication parameters at the synapses (neuron-neuron connection points) so that their synaptic weight, which is the strength of the synapse, is $w_z = 0.065$ for both synapses. The spike-raising phase has a time constant τ_r equal to 2% of the inter-spike interval (ISI), which is 2.64 ms, and the decaying phase has a time

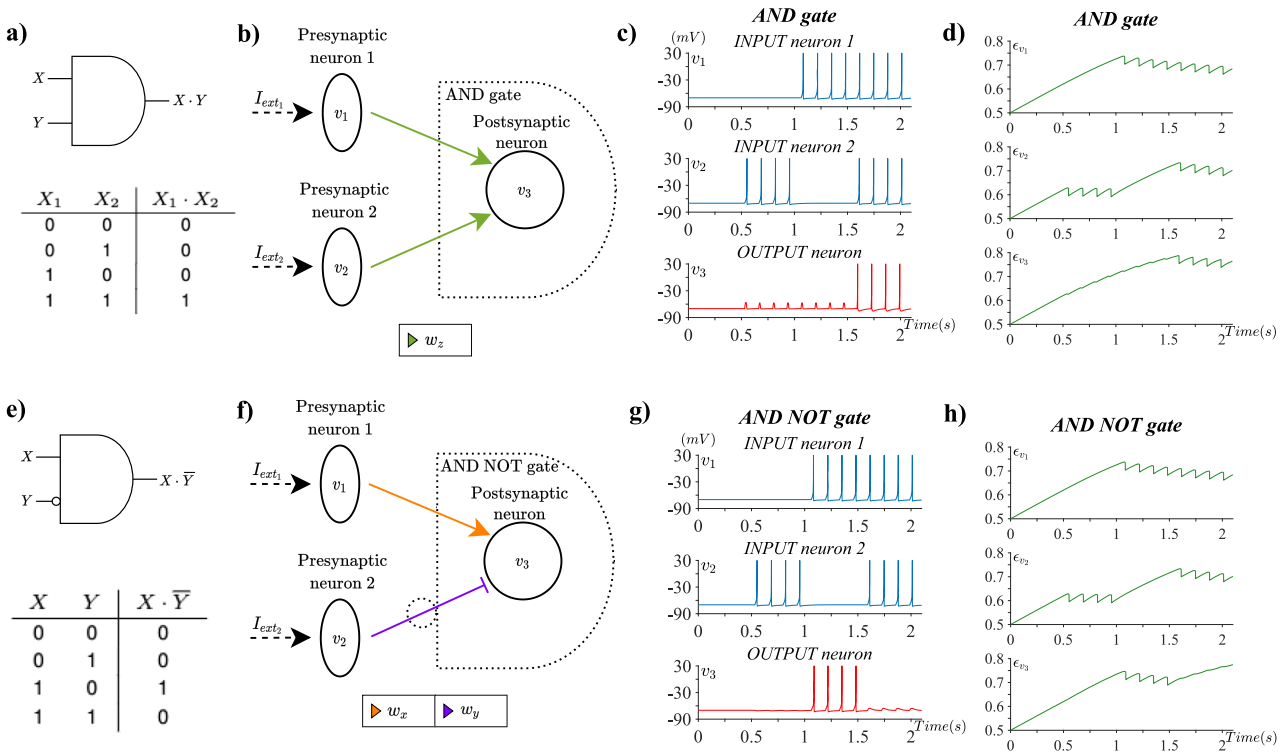


Fig. 1 | Gating response of the neuronal AND and AND NOT gates. a The logic gate formal symbolic representation of the AND and below its truth table (**b**) the functional graph of the neurons and connections, including types, composing the AND gate (**c**) the functional validation of the AND gate following the truth table order in (**a**). **d** The time-series metabolic burden per neuronal unit. **e** The logic gate formal symbolic representation of the AND NOT and below its truth table (**f**) the functional graph of the neurons and connections, including types, composing the AND NOT gate. **g** The functional validation of the AND NOT gate following the truth table order in (**e**). **h** The time-series metabolic burden per neuronal unit. For the excitatory synapse, the synaptic parameters are set as: $w_x = 0.06$, $\tau_r = 19.80$ ms and $\tau_d = 26.40$ ms. For the inhibitory synapse, they are chosen as: $w_y = 0.18$, $\tau_r = 19.80$ ms and $\tau_d = 59.40$ ms. The amplitude of the stimulating current is $I = 4$ pA (**a**–**h**).

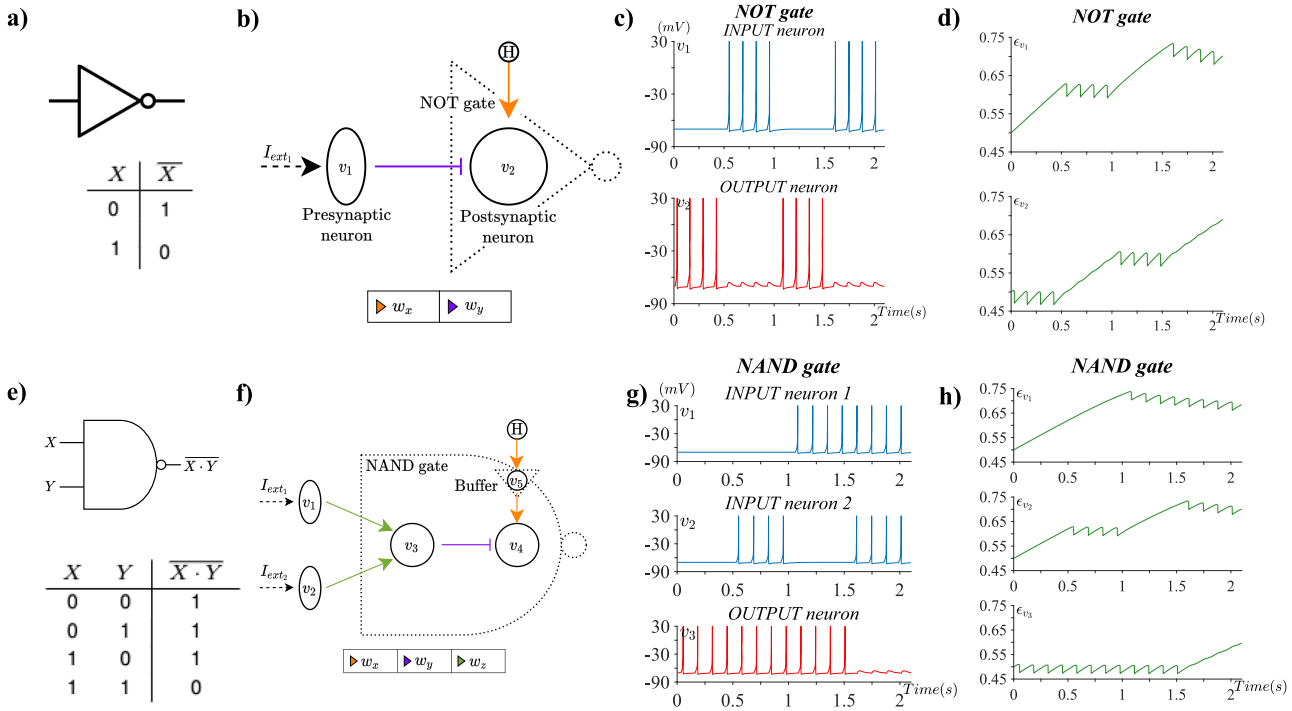


Fig. 2 | Gating response of the neuronal NOT and NAND gates. a The logic gate formal symbolic representation of the NOT and below its truth table. **b** The functional graph of the neurons and connections, including types, composing the NOT gate. **c** The functional validation of the NOT gate following the truth table order in (**a**). **d** The time-series metabolic burden per neuronal unit. **e** The logic gate formal symbolic representation of the NAND and below its truth table. **f** the functional graph of the neurons and connections, including types, composing the NAND gate. **g** The functional validation of the NAND gate following the truth table order in (**e**). **h** The time-series metabolic burden per neuronal unit. For the excitatory synapse, the synaptic parameters are set as: $w_x = 0.06$, $\tau_r = 19.80$ ms and $\tau_d = 26.40$ ms. For the inhibitory synapse, they are chosen as: $w_y = 0.18$, $\tau_r = 19.80$ ms and $\tau_d = 59.40$ ms. The amplitude of the stimulating current is $I = 4$ pA. Gating response of the neuronal NAND gate. The synaptic parameters of the AND gate are $w_z = 0.065$, $\tau_r = 2.64$ ms and $\tau_d = 3.96$ ms; for the NOT gate excitatory synapse $w_x = 0.06$, $\tau_r = 19.80$ ms and $\tau_d = 26.40$ ms; for the NOT gate inhibitory synapse $w_y = 0.18$, $\tau_r = 19.80$ ms and $\tau_d = 59.40$ ms. The amplitude of the stimulating current is $I = 4$ pA (**a**–**h**).

constant τ_d equal to 3% of the ISI, which is 3.96 ms. The input patterns are chosen according to the traditional incremental order used in truth tables, and the input levels are changed every half of a second. Fig. 1d shows the metabolic energy time-series analysis of all neuron units. We consider metabolic burden as the energy expenditure being the ratio of ATP/ADP (more details in our Materials and Methods section). We show that the energy expenditure has a fluctuation behaviour between 0.5 and 0.8 a.u. cost. The pattern of the fluctuations is visually associated with spike events, since the energy cost is dependent on the neuron membrane voltage. Energy expenditure increases during non-spike events due to a series of possible events, including neurotransmitter packaging and recycling[18], maintaining ion gradients[19,20], cytoskeletal dynamics, and protein synthesis[20].

The first newly designed neuronal logic gate developed is the AND NOT gate, Fig. 1a–d. For this gate, we need to consider both excitatory and inhibitory neuronal types, to account for its more complex Boolean logic. Its excitatory and inhibitory synaptic weights are set as $w_x = 0.06$ and $w_y = 0.18$ respectively. The synaptic time constant values are chosen as $\tau_r = 15\%ISI = 19.80$ ms and $\tau_d = 20\%ISI = 26.40$ ms for the excitatory synapse, and $\tau_r = 15\%ISI = 19.80$ ms and $\tau_d = 45\%ISI = 59.40$ ms for the inhibitory synapse. Fig. 1h shows the energy temporal analysis of all neuron units for the AND NOT gate, also showing fluctuation behaviour between 0.5–0.8 a.u. cost for similar reasons as the AND gate.

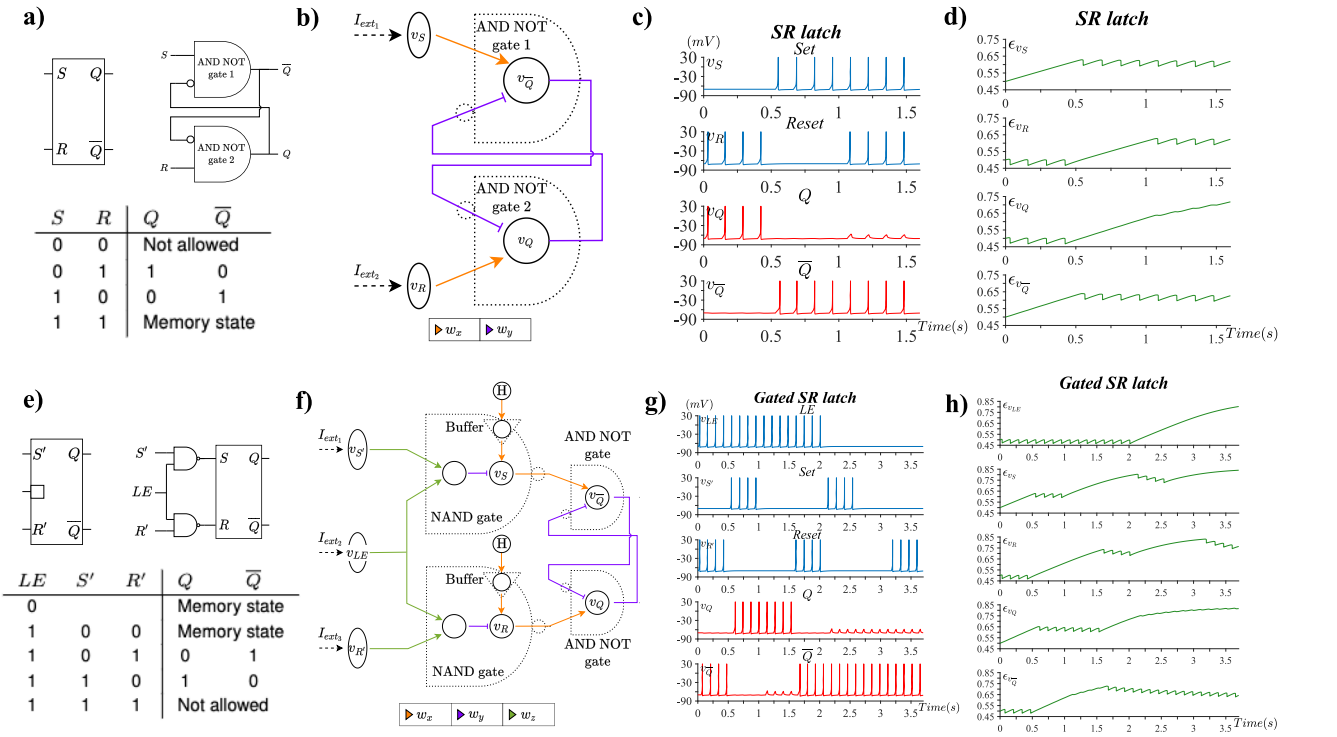
Afterwards, the neuronal NOT gate is presented, Fig. 2a–e. The neuronal NOT gate consists of a special case of the AND NOT gate; hence, the synaptic parameters w_x, w_y, τ_r, and τ_d are chosen as in the case of the AND NOT gate. The input pattern is generated as a sequence that toggles between the two states every 0.5 seconds. Here, to account for the ability to switch the level [1] based on the input we add an excitatory input (H node in Fig. 2b) that is always at the high level, namely a neuron that is always firing. We thus obtained fluctuations at the outputs at the desired [0] level (Fig. 2c), which did not lead to spike events, and thus the operation validity of the NOT gate remained correct. In addition, the metabolic energy cost is similar to previous cases.

Lastly, Fig. 2e–h shows how the neuronal NAND gate changes over time when the above-mentioned incremental input pattern is applied. The AND gate included in the NAND cascaded architecture makes use of the following synaptic parameters of the AND and NOT gates reported before. The NOT gate within the NAND architecture makes use of the same synaptic parameters as the AND NOT gate. Similarly, the neuronal buffers are realised using the synaptic parameters of the excitatory synapse of the AND NOT gate. While the association of the energy cost with spike events still stands, the fluctuations tend to follow the different stability points based on the gate outputs when compared to previously explored AND and NOT gates.

Neuronal latches and D flip-flop

Figure 3a–d shows the neuronal SR latch, where each membrane potential corresponds to digital logic signals. The set signal is represented by v_s, and the outputs are v_Q and $v_{\overline{Q}}$. The latch avoids the condition where both set and reset neurons are at rest, as this is not allowed. For example, when $S = 1$ and $R = 0$, Q becomes 0, and $\overline{Q}$ becomes 1. Similarly, when $S = 0$ and $R = 1$, Q is 1, and $\overline{Q}$ is 0. If both S and R are 1, the latch enters a memory state, storing its previous outputs. However, when $S = 0$ and $R = 0$, both Q and $\overline{Q}$ are 0, which is a forbidden condition, as it breaks the fundamental relationship between Q and $\overline{Q}$.

Most digital flip-flops are based on a fundamental primary-secondary architecture, as shown in Figure 4, which consists of two gated SR latches in cascade. The primary transmits input values, while the secondary holds or samples these outputs. A shared clock signal (*CLK*) controls both, with the secondary receiving it directly and the primary receiving an inverted signal.



S	R	Q	$\overline{Q}$
0	0	Not allowed	
0	1	1	0
1	0	0	1
1	1	Memory state	

LE	S′	R′	Q	$\overline{Q}$
0			Memory state	
1	0	0	Memory state	
1	0	1	0	1
1	1	0	1	0
1	1	1	Not allowed	

Fig. 3 | Gating response of the neuronal SR latch and GATED SR latch, with stimulating current $I = 4$ pA. a The logic gate formal symbolic representation of the SR latch and below its truth table. **b** The functional graph of the neurons and connections, including types, composing the SR latch. **c** The functional validation of the SR latch following the truth table order in (**a**). **d** The time-series metabolic burden per neuronal unit. **e** The logic gate formal symbolic representation of the GATED SR latch and below its truth table. **f** The functional graph of the neurons and connections, including types, composing the GATED SR latch. **g** The functional validation of the GATED SR latch following the truth table order in (**e**). **h** the time-series metabolic burden per neuronal unit.

When $CLK = 0$, the primary is active and responds to input changes, while the secondary remains in memory mode. When $CLK = 1$, the primary locks, and the secondary becomes transparent, setting its outputs to the primary's fixed values. This configuration represents a positive-edge flip-flop, sensitive to the rising clock edge. Figure 4 shows membrane potentials for the neuronal D flip-flop, where the input pattern corresponds to the membrane potential v_D. This device, triggered by the clock signal (v_{CLK}), shows rising clock edges with arrows. Testing with increased stimulation current ($I = 7$ pA) recalculated synaptic weights for AND NOT gates as $w_x = 0.11$ and $w_y = 0.67$, and for AND gates as $w_z = 0.099$. Energy costs fluctuate between 0.5 and 0.8, depending on voltage and network complexity in each circuit.

Discussion

Our novel usage of combinations of inhibition and excitation offers an effective tool for building more complex circuits, as it allows us to expand our library of gating responses to include AND NOT, NOT, and NAND gates. The realisation of each gate involved different choices regarding the synaptic model parameters. For the AND NOT gate, we can observe that the value of the weights related to the inhibitory synapse ($w_y = 0.18$) is higher than that of the excitatory synapse ($w_x = 0.06$). The reason for this could be related to the fact that when excitatory and inhibitory stimuli occur nearby, the inhibition must overcome the excitation, resulting in the resting state. Specifically for the AND NOT gate, the decay time constant of the inhibitory synapse ($\tau_d = 45\%ISI = 59.40$ ms) is chosen with greater respect the one of the excitatory synapse ($\tau_d = 20\%ISI = 26.40$ ms), because we addressed to the issues related to the mismatched synchronization by relaxing the dynamic of the inhibition. Furthermore, it can be noticed that the synaptic time constants related to the AND gate ($\tau_r = 2\%ISI = 2.64$ ms and $\tau_d = 3\%ISI = 3.96$ ms) are significantly smaller than in the case of the AND NOT gate. Nonetheless, these values fall within physiological ranges reported in the literature. For instance, the fastest AMPA receptors display a time constant of 0.18 ms, while the NMDA receptor of pyramidal cells is up to 89 ms[21].

Overall, Izhikevich neurons are well-known for accurately capturing the dynamics of membrane potential and the recovery variable, which represents the activation of ion channels in the cell membrane[22]. Specifically, the Izhikevich model can reproduce 20 of the most prominent features of biological neurons, including tonic spiking, intrinsic bursting, and chattering[23]. This contrasts with simpler models such as the leaky integrate-and-fire (LIF) model, which can only reproduce basic spiking. In our simulations, we have operated within the tonic spiking regime, which is characteristic of midbrain dopaminergic neurons[24], thalamic relay neurons[25], and striatal spiny projection neurons[26]. Our selection of a reduced-variable model enables greater transparency in mapping logic operations to specific network configurations, reducing the parameter space and improving design clarity. This approach facilitates reproducibility and interpretability when constructing elementary logic gates and memory elements in spiking neuronal systems. While our model abstracts dendritic morphology and conduction delays, these features are not essential to the primitives investigated here, which depend primarily on spike timing and connectivity. The exploration of how such biophysical and spatial factors shape biocomputation, enabling the transition from idealized logic gates to more robust, can lead to further biologically grounded architectures. Our designs are meant to run in microstructured, MEA-interfaced neuron-on-a-chip systems where connectivity is defined geometrically and stimulation is externally timed. In this context, synaptic dynamics are *managed by design*. We use a short, supervised calibration phase (if needed) to nudge weight ratios and pathway alignments into the required operating region; plasticity is then *disabled/held* and the circuits run clocked, with fixed parameters. During the run phase we do not rely on ongoing plasticity. This "calibrate-then-hold" workflow matches chip-based demonstrations in which logic behavior is achieved without connectivity changes during operation.

The versatility of our sequential logic circuits is demonstrated by their ability to scale easily, as larger circuits can be constructed by building on pre-trained elementary logic gates. Therefore, synaptic parameters can be

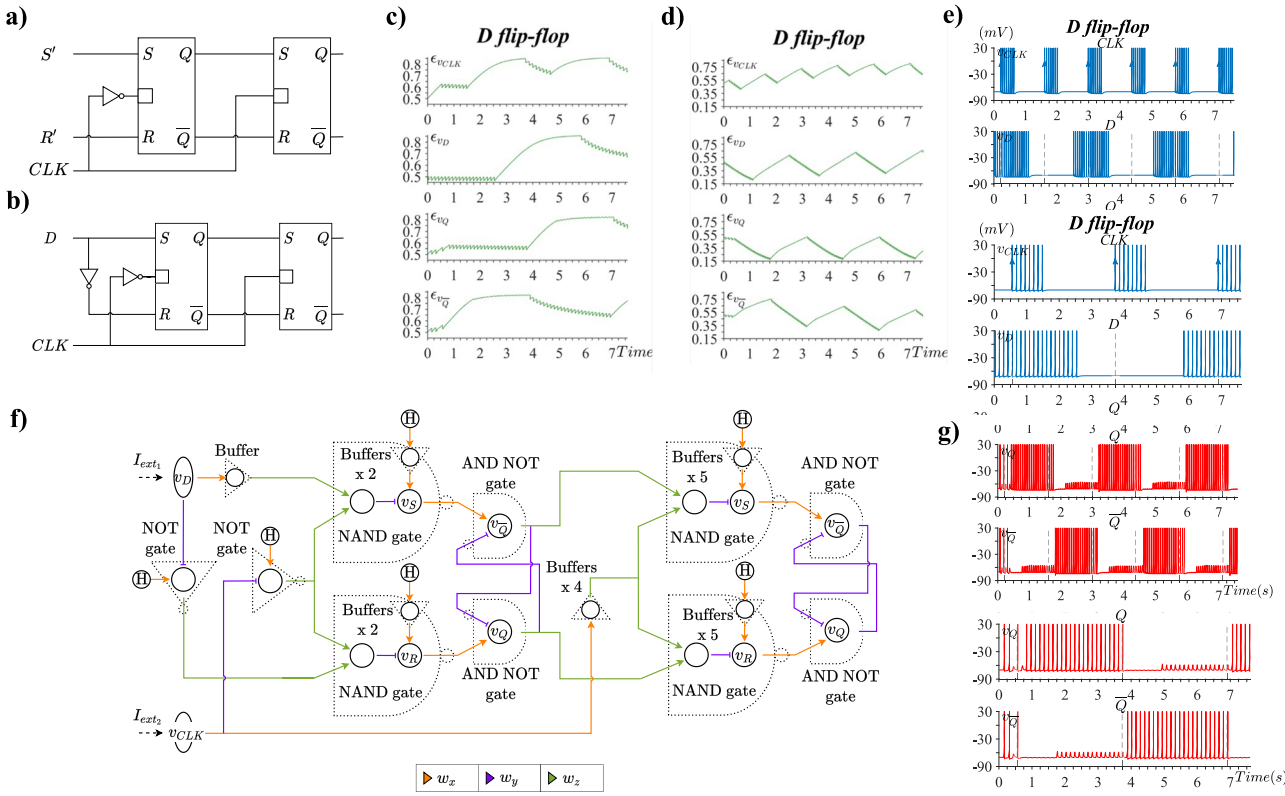


Fig. 4 | Gating response of the neuronal flip-flop and D flip-flop with stimulating current $I = 4$ pA and $I = 7$ pA. a Symbolic representation of the flip-flop. **b** The symbolic representation of the D flip-flop. **c** The time series burden of the D flip-flop inputs and output neuronal units with $I = 4$ pA. **d** The time series burden of the D flip-flop inputs and output neuronal units with $I = 7$ pA. **e** The functional validation of the D flip-flop inputs D and CLK. **f** The functional graph of the neurons and connections, including types, composing the D flip-flop and (**g**) the functional validation of the outputs of the D flip-flop, Q and $\overline{Q}$.

transferred in a modular fashion. This allows scaling beyond flip-flops, interconnecting modules to form larger sequential circuits. However, scaling increases the number of design variables to manage, including latency profiles, buffers, and clock configuration. Variations in excitatory and inhibitory inputs across different layers, especially in complex neuron network structures, cause an increase in spike latency, leading to synchronisation issues within the circuit. All designs are sensitive to such effects, desynchronisation can yield logic errors or operational failure. For such reasons we introduced *Neuronal Buffers* in our designs. With the exception of AND and AND NOT gates, buffers were added to excitatory neurons to synchronise misaligned excitatory and inhibitory output spikes. Advancing scalability will require addressing variability in network properties, quantifying latency, and establishing formal buffering guidelines. Overcoming these obstacles will unlock innovative approaches to biocomputing scalability, potentially incorporating our framework as a core design methodology. In addition, to capture the essence of clock-based solutions, our D-flip-flop solution includes a clock input, which is yet to be fully validated experimentally. Sequential operation here uses a periodic readout window to sample spikes, and the logic remains spike-driven. In vitro, this timing can be generated by small pacemaker/oscillator microcircuits or by patterned electrical/optical stimuli on MEAs. The same buffer-based path-depth rule we apply to data paths also governs the distribution of the timing signal, limiting skew so that sampling occurs when synaptic drive is stable. Advancing scalability will require addressing variability in network properties, quantifying latency, and establishing formal buffering guidelines. Clocks are essential in digital systems; analogous timing signals are likely to be important for coordinating information flow in biological implementations. Our results support scalability by construction: the same parameterized gate primitives (AND, AND NOT, NOT) can be composed into latches and a positive-edge-triggered D flip-flop, with neuronal buffers providing a systematic abstration for timing alignment. Empirically, energy proxies remained bounded (approx. 0.5–0.8 a.u.) as circuit complexity increased, suggesting that metabolic cost does not diverge under clocked operation. The primary scalability limits are therefore temporal than fucntional—arising from accumulated synaptic/neuronal delay, clock skew, and excitatory-inhibitory misalignment. We mitigate these with (i) a layer-matching rule and strategic buffer insertion, (ii) longer inhibitory decay constants to relax coincidence constraints, and (iii) a synchronous, edge-triggered regime. In practice, a neuronal "standard cell" library with characterized weights and time constants, togehter with hierarchical clock distribution (provided by a neuronal oscillator or patterned stimulation), and staged buffering/fan-out trees, should support growth from individual flip-flops to multi-bit registers and finite-state machines. A natural next step towards larger assemblies is to quantify propagation delay, jitter, and energy per toggle across parameter corners to establish robustness envelopes. Because absolute energy depends on the chosen biological ans hardware substrate, we report event counts and latencies instead, enabling reproducible cross-platform estimation. Correct operation of the proposed gates does not depend on finely tuned parameter values but on simple qualitative relationships that hold over a range of settings. For the AND gate, single presynaptic inputs are configured to remain subthreshold, whereas coincident inputs reliably drive the postsynaptic neuron to spike. For the AND-NOT and NOT gates, inhibition is set to dominate whenever it overlaps with excitation, while isolated excitatory drive remains sufficient to elicit a spike; this is ensured in practice by choosing inhibitory synapses to be clearly stronger and longer-lasting than excitatory ones. Because spike times can drift across layers, we insert neuronal buffers to keep input paths of comparable depth, which limits relative misalignment and preserves the intended coincidence or veto relations. The same principles extend to sequential elements: clean distribution of a timing signal and allowance for propagation through the constituent gates are sufficient for stable edge-triggered operation. Empirically, we observe that as long as these qualitative

conditions are maintained—strong-plus-longer inhibition during overlap, two-input excitation outweighing single-input drive, and buffered path balancing—the circuits retain their truth-table behavior without requiring delicate, case-by-case tuning.

Experimental biocomputing has achieved remarkable advancements through the years, however, based on the results of this paper, experimental biocomputing need to show how scalable and reliable they are, not only for neurons but for all other types of biocomputing systems. Current approaches have several drawbacks, including a lack of scalable designs, a lack of packaged solutions, and an over-reliance on complex predictive tools to manage high-intensity information flow and operational accuracy[2]. Potential experimental routes include neuron-on-a-chip platforms. This technology offers the ability to isolate neurons, control synaptic connections, and interface with stimulation and recording tools. Topology-control mechanisms on such platforms may enable information-flow control. We demonstrate, in silico, that sequential logic circuits can be implemented with spiking neurons, supporting digital-style processing and storage. Using sequential logic circuits as building blocks, cascading modules can create larger computing systems. Neuron-on-a-chip systems could provide a platform to test these designs experimentally. The ability to also incorporate storage can take biocomputing using neurons from mere function towards much larger, complex, scalable, and modular computing. Realising these elements experimentally would be a significant step toward practical biological computing, with potential applications in adaptive control[27], drug testing[9], and interactive systems[6]. The circuits demonstrated here require precise control over synaptic connectivity and temporal dynamics, which are not currently achievable with spontaneous or passively recorded neural activity. However, recent advances in neural engineering have enabled in-vitro implementation of Boolean logic gates using neurons cultured in polydimethylsiloxane (PDMS) microstructures combined with high-density microelectrode arrays (HD-MEAs), providing fine-grained spatial and temporal stimulation control[28]. These developments offer a promising platform for translating our model into experimental settings. We also acknowledge the possibility of neuromorphic hardware implementations. On these platforms the same contract—event-driven synapses with rise/decay, a configurable threshold and sampling window, buffer primitives, bounded fan-in/out, and a low-skew timing net—suffices to reproduce the demonstrated truth tables. Recent neuromorphic efforts span fixed-topology event fabrics, programmable neuron/synapse blocks, and FPGA-based prototypes that emphasize throughput/energy measurement and tool support[29–36]. Those works typically report performance of learning or inference workloads, but they do not specify neuron-biophysical recipes for synchronous sequential logic. Our results fill that gap as we give gate-level primitives, skew control via neuronal buffers, and edge-triggered storage, all of which can be compiled to such platforms or realized in wetware. Our designs signal an exciting and illuminating journey into scalable and realisable neuron-based biocomputing sequential circuits. This paves the way for integrating human-designed intelligence into living biological machines, opening up a future where the fusion of technology and biology redefines the boundaries of computing.

Methods

In-silico experiments

In-silico experiments depicted in our Results section were computed using MATLAB R2021b Simulink (MathWorks, Natick, MA, USA). The equations are discretized using the explicit Euler method with step size dt = 0.5 ms. The Simulink parameter sample time, which indicates when a Simulink block produces outputs and updates its internal state, is chosen to be equal to dt. The Izhikevich neurons involved in the model are characterised by tonic spiking behaviour,[37]. The reversal potential E_{syn} is set as 0 for excitatory synapses and as − 75 for inhibitory synapses. All the logic circuits are driven by an external stimulating current equal to I = 4 pA. Finally, an example of logic circuit with higher stimulating current I = 7 pA is also provided.

Engineering Neuron Communications

One of the key aspects of the design of neuronal logic gates is the possibility of cascading, that is, connecting more of them using the gates output as the inputs of another consecutive gate. Hence, here we formalise the neuronal logic gates as being made only by the postsynaptic neuron and its synapses with the adjusted weights w_i. Notice that the presynaptic neurons are not part of the gate. The inputs of the logic gates are represented by the membrane potentials v_1 and v_2, while the external current I_{ext_i} are only used in the first layer to communicate the input pattern to the network. With such definition, these building blocks can be intuitively layered to build more complex circuits. For instance, the membrane potential v_3, can be used as one of the two inputs of another neuronal logic gate.

Neuronal AND NOT gate and NOT gate

The AND gate is the logic gate that implements the logical conjunction. That means that its output is at the high only if both of its inputs are at the high level, as indicated in the Truth Table in Fig. 1a). Similarly, a neuron that fires only if it concurrently receives two presynaptic stimuli implements the same boolean rule, and we define it as neuronal AND gate. Figure 1b) describes the functional scheme of the neuronal AND gate, which consists in 3 neurons. Two presynaptic neurons are connected to the same postsynaptic neuron through two distinct excitatory synapses. In all the functional schemes provided in this work, the following graphical conventions will be adopted. The dashed arrows indicate the external stimulating current I_{ext_i}, always chosen as step or rectangular functions. Neurons which cover only a presynaptic role (i.e., they do not form any synapses in which they represent the postsynaptic neuron) are depicted by ellipses, characterised by their membrane potential variable. The circles stand for the postsynaptic neurons. The equivalent traditional symbol of the logic gate is reported with dotted line.

The basic idea of the neuronal AND gate design is that we need to regulate the inputs' influence on the synapses so that the firing of just one presynaptic neuron is not enough to trigger the output response. Instead, the firing of both presynaptic neurons must produce a sufficiently strong stimulation such that it causes the firing of the postsynaptic neuron, thanks to the spatial summation mechanism. This could be done by regulating the synaptic weights w_i, because they represent the strength of the synapses, namely how much a presynaptic input affects its corresponding postsynaptic neuron. Since the AND gate acts symmetrically, meaning that the result of the input sequence [1 0] is the same of the sequence [0 1], the two synaptic weights w_i need to be equal. The value of w_i is here computed empirically between real values of 0 and 1, searching for the minimum value of w_i that makes the output neuron firing in the condition with both inputs high. We refer to the weight value of the AND gate's synapses obtained with such empirical procedure as w_z. In this sense, the OR gate can be viewed as a simple adaptation of the AND gate, using the same network topology but relaxing the synaptic integration requirement: whereas the AND gate requires coincident firing to exceed the postsynaptic threshold, the OR gate is satisfied by any single input reaching that threshold independently. The difference thus lies purely in the synaptic weight regime and threshold proximity, rather than in architectural complexity. The empirical tuning procedure thus consists of identifying the minimum value of w_i that allows the output neuron to fire when only one of the two presynaptic neurons is active. To avoid undesired firing in the absence of inputs, this weight must also respect the constraint $w_i < \theta$, where θ is the firing threshold of the postsynaptic neuron. We denote the resulting optimal synaptic weight for the OR gate as w_o.

In ref. 38 L. Yoder theorized a novel type of logic gates, particularly suitable for biocomputed implementations with neurons, that can be flexibly used to develop several computational solutions. As shown in Fig. 1, it consists in an AND gate in which one of its two inputs is inverted. L. Yoder referred to this special type of logic gate as *AND NOT gate*. Considering a traditional AND gate, its output is 1 if and only if both inputs are 1. Instead, for the statement X AND NOT Y, the high output requires X to be 1, while Y needs to be 0, because of the logical negation. In any other cases the output is

0, as depicted in the Truth table (Fig. 1a). Notice that, unlike traditional gates, as AND and OR gates, the AND NOT gate is not symmetrical in respect its inputs. L. Yoder proposed that the gating behaviour of the AND NOT gate could be considered similar to a neuron response based on the following modelling assumption. Lets consider a neuron with one excitatory synapse and one inhibitory synapse, along the following:

- the inhibitory effect of inhibitory stimuli is able to perfectly counterbalance the excitation elicited by excitatory stimuli. That means that if the neuron simultaneously receives both an excitatory input and an inhibitory input, the output response is repressed, and so the neuron remains at the resting state;
- the inhibitory input is not able to make the postsynaptic neuron firing, i.e. particular case of spiking behaviour, as rebound spike[39], are excluded;
- the firing of the only presynaptic excitatory neuron is enough to elicit the firing of the postsynaptic neuron. This can be achieved with strong synaptic strength.

Therefore, the neuron will fire if and only if the excitatory input is firing and the inhibitory input is quiet, while in any other case, it will remain in the resting state. This neuronal response is curiously similar to the truth table of the AND NOT gate. The excitatory input can be considered as the logic variable X, while the inhibitory input is the logic variable Y. Hence, a neuronal AND NOT gate can be developed considering a neuron with an excitatory synapse and an inhibitory synapse, as illustrated in Fig. 1, where the bar-headed arrow stands for the inhibitory connection. The synaptic weight w_i of the excitatory synapse has to be chosen such that if the excitatory input is high (the presynaptic neuron 1 is firing), then the postsynaptic neuron fires. This can be done empirically by considering the AND NOT gate with only the excitatory input high and increasing its w_i until the postsynaptic neuron is at the high level. From now on, we will refer to the weight value of the excitatory AND NOT gate synapse obtained with such empirical procedure as w_x. On the other hand, the synaptic weight associated to the inhibitory synapse must be set such that if both excitatory and inhibitory inputs are high, then the postsynaptic neuron must be in a resting state. Now, the value of w_i can be computed considering the AND NOT gate with the weight of the excitatory synapse w_x previously found, setting both inputs high and increasing the inhibitory weight until the postsynaptic response is repressed. We call w_y the value of the weight found for the AND NOT gate inhibitory synapse.

The NOT gate consists of a single input gate, which implements the logical negation. That means that if its input is 0, the output response is 1, and vice versa. It can be noticed that the NOT gate can be directly obtained from the neuronal AND NOT gate. Indeed, looking at the AND NOT gate Truth Table 2, the output of the last two conditions, where $X = 1$, corresponds to the negation of Y. Therefore, an AND NOT gate in which the input X is always high implements a NOT gate that respects the input Y. Adopting the aforementioned scheme, a neuronal NOT gate can be designed by using an excitatory input that is always at a high level, namely a neuron that is always firing. Spontaneously and continuously active neurons are demonstrated to exist in the brain[3840]. For instance, they are involved in the awake condition, and sleep requires their inhibition. Notice that, since the logic NOT gate has only one input, here the “effective" input is represented by the inhibitory input. Figure 2 depicts the functional scheme of the NOT gate. The continuously active neuron is implemented as a presynaptic neuron whose external stimulating current is always ON. Since we are interested only in the state of the inhibitory input, the continuously firing neuron is simply represented by an ’H’, which stands for high level. Since the NOT gate shares the same network structure of the AND NOT gate, the synaptic weights assume the same values w_x and w_y.

Network synchronisation and neuronal buffers

Our preliminary tests displayed that inhibition is highly sensitive to input timing. Indeed, considering the neuronal AND NOT gate, if the excitatory and inhibitory inputs are not well-synchronized, the inhibition could not be effective. Therefore, even though both inputs are high, the output response could not be the desired resting state, but rather the firing state. Synchronisation becomes a critical issue in the case of logic gates cascading or circuits. Considering more logic gates consecutively connected, the implementation of each Izhikevich neurons elaborates the received synaptic inputs to compute the resulting membrane potential. These operations require a finite amount of time. Therefore, each neuron block inevitably introduces a delay on the chain of signals transmission. As a consequence, input synchronisation is not always ensured. Here we address to the aforementioned challenges as follows. First, the time constants related to the synapses are increased, such that the variations of the conductances require more time and so the inhibitory and excitatory effects last longer. Secondly, using an approach inspired by digital electronics, we introduce the use of *neuronal buffers*. In digital electronics, a digital buffer is a gate whose output state is equal to its input state. Digital buffers are commonly used for electrical isolation, impedance matching and for restoring degraded digital signals after propagation over long distances. For our purpose, the buffer can be used to simply introduce a delay in the input signals and counterbalance their asynchronization. A neuronal buffer can be developed as a neuron with a unique excitatory synapse, with a value of synaptic weight chosen such that if its associated intput neuron is firing, also the output neuron fires. Notice that the AND NOT gate also receives only one excitatory input that is able to make its output fire. Hence, the buffer synaptic weight can be chosen as the weight of the excitatory synapse of the AND NOT gate w_x. The input synchronisation along the neuronal circuits is guaranteed, ensuring the following condition: when more gates are connected together, logic gate inputs must pass through the same number of neuron layers in order to introduce the same delay. Whenever a branch of the neuronal circuit does not respect this condition, neuronal buffers are added. Logic circuits with poor synchronisation can be fixed with the use of neuronal buffers.

Neuronal Cascaded Circuits

Digital electronics often benefits from the use of NAND gates. Specifically their importance is related to the fact that any Boolean function can be implemented with a combination of NAND gates. This powerful feature, which also includes the NOR gates, is called functional completeness. The NAND gate is defined as an AND gate followed by a negation operation. Consequently, its output is low only if both its inputs are high, while it is high in any other cases (Fig. 2e). Since we already discussed how to develop the neuronal AND gate and the neuronal NOT gate, a straightforward implementation of the neuronal NAND gate can be achieved by cascading a neuronal AND gate with a NOT gate. This simple cascaded structure is depicted in Fig. 2f). The same values of synaptic weights w_x and w_y of the AND NOT gate implementation can be used. Instead, we refer to the synaptic weight value of the AND synapses as w_z. Now lets focus on the synchronisation of the scheme in Fig. 2f). Consider the neuron related to the NOT gate, with membrane potential v_4. Its inhibitory synaptic input passes through two layers of neurons, specifically the one associated to the presynaptic neurons (with membrane potentials v_1 and v_2) and the one related to the AND gate (with membrane potential v_3). On the contrary, the excitatory synaptic input passes only through one layer of neurons, which is the continuously firing neuron. Hence, the synchronisation condition is not fulfilled, and the logic circuit could not follow the desired behaviour. The circuit can be adjusted by adding a neuronal buffer in the branch related to the excitatory synapse of the NOT gate, as depicted in Fig. 2f). Since now both synaptic inputs pass through two layers of neurons, the condition is satisfied.

Neuronal latches

All the neuronal implementations of logic circuits considered so far make use of combinatorial logic. In combinatorial circuits, their outputs are fully defined by the present values of the inputs. Therefore, they could be defined as static, meaning that they do not depend on the previous values of inputs and outputs. Differently in sequential logic circuits, the definition of the

outputs relies also on the sequence of past inputs. For this reason the logical outputs are often referred as *states*, which recall the storage capability of sequential logic.

One of the simplest sequential circuits is the *set-reset latch* (SR latch). It consists in a 1-bit memory, in which its state is defined asynchronously. SR latches are commonly built using NOR gates or NAND gates. Even though we already presented a model of neuronal NAND, here we discuss the realisation of SR latch based on AND NOT gates. NAND gate implementation involves a larger number of neurons that respect the AND NOT gate, and hence its computational cost is higher. The logic scheme of the AND NOT gate-based SR latch is shown in Fig. 3. The SR latch is comprised of two AND NOT gates, in which each inverted input (the *Y* input of the original AND NOT gate) is obtained using the output of the other gate. Here, the not inverted inputs (the *X* input) are called set (*S*) and reset (*R*). The two neuronal AND NOT gates have mutual inhibitory feedback, meaning that the response of the output neuron of each gate is used as the inhibitory input of the other gate. The functional scheme of the neuronal SR latch is displayed in Fig. 3b). The synaptic weights of the excitatory and inhibitory synapses can be chosen according to the AND NOT gate implementation. A critical point of the neuronal latch is that here, since inhibitory inputs are obtained from the outputs of the gates, they will always be delayed relative to the excitatory inputs. Indeed, the rule based on the number of neuronal layers is not respected, and there is no neuronal buffer configuration that can re-equalize them. As instance, if we add a buffer in the branch related to the excitatory input of AND NOT gate 1, this gate will now satisfy the rule, but the inhibitory input of AND NOT gate 2 will be delayed even more. The strategy that we adopt here to solve this issue consists in making the inhibitory effect last longer and relaxing the synchronisation requirement. This can be achieved exploiting the temporal summation mechanism, obtained with increased time constants of the inhibitory synapses.

As previously observed, the SR latch is always transparent. This circuit can be further modified with an additional input, called *latch enable* (LE), to develop a latch which becomes transparent only for a specific level of such input. This type of latch is often called a *gated SR latch*. It represents an example of synchronous circuit and specifically defined as level-sensitive because its transparency depends on the level of the clock signal *LE*. The logic scheme of the gated SR latch is depicted in Fig. 3e–h. It is composed of an SR latch in which the *S* and *R* inputs pass through a first layer of NAND gates. The NAND gates take as inputs the actual set and reset of the gated latch, which we call S' and R', and the *LE* input, which is in common between both gates. The overall behaviour of such device is shown in Truth Table 3. A neuronal gated SR latch can be developed connecting the neuronal building blocks which implement the NAND gates and the SR latch following the same scheme of the gated SR latch, as illustrated in Fig. 3.

Neuronal flip-flops

Contrary to gated SR latch that are level-sensitive, i.e., adopting the clock signal (*LE*) as a periodic square wave, the latch is transparent only during the high semi-periods of the clock[41], flip-flop circuits are defined as edge-sensitive. They become sensitive only for a brief time, which is triggered by the transition of the clock between the two levels. In other words, the circuit samples the input values at the clock edge. Depending on the type of flip-flop, this could be the rising edge (*positive-edge flip-flop*), the falling edge (*negative-edge flip-flop*) or both of them (*dual-edge flip-flop*).

A straightforward modification of the primary-secondary flip-flop can be done in order to avoid the not-allowable condition. The S' and R' signals are substituted by a unique input signal *D*, which is sent directly for the *S* input and inverted for the *R* input, as depicted in Fig. 4. The above-mentioned flip-flop is called *D flip-flop*, where D stands for data, because it requires a unique data input, or for delay, because the variations of the input are reported in the output after a certain delay defined by the next clock edge. The network structure that implements the neuronal D flip-flop is illustrated in Fig. 4. With this large circuit, the use of neuronal buffers becomes fundamental, especially in the neuronal NOT gates, due to the additional connection with the continuously firing neurons. Notice that all the sequential circuits presented in the current and the previous Sections can be implemented by using AND NOT gates, NOT gates, AND gates, and neuronal buffers. Therefore, all the values of synaptic weights can be chosen according to the ones defined in each building block. On the whole, three values of synaptic weights are needed, especially w_x and w_y, which are in common between the AND NOT gates, the NOTs, and the buffers, and w_z, which is used for the synapses of the AND gates.

Neuronal Energy Model

To account for metabolic constraints on spiking, we integrate a modified energy-based leaky integrate-and-fire (eLIF) model[42]. The energy variable in the eLIF model is a proxy for the ATP/ADP ratio, reflecting the neuron's metabolic state. This variable affects the neuron's firing capability, with a lower energy state limiting the neuron's ability to emit spikes. Energy dynamics are determined by a balance between energy consumption due to spike generation and maintenance of membrane potential, and energy production, which is primarily governed by mitochondrial oxidative phosphorylation.

The eLIF model thus provides a biophysically plausible method to study the interplay between neuronal electrophysiology and energy metabolism. It offers a valuable tool for exploring how energy limitations may contribute to the pathophysiology of various neurological conditions.

$$\text{if } V < V_{th} \text{ or } \varepsilon < \varepsilon_c \begin{cases} C_m \dot{V} = g_L(E_L - V) + I_{syn} + I_e \\ \tau_\varepsilon \dot{\varepsilon} = (1 - \frac{\varepsilon}{\alpha\varepsilon_0})^3 \frac{V - E_f}{E_d - E_f} \\ E_L = E_0 + (E_u - E_0)(1 - \frac{\varepsilon}{\varepsilon_0}) \end{cases} \quad \text{else} \quad \begin{cases} V \leftarrow V_r \\ \varepsilon \leftarrow \varepsilon - \delta \end{cases} \tag{1}$$

In Eq. (1), the membrane potential (V) is the electrical potential difference across the neuron's membrane, essential for the transmission of the neuronal signal. The membrane capacitance (C_m) reflects the neuron's ability to store and maintain charge, while the leak conductance (g_L) represents the propensity of the neuron to lose ions across its membrane, affecting the membrane potential. The leak reversal potential (E_L) is the equilibrium potential for the leak current.

Synaptic currents (I_{syn}) represent the inputs from other neurons, and external input currents (I_e) corresponds to stimuli from outside the neuronal network. The energy variable (ε) is analogous to the ATP/ADP ratio and impacts the neuron's firing capability. The threshold potential (V_{th}) is the critical value the membrane potential must exceed for the neuron to fire an action potential, and the reset potential (V_r) is where the membrane potential is set post-spike. The time constant for the energy variable (τ_ε) dictates the pace at which the energy state changes.

Furthermore, ε_c indicates the critical energy level required for spike generation, while α is a measure of the neuron's energetic health. The typical energy value (ε_0) reflects the baseline energy state of the neuron, and δ is the energy cost incurred by the neuron upon firing an action potential. These variables are interwoven in differential equations that capture the dynamic interplay between the neuron's membrane potential and its energetic state, offering insights into how electrophysiological properties are modulated by metabolic constraints[42].

Data availability

The datasets generated and/or analyzed during the current study are not publicly available because they consist of simulation outputs that depend on the full model configuration and analysis pipeline for interpretation; the scripts/configuration files and representative output traces are available from the corresponding author on reasonable request.

Received: 19 August 2025; Accepted: 17 March 2026;
Published online: 27 April 2026

References

1. Moore, A. L. & Shi, L. Emerging challenges and materials for thermal management of electronics. *Mater. Today* **17**, 163–174 (2014).
2. Grozinger, L. et al. Pathways to cellular supremacy in biocomputing. *Nat. Commun.* **10**, 5250 (2019).
3. Xue, K. et al. Biomaterials by design: Harnessing data for future development. *Mater. Today Bio* **12**, 100165 (2021).
4. Végh, J. & Berki, A.J. On the role of speed in technological and biological information transfer for computations. *Acta Biotheoretica* **70**, 26 (2022).
5. Cai, H. et al. Brain organoid reservoir computing for artificial intelligence. *Nat. Electron.* **6**, 1032—1039 (2023).
6. Kagan, B. J. et al. In vitro neurons learn and exhibit sentience when embodied in a simulated game-world. *Neuron* **110**, 3952–3969 (2022).
7. DeMarse, T. B. & Dockendorf, K. P. Adaptive flight control with living neuronal networks on microelectrode arrays. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005*., vol. 3, 1548–1551 (IEEE, 2005).
8. Barros, M. T., Doan, P., Kandhavelu, M. et al. Engineering calcium signaling of astrocytes for neural–molecular computing logic gates. *Sci. Rep.* **11**, 1–10 (2021).
9. Smirnova, L. et al. Organoid intelligence (oi): the new frontier in biocomputing and intelligence-in-a-dish. *Front. Sci.* **1**, 1017235 (2023).
10. Kagan, B. J. Two roads diverged: Pathways toward harnessing intelligence in neural cell cultures. *Cell Biomater.* **1**, 100156 (2025).
11. Bennet, D., Vo-Dinh, T. & Zenhausern, F. Current and emerging opportunities in biological medium-based computing and digital data storage. *Nano Sel.* **3**, 883–902 (2022).
12. Yim, S. S. et al. Robust direct digital-to-biological data storage in living cells. *Nat. Chem. Biol.* **17**, 246–253 (2021).
13. Basso, G. & Barros, M. T. Biocomputing model using tripartite synapses provides reliable neuronal logic gating with spike pattern diversity. *IEEE Trans. NanoBioscience* **22**, 401–412 (2022).
14. Goldental, A., Guberman, S., Vardi, R. & Kanter, I. A computational paradigm for dynamic logic-gates in neuronal activity. *Front. Computational Neurosci.* **8**, 52 (2014).
15. Knijnenburg, T. A. et al. Logic models to predict continuous outputs based on binary inputs with an application to personalized cancer therapy. *Sci. Rep.* **6**, 36812 (2016).
16. Adonias, G. L. et al. Reconfigurable filtering of neuro-spike communications using synthetically engineered logic circuits. *Front. Comput. Neurosci*. **91**, 91–107 (2020).
17. Adonias, G. L. et al. Utilizing neurons for digital logic circuits: a molecular communications analysis. *IEEE Trans. NanoBioscience* **19**, 224–236 (2020).
18. Yuan, Y., Huo, H. & Fang, T. Effects of metabolic energy on synaptic transmission and dendritic integration in pyramidal neurons. *Front. computational Neurosci.* **12**, 79 (2018).
19. Engl, E. & Attwell, D. Non-signalling energy use in the brain. *J. Physiol.* **593**, 3417–3429 (2015).
20. Zhu, F., Wang, R., Pan, X. & Zhu, Z. Energy expenditure computation of a single bursting neuron. *Cogn. Neurodyn.* **13**, 75–87 (2019).
21. Roth, A., van Rossum, M. C. W. et al. Modeling synapses. *Computational modeling methods neuroscientists.* **6**, 139–160 (2009).
22. Almog, M. & Korngreen, A. Is realistic neuronal modeling realistic? *J. Neurophysiol.* **116**, 2180–2209 (2016).
23. Florio, M. D., Kahana, A. & Karniadakis, G. E. Analysis of biologically plausible neuron models for regression with spiking neural networks (2023). 2401.00369.
24. Radulescu, A. R. Mechanisms explaining transitions between tonic and phasic firing in neuronal populations as predicted by a low dimensional firing rate model. *PloS one* **5**, e12695 (2010).
25. Connelly, W. M., Crunelli, V. & Errington, A. C. Variable action potential backpropagation during tonic firing and low-threshold spike bursts in thalamocortical but not thalamic reticular nucleus neurons. *J. Neurosci.* **37**, 5319–5333 (2017).
26. Matityahu, L. et al. A tonic nicotinic brake controls spike timing in striatal spiny projection neurons. *Elife* **11**, e75829 (2022).
27. Novellino, A. et al. Connecting neurons to a mobile robot: an in vitro bidirectional neural interface. *Computational Intell. Neurosci.* **2007**, 12725 (2007).
28. Martin, L., Rizzi, L. G. & Vassanelli, S. Engineered biological neuronal networks as basic logic operators. *Front. Computational Neurosci.* **19**, 1559936 (2025).
29. Simpson, D., Harkin, J., McElholm, M. & McDaid, L. Smart watchdog for risc-v: A novel spiking neural network approach to fault detection. *IEEE Transactions on Circuits and Systems II: Express Briefs* (2025).
30. Farsa, E. Z., Ahmadi, A. & Keszocze, O. Reconfigurable digital fpga implementations for neuromorphic computing: A survey on recent advances and future directions. *IEEE Transactions on Emerging Topics in Computational Intelligence* (2025).
31. Farsa, E. Z., Ahmadi, A., Maleki, M. A., Gholami, M. & Rad, H. N. A low-cost high-speed neuromorphic hardware based on spiking neural network. *IEEE Trans. Circuits Syst. II: Express Briefs* **66**, 1582–1586 (2019).
32. Heidarpur, M., Ahmadi, A., Ahmadi, M. & Azghadi, M. R. Cordic-snn: On-fpga stdp learning with izhikevich neurons. *IEEE Trans. Circuits Syst. I: Regul. Pap.* **66**, 2651–2661 (2019).
33. Pehle, C. et al. The brainscales-2 accelerated neuromorphic system with hybrid plasticity. *Front. Neurosci.* **16**, 795876 (2022).
34. Frenkel, C., Lefebvre, M., Legat, J.-D. & Bol, D. A 0.086-mm^2 12.7-pj/SOP 64k-synapse 256-neuron online-learning digital spiking neuromorphic processor in 28-nm CMOS. *IEEE Trans. Biomed. Circuits Syst.* **13**, 145–158 (2019).
35. Davies, M. et al. Loihi: A neuromorphic manycore processor with on-chip learning. *Ieee Micro* **38**, 82–99 (2018).
36. Akopyan, F. et al. Truenorth: Design and tool flow of a 65 mw 1 million neuron programmable neurosynaptic chip. *IEEE Trans. computer-aided Des. Integr. circuits Syst.* **34**, 1537–1557 (2015).
37. Izhikevich, E. M. Simple model of spiking neurons. *IEEE Trans. neural Netw.* **14**, 1569–1572 (2003).
38. Yoder, L. Explicit logic circuits discriminate neural states. *PloS one* **4**, e4154 (2009).
39. Izhikevich, E. M. Which model to use for cortical spiking neurons? *IEEE Trans. neural Netw.* **15**, 1063–1070 (2004).
40. McCormick, D. A. Neuronal networks: Flip-flops in the brain. *Curr. Biol.* **15**, 294–296 (2005).
41. Pedroni, V. A.*Digital electronics and design with VHDL* (Morgan Kaufmann, 2008).
42. Fardet, T. & Levina, A. Simple models including energy and spike constraints reproduce complex activity patterns and metabolic disruptions. *PLoS Computational Biol.* **16**, e1008503 (2020).

Acknowledgements

No funding was received for this research.

Author contributions

M.T.B. and G.B. designed the study. G.B. conducted the in-silico experiments, collected and curated the data. M.T.B., G.B., and R.S. analysed the results. M.T.B., G.B., and R.S. wrote and revised the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s44335-026-00066-4.

Correspondence and requests for materials should be addressed to Michael Taynnan Barros.