Hyper-Flexible Neural Networks: Rapidly Switching between Logic Operations in a Compact 4-Neuron Circuit

Alexander James White[1 2 * ✉], Belle Liu[1 3 *], Ming-Ju Hsieh[1 3], Kuo-An Wu[3], and Chung-Chuan Lo[1 4 ✉]

[1] *Institute of Systems Neuroscience, National Tsing Hua University, Hsinchu, Taiwan*

[2] *Interdisciplinary PhD Program, National Tsing Hua University, Hsinchu,Taiwan*

[3] *Department of Physics, National Tsing Hua University, Hsinchu, Taiwan*

[4] *Brain Research Center, National Tsing Hua University, Hsinchu, Taiwan*

Abstract

Neural networks of various levels exhibit rapid adaptability to diverse environmental stimuli. Such fast response times imply that adaptation cannot rely solely on synaptic plasticity, which operates on a much slower timescale. Instead, circuits must be inherently hyper-flexible and receptive to switches in functionalities. In this study, we show that a 4-neuron circuit can rapidly and controllably switch between 24 unique functions, while maintaining the same set of synaptic weights. Specifically, in order to systematically classify the outputs relative to inputs, we classify unique types of information processing in terms of 8 non-trivial logical truth tables (AND, OR, XOR, etc.). Furthermore, we test 3 different classes of input characteristics — difference in magnitude, timing and phase between input signals — and show that this small circuit can switch between different computations simply by adjusting its bias current. Finally, we demonstrate that this flexibility can be used to reduce the traditional nine gate adder into two 4-neuron circuits. This provides a computational foundation for how neural adaptability can occur on timescales much shorter than plasticity, an aspect important yet less explored in previous literature. This is a novel way to control a neural circuit, and could lead to new types of computing, especially in the growing field of neuromorphic computing.

* Contributed equally to this work.

✉ Send Correspondence to Chung-Chuan Lo: cclo@mx.nthu.edu, or Alexander White: ajw@lolab-nthu.org

Introduction

Across all organisms, nervous systems have been capable of dealing with the vast variability of their environment. For instance, studies have shown how small neural networks in crabs and aplysia are able to switch their function in response to different environment conditions [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]. Furthermore, cortical studies have shown how contextual cues can cause the networks to change activity patterns [13, 2, 7, 14, 15, 16, 17, 18]. Collectively, these phenomena of performing different behaviors based on contextual or environmental information is known as flexibility.

Interestingly, this flexibility is important beyond just choosing the right behavior. There is evidence that flexible responses occur on a microscale [19, 17]. That is, contextual information from elsewhere in the brain will cause different responses in small collections of neurons [15, 16, 19, 17, 18, 20, 21, 22]. This flexibility extends beyond just choosing the right behavior, and can also be used in information gating [7, 23, 24, 25, 22], motor planing [26, 5, 1, 17], and influenceing decision networks [7, 23, 24, 25, 17, 4]. Moreover, there is evidence that flexibility can exist on a microscale – that is, contextual information can elicit different responses in small collections of neurons, not just large-scaled networks [19, 17, 15, 21, 23, 27, 22]

This mechanism is noticeably absent in artificial neural networks, especially deep and convolutional neural networks (DNN and CNN). Such networks perform a single, designated task exceptionally well, but cannot generalize well as the solution hinges on a set of learned synaptic weights[28, 29, 30]. If the task changes, so must the weights. [31, 32, 33, 28, 29]. While animals do indeed use this strategy to learn tasks, the lack of flexibility indicates that synaptic weights alone do not explain the whole story. Many organisms are able to respond to a diverse array of situations without having to relearn a new set of synaptic weights [15, 19, 5]. This is most clear in the motor cortex, where synaptic weights don't change from task to task [26, 1, 5, 30].

It is suspected that flexibility in the nervous system stems from its highly recurrent nature. Furthermore, these networks contain bifurcation, a qualitative change in the behavior of the system as a parameter, such as contextual input, is varied, leading to the emergence of new dynamic states. Moreover, it is proposed that this is a consequence of the network being near multiple bifurcations, i.e. at a position in which state changes are easy to induce [26, 34, 35, 36, 37, 38]. Being near the cusp of state changes allows the network to rapidly switch to qualitatively different dynamics upon manipulation of certain control parameters. However, while several studies have attempted to understand how bifurcations aid flexibility[39, 38, 4], it is still unclear how the nervous system can control networks near such complicated bifurcation structures without altering synaptic weights.

In our previous work, we identified a hyper-flexible 4-neuron circuit, named CRIREL (Coupled Recurrent Inhibitory and Recurrent Excitatory Loops). It is composed of two sets of strongly recurrently coupled neurons: a

recurrently connected excitatory pair and a recurrently connected inhibitory pair, where these pairs are then weakly coupled with one another. We showed that CRIRELs are flexible because we can control an underlying double cusp bifurcation present in the network [37, 40, 38], which allows for new stable or unstable states to emerge as control parameters (in this case, synaptic weights) are varied. However, synaptic weights are not the only controllable parameters, and here we demonstrate how varying bias currents (baseline activity level, implemented as a constant background input current) and input, while keeping synaptic weights fixed, also induces bifurcations and results in flexibility. Therefore, through a detailed study of the dynamics of this microcircuit, we propose a template for how rapid flexibility can be present in neural networks without evoking slow time scale adaptation mechanisms.

In order to systematically investigate the repertoire of functionalities within CRIREL, we classify its output in terms of logic gate operations (AND, OR, etc.). These operations could be performed in terms of different types of input characteristics, and here we choose three that are often relevant in neuroscience – the difference in magnitude, timing, and phase of two input signals. Therefore, during one computation, the CRIREL circuit would be comparing one aspect of two input signals, and produce a response. For example, performing an AND operation in terms of input magnitude means that the circuit will only report "on" whenever two input values have the same magnitude, and will report "off" otherwise. We found it natural to classify possible unique types of outputs given these two inputs in terms of logical truth tables (i.e. AND, OR, XOR, etc.) [Table 6]. This has the advantage of cleanly classifying all the different ways the network operates upon its input. We show that for all three types of input classes, we are able to generate all 8 possible nontrivial logic truth tables. Moreover, we show all 24 unique functions coexist for a given synaptic weight and that changes in functionality arise solely from the change in the bias current.

Results

The CRIREL circuits

To examine flexibility in small neural networks, we start with a CRIREL microcircuit [Fig 1a] [38]. This microcircuit is simultaneously near two cusp bifurcations [Fig 1b], and as such is capable of a simple two "bit" version of working memory [Fig 1b-c]. Here, the "bits" are loosely whether one of the inhibitory neurons is ON or is OFF. Note, two identical inputs pushed the memory circuit into a $(1, 1)$ state, while asymmetric inputs pushed the memory circuit into a $(1, 0)$ or $(0, 1)$ state.

Taking this two-bit classification as a starting point, we can classify functions in terms of binary logical operations. We take this input classification scheme, but associate the network with either an ON state or an OFF

state as measured by the collective activity of the excitatory subsystem. This gives rise to 8 unique non-trivial input combinations (Table 6), where the two inputs into the circuits can be classified as 1 or 0. The binary pair $(1, 1)$ references if the inputs are "roughly" the same. Likewise, we can consider whether an out neuron is on or off as a binary output of 1 or 0.

We further add two input neurons, which convert square impulses into spike trains [Fig 1d]. These neurons are solely for the purpose of making sure that the input into the circuit is biologically realistic. We also provide an output neuron, this is to concretely categorize the variable firing rate in the excitatory subsystem as a binary output. The output is considered one if it fires, and zero if it does not.

The magnitude logic

The most straightforward type of input to consider is the differences in magnitude between two square pulse inputs [Fig 2a]. After being fed into the input layer, the magnitude of the square pulses is converted into a difference in firing rate, and roughly corresponds to different magnitudes of post synaptic excitatory input. We consider the input to be $(1, 1)$ if the magnitudes A are roughly equal. Likewise, if the inputs are different by a fixed amount, ΔA they are considered $(1, 0)$ or $(0, 1)$. An absence of input is considered $(0, 0)$. It is important to note, that the magnitude A and difference in magnitude ΔA are both free parameters [sFig 2].

As a concrete demonstration, we begin with an AND gate. For the particular example, we choose the Magnitude for 1 to be $A = 1\ nA$ for the LIF neuron. We set $\Delta A = 0.3\ nA$. When the network is presented with the input protocol, we see that the output neuron fires only when the signal is in the $(1, 1)$ state [Fig 2b]. Simply by changing the bias currents, we can change the gate into an XOR gate [Fig 2c], where during the $(1, 1)$ signal the output is silent, but fires whenever the signal is in the $(1, 0)$ or $(0, 1)$ signal. Next, we show how we can change the bias currents, from AND to XOR in the middle of the input protocol [Fig 2d] and the gate switches with a latency of 40 $msec$ or about two spikes [Fig 2e]. A majority of the transitions are possible, but a few require global recent (strong negative input)[sFig 1].

Next, we show that all 8 unique logic gates exist for at least some set of bias currents. We test this for LIF neurons [Fig 3a] and for IZH neurons [Fig 3b]. The bias currents used to produce these 8 gates can be found in the supplement [sTab 7 and sTab 8]. We provide the parameter sweep for 6 of the logic gates as a 2-dimensional sweep of excitatory subsystem bias and inhibitory subsystem bias [Fig 3c and 3d]. Because IMP and NIMP require asymmetric bias currents, they don't fit neatly into a 2-dimensional sweep, and are not shown.

Most of the logic gates are robust against variation in the bias currents. This robustness can be quantified as the square root of the area of each gate's parameter region in the 2-dimensional sweep[fig 3c] of excitatory

subsystem bias, and inhibitory subsystem bias [Fig 3e]. This roughly quantifies the change in bias that each gate can tolerate. Some gates, like XOR and NAND are very robust, while XNOR and NOR, require greater precision in bias current.

The gates also respond rapidly to the presentation of input. To quantify response time, we measure how long it takes for the '0' (OFF) response of the output neuron to switch to a '1' (ON) response. Here, we measure the time difference from when the input is presented to the network to the timing of the first spike in the output neuron. We sampled multiple different bias current configurations, some are faster than others, but all roughly respond in a few tens of milliseconds, which matches the scale of the membrane time constant of the neurons in the models.

The temporal logic

The CRIREL logic gates are not limited to processing the differences between the magnitude of two inputs, but can process the differences in the timing between two inputs [Fig 4a]. Here we define two inputs arriving at the same time as $(1, 1)$. Likewise, two inputs arriving at different times are $(1, 0)$ and $(0, 1)$. As before, the absence of any signal is considered $(0, 0)$. Note that the magnitude of all inputs is the same. Again, we can test robustness by considering magnitude A and timing difference Δt as free parameters [sFig 3] It is important to note, that the synaptic weights are the same as the synaptic weights in the magnitude example, only the input protocol was changed.

As a concrete demonstration, we show a XNOR gate (not exclusive or) [Fig 4b]. Here, the network and thus output neuron fires in the absence of input $(0, 0)$. Whenever the signal $(1, 1)$ is present, the firing rate increases. However, whenever two signals arrive at different times $(1, 0)$ and $(0, 1)$, the circuit turns itself off, and the output neuron is silent. We can easily transform the network into any other of the 8 logic gates. Here we show the NAND gate[Fig 4c], again the absence of signal $(0, 0)$ is firing, but the network fires when there is a timing difference $(1, 0)$ and $(0, 1)$ and is silent when the timing is the same $(1, 1)$. We also highlight the NIMP gate, which is the logical negation of the Imply gate [Fig 4d]. This gate shows no response if the timing is perfectly synchronized $(1, 1)$ or the blue signal leads $(1, 0)$, however shows a response if the yellow signal leads $(0, 1)$. We refer to this also as a surprise gate, as one can imagine if the blue signal is trying to predict the yellow signal (with some time lag) the network remains silent. However, if the prediction fails, the network turns on to alert to a failure of prediction. Finally, we conclude with an XOR gate example [Fig 4e].

Again, we show that all 8 unique logic gates exist for at least some set of bias currents, and test this for both LIF neurons [Fig 5a] and for IZH neurons [Fig 5b]. The bias currents used to produce these 8 gates can be found in the supplement[sTab 9 and sTab 10]. We provide the parameter sweep for 6 of the logic gates as a 2-dimensional

sweep of excitatory subsystem bias, and inhibitory subsystem bias [Fig 5c and 5d]. Because IMP and NIMP require asymmetric bias currents, they don't fit neatly into a 2-dimensional sweep.

The phase logic

Given the ability to look at the relative timing of two inputs, it is natural to ask if computation can be done on phase differences between two periodically repeating inputs. Again, the input neurons take a pure sinusoidal signal and convert it into a spiking signal [Fig 6a]. Here it is important to note we do not convert the sine-wave into a spiking signal, rather we feed the raw sine-wave into the CRIREL network directly.

As expected, the definition of the inputs and output change. We consider $(1, 1)$ to be when the two input waves are synchronized. Similarly, we consider $(0, 1)$ to be phased delayed and $(1, 0)$ to be phased advanced. Interestingly, we can define $(0, 0)$ as two anti-synchronized inputs. Here, an output neuron is considered ON if, at any time during a full cycle of the sinusoidal input, the output neuron spikes. Often, the output neuron will fire during the maximum of the input wave [Fig 6b and 6c], but for anti-synchronized $(0, 0)$ the output neuron fires during the average value[Fig 7a NOR].

As a concrete example, we point out 3 examples: AND [Fig 7a AND], NIMP Advanced [Fig 6b] and NIMP delayed [Fig 6c], and NOR[Fig 7a NOR]. For example, AND acts as a coincidence detector and fires only when the two signals are synchronized [Fig 7a AND]. NIMP (or Not Imply) we call the surprise detector[Fig 6b], as if you imagine one wave predicting the other wave, then the output neuron will only fire when the prediction lags behind the measurement, that is the prediction is no longer accurate. More concretely, that means NIMP's output neuron fires whenever it is in the $(1, 0)$ state, not the $(0, 1)$. Notice, that the distinction between $(1, 0)$ and $(0, 1)$ is arbitrary, and as such there are two symmetric versions of NIMP that can be easily switched between by flipping the asymmetry present in the bias currents [Fig 6c]. The NOR gate [Fig 7a] only fires when the two waves are perfectly anti-synchronized, it is silent in the other three cases.

We also show that all 8 unique logic gates exist for phase type computations, and test this for both LIF neurons [Fig. 7a]. The bias currents used to produce these 8 gates can be found in the supplement [sTab 11]. We provide the parameter sweep for 6 of the logic gates as a 2-dimensional sweep of excitatory subsystem bias, and inhibitory subsystem bias [Fig 7b]. Because IMP and NIMP require asymmetric bias currents, they don't fit neatly into a 2-dimensional sweep.

It is also important to discuss how precise the synaptic weights must be. The gates would be useless if they require a specific synaptic weight and cannot tolerate any deviation from this value. We find that most gates are robust up to 10% difference in the synaptic weights [sFig 4].

Application: the adder circuits

Flexible gate switching is a unique property of CRIREL logic gates. To demonstrate its advantages, we recreated the well-documented 4-bit ripple carry adder circuit, which is a digital logic circuit used for performing multi-bit binary addition. It is composed of four 1-bit full adders [Fig 8a]. A full adder takes in three inputs, the two summands, and the carry from the previous digit [Fig 8b]. This is fed into two CRIREL's, one for computing the sum, the other for computing the carry. We add a buffer neuron to regularize the output firing rate. Note that a traditional transistor based adder requires 9 NAND logic gates[Fig 8c] [41], while only two neuromorphic logic gates are required only.

Here, it is important to note that the two summands are the traditional magnitude inputs from above. Here the carry induces a change in the bias current of the CRIREL's to switch the logic gate, from AND/OR for the carry CRIREL, and XOR/XNOR for the sum CRIREL [Fig 8d]. The bias currents used to switch can be found in [TAB] and are graphically represented as well [Fig 8e]. To implement a 4-bit ripple adder, we connect four 1-bit full adders together [Fig 8f]. A 1-bit adder requires at least 50 milliseconds to generate a correct response time [Fig. 8g]. Carry propagation delay necessitates a minimum 200-millisecond delay for a 4-bit ripple carry adder.

Finally, we investigated CRIREL's noise tolerance, which is essential to a network's usability. We applied Gaussian noise to the bias and measured the accuracy of the logic operation (see Materials and Methods). We quantified both absolute noise (measured in nA) and relative noise (percentage of bias current). Results show CRIREL-based adders have a 0.1nA tolerance for absolute noise and a 10% tolerance for relative noise [Fig. 8h] (XNOR is an exception ranging between -10% to $+5\%$).

Discussion

In this study, we conclusively show that small spiking recurrent neural microcircuits near a double-cusp bifurcation are incredibly flexible. Here, by picking a set of synaptic weights such that the bias currents can controllably unfold the double-cusp bifurcation, we are able to switch between an incredibly diverse range of different operations. To concretely define unique functions, we used logical truth tables to associate each function with a logic gate. Moreover, we showed that these logic gates can process three different types of inputs, namely, magnitude, temporal and phase. Furthermore, we found that the gates are robust against membrane noise, input properties, operated over a range of synaptic weights. Strikingly, we showed that these results were independent of the neuron type, as we found working versions in LIF and IZH neurons. Finally, we showed that the flexible gate switching can be used to create a 4-bit ripple carry adder.

This is in line with previous research that has shown that recurrent neural networks are more flexible than their feedforward counterparts [38]. Specifically, this study takes advantage of a nearby cusp bifurcation [37]. This tracks with the plethora of studies that show networks near bifurcations are inherently more flexible [26, 34, 35, 36, 37]. Studies have been able to embed complex functions into networks that are near multiple bifurcations [26, 40, 30], and other studies have shown, that networks far away from bifurcations means that the network has redundant neurons [37]. Here, we push this idea to its limit, by fitting 24 unique functions into a 4-neuron microcircuit. Additionally, we show that one can rapidly switch between these functions without changing the synaptic weights, making use only of the bias currents.

This provides a particularly powerful technique, as often switching functions is seen as a neuromodulatory process [10, 11, 12, 1], or a synaptic weight process [32, 31, 30, 29, 28]. While clearly, these process can and do change a network's function, they are not the only means. Contextual information, and top-down control have been shown to optimize [34, 35, 15, 17, 19, 42, 43], or even switch the function of a decision networks in the prefrontal cortex [44, 45, 4]. Here, we show that bias current (which can be thought of as weak constant input) can rapidly switch the function of a microcircuit, allowing a potential mechanism for integrating contextual information or top-down control. This is a tantalizing direction for future investigation, especially if a learning algorithm can be used to embed multiple tasks into recurrent networks.

Another potential avenue for future work involves the robustness of some of the gates. Gates like AND were the least robust of the operations, being the least tolerable to noise. However, it is known that gap junctions are capable of averaging out the activity levels of two neurons. If a gap junction (which is commonly observed in parvalbumin-containing neurons [46, 47]) is added between the inhibitory neurons, this should expand the basin attraction associated with the $(1, 1)$ state. A larger $(1, 1)$ state should translate into more robustness to noise for the AND state, at the cost of robustness of the XOR state. This remains to be thoroughly tested.

Finally, while we give a useful account of small microcircuits, the approach needs to be expanded to larger networks. Some studies have successfully embedded arbitrary motor commands into E/I balanced networks near multiple bifurcations. It remains to be seen how many unique functions can be embedded in a larger network, and if this technique of unfolding the bifurcation in predictable ways can be of use. Specifically, can learning to be used to configure a group of 4-neuron CRIREL circuits, or even a larger more general network? If so, that could open up new opportunities to minimize neuron count, but maximize task number. This could have major consequences for edge deployed neuromorphic computing.

To conclude, we made use of CRIREL networks near a double cusp bifurcation, and were able to control the unfolding of the bifurcation with the bias currents. We demonstrated that these microcircuits have a rich repertoire

of 22 unique functions. Ultimately, this provides a computational foundation for how neural adaptability can occur on timescales much shorter than plasticity. This is a novel way to control a neural circuit, and could lead to new types of computing, especially in the growing field of neuromorphic computing.

Methods

We make use of two neuron models, the Leaky Integrate-and-Fire model, and the Izhikevich model. This is to show that flexibility is a property of the network, not the underlying neurons. Moreover, it shows the flexibility is robust to model choice.

Leaky Integrate-and-Fire Model and Synaptic Model

We implement our Leaky Integrate-and-Fire (LIF) model using our in house simulator Flysim [48]. Recall an LIF neurons dynamics are given by the equation

$$C_m \frac{dV_i}{dt} = -g_L \left(V_i - E_L\right) + \sum_j I_{i,j} \tag{1a}$$

$$I_{ij} = \sum_j g_{ij} s_{ij} \left(V_i - E_{ij}\right). \tag{1b}$$

V_i is the membrane potential i, C_m is the membrane capacitance, g_L denotes the leak conductance, and E_L represents the leak potential. Here, g_{ij} corresponds to the synaptic conductance between neurons. We use s_{ij} to denote the synaptic variable. Lastly, E_{ij} denotes the synaptic reversal potential.

We utilize an excitatory synapse (AMPA-like), referred here as Exc, and an inhibitory synapse (GABA-like), referred here as Inh, as the neurotransmitters for excitatory and inhibitory neurons. We refer to the two excitatory neurons as E1 and E2, and the two inhibitory neurons as I1 and I2 (see Fig 1A). They are almost always considered symmetric, except when different inputs or bias currents are provided. The equations for the synapses are shown below and parameters for various neurons are given in Table 1

$$\frac{ds}{dt} = -\frac{s}{\tau} + \sum_k \delta\left(t - t^k\right) \tag{2a}$$

When the membrane potential exceeds the threshold voltage ($V_{threshold}$), a neuron generates a spike and then is reset to (V_{reset}). There is no change in voltage until after the refractory period lapses.

$$V_i = \begin{cases} \xrightarrow{spike} V_{reset}, V_i > V_{threshold} \\ V_i \qquad\qquad , V_i \geq V_{threshold} \end{cases} \tag{3}$$

Neuron	Synapse Type	E_{ij} (mV)	τ (ms)
E_A, E_B, E1, E2, E_{Out}	Exc	0	2
Buffer	Exc	0	20
I1, I2	Inh	-90	5

Table 1: Reversal potentials and time constants for different neurotransmitters.

Neuron	E1, E2, I1, I2,	E_{Out}, E_A, E_B	Buffer
C_m (nF)	0.5	0.5	0.5
g_L (μS)	25	25	25
E_L (mV)	-70	-70	-70
$V_{\text{threshold}}$ (mV)	-50	-50	-50
V_{reset} (mV)	-55	-55	-50.1
Refractory period (ms)	2	5	5

Table 2: Neurons parameters used in the study.

The parameters for the LIF neuron are shown in Table 2.

We must stress, the synaptic weights for every function are the same. The synaptic weights used for the LIF model are listed in Table 3.

g_{ee}	30
g_{ei}	30
g_{ie}	-60
g_{ii}	-30
g_o	50

Table 3: Synaptic values used throughout the paper, except where explicitly noted.

Izhikevich Model

The Izhikevich neuron is a nonlinear-generalized quadratic integrate and fire neuron[49]. We implement the Izhikevich neuron (IZH) in Mathematica. We use the following version of the IZH neuron

$$\frac{dV}{dt} = \frac{1}{k}V^2 - 3V + 40 + 100b - W + I_{bias} + \sum g_i S_i$$

$$\frac{dW}{dt} = a(bV - W)$$

$$\text{When } V > 130,\ V \to c,\ W \to W + d$$

Voltage was set to be akin to millivolts with an always positive voltage, as to aid eventual implementation on a neuromorphic chip. When the voltage hits 130, the system resets to voltage c and the gating variable W is reset to $W + d$. The synapse gating variables S were governed by the following equation

$$\tau \frac{dS}{dt} == S + \delta(t - t_{spike})$$

where τ is a time constant, and δ is the Dirac delta function, raising the value of the synaptic gating variable S by one unit every spike. The parameters (which should be considered unitless) are given in Table 4.

Neuron	E1, E2, I1, I2, E_{Out}
k	25
a	$\frac{1}{50}$
b	.2
c	10
d	0.1
τ	10

Table 4: Neurons parameters used in the study.

Synaptic Weights between each neuron are given in Table 5.

Protocol for Binary Inputs

To properly define a unique function, we use logic gates as a useful classification of different functions. Classically, the input of a logic gate operates on a binary input of two digits, either 0 and 1. There are 4 possible combinations of input $(1, 1)$, $(0, 1)$, $(1, 0)$, and $(0, 0)$ and two possible outputs, 1 or 0. There are 16 possible combinations of inputs and outputs (fig M1).

g_{ee}	3
g_{ei}	5
g_{ie}	-10
g_{ii}	-10
g_o	3

Table 5: Synaptic values used throughout the paper, except where explicitly noted

However, a neural circuit is not constrained to be binary. Neurons can respond to many different properties of an input. Here, we define an $(1, 1)$ input as two inputs that are approximately the same (see supplemental material for the precise range of sameness). If one input is larger in magnitude, it is $(1, 0)$ or $(0, 1)$. If one input arrives before the other (so-called temporal logic) we can also define it as $(1, 0)$ or $(0, 1)$. Finally, if one input is phase delayed to the other input, we can classify it as $(1, 0)$ or $(0, 1)$. For both magnitude and temporal logic, $(0, 0)$ is defined as an absence of input. However, for phase logic, $(0, 0)$ is defined as perfect asynchrony. The truth tables can be found in Table 6

In the LIF implementation, for both magnitude logic and temporal logic, we present the input neurons of the network with two square inputs. For the 1 symbol, the square wave has an amplitude of 5 nA. The 0 symbol has an amplitude of 3.5 nA (except during robustness testing).

In the LIF implementation, for both magnitude logic and temporal logic, we directly present the CRIREL circuit with the square pulse. For the 1 symbol the square wave has an amplitude of 2.5 The 0 symbol has an amplitude of 2.5 (except during robustness testing). Recall that the Izhikevich neuron is unitless.

For phase logic, in both versions, we present a sinusoidal input directly to the CRIREL circuit. In the LIF model, the wave has a period of 40 ms, and an amplitude of 2 nA. Phase delay was defined as $\frac{\pi}{10}$ radians. For the IZH model, the wave has a period of 400 ms, and an amplitude of 1. Phase delay was defined as $\frac{\pi}{8}$ radians.

Categorizing Logic Gates

To categorize the logic gates, we used a protocol in which the input given to the circuit follows the following order: $(0, 1), (1, 1), (1, 0), (0, 1)$, with a duration of 1 s for each input, and a window of 0.5 s between each input. We then check whether the output neuron has any spikes. If so, we label that output as 1, if not, it is 0. We then construct a logic gate truth table, and assign a logic gate accordingly

Accuracy Testing for Ripple Carry Adder

To test the accuracy and robustness to noise, we inject all neurons in the 4-neuron circuit (E1, E2, I1, I2) with a Gaussian white noise. The mean of the white noise is always zero. We sweep exponentially through increasing standard deviation strength. For the absolute noise, we sweep from a standard deviation of .01 nA to a standard deviation 1.0 nA. For the relative noise condition, we sweep from standard deviation of 0.1% of the maximal bias of each neuron to 100% of the maximal bias of each neuron. We present 1000 randomly chosen pairs of 4-bit numbers, and run 10 trials (each a different Gaussian noise). If the adder returns the correct result for all pairs of numbers and trials -that is 100 % accuracy, we consider it robust.

Declaration of Conflicts of Interest

AJW, BL and CCL are inventors in Taiwan Patent No. #I725914 and a pending United States Patent related to the research presented in this paper. Additionally, AJW, BL, and CCL have received royalty payments from Egis Technology Inc. for granting an exclusive license of these patents. The licensing arrangement has been reviewed and approved by The National Tsing Hua University in accordance with its policy on the management and commercialization of intellectual properties.

References

[1] V. A. Webster-Wood, J. P. Gill, P. J. Thomas, and H. J. Chiel, "Control for multifunctionality: bioinspired control based on feeding in Aplysia californica," *Biological Cybernetics*, vol. 114, pp. 557–588, Dec. 2020.

[2] G. R. Yang, M. R. Joglekar, H. F. Song, W. T. Newsome, and X.-J. Wang, "Task representations in neural networks trained to perform many cognitive tasks," *Nature Neuroscience*, vol. 22, pp. 297–306, Feb. 2019.

[3] D. W. Morton and H. J. Chiel, "Neural architectures for adaptive behavior," *Trends in Neurosciences*, vol. 17, pp. 413–420, Jan. 1994.

[4] C. K. Machens, R. Romo, and C. D. Brody, "Flexible Control of Mutual Inhibition: A Neural Model of Two-Interval Discrimination," *Science*, vol. 307, pp. 1121–1124, Feb. 2005.

[5] D. N. Lyttle, J. P. Gill, K. M. Shaw, P. J. Thomas, and H. J. Chiel, "Robustness, flexibility, and sensitivity in a multifunctional motor control model," *Biological Cybernetics*, vol. 111, no. 1, pp. 25–47, 2017.

[6] K. V. Shenoy, M. T. Kaufman, M. Sahani, and M. M. Churchland, "A dynamical systems view of motor preparation: Implications for neural prosthetic system design," *Progress in brain research*, vol. 192, pp. 33–58, 2011.

[7] X.-J. Wang, "Decision Making in Recurrent Neuronal Circuits," *Neuron*, vol. 60, pp. 215–234, Oct. 2008.

[8] A. Destexhe and E. Marder, "Plasticity in single neuron and circuit computations," *Nature*, vol. 431, pp. 789–795, Oct. 2004.

[9] J. L. Leonard, "Network Architectures and Circuit Function: Testing Alternative Hypotheses in Multifunctional Networks," *Brain, Behavior and Evolution*, vol. 55, no. 5, pp. 248–255, 2000.

[10] E. Marder and V. Thirumalai, "Cellular, synaptic and network effects of neuromodulation," *Neural Netw*, vol. 15, pp. 479–493, July 2002.

[11] R. M. Harris-Warrick and E. Marder, "Modulation of neural networks for behavior," *Annual Review of Neuroscience*, vol. 14, pp. 39–57, 1991.

[12] C. I. Bargmann, "Beyond the connectome: how neuromodulators shape neural circuits," *Bioessays*, vol. 34, pp. 458–465, June 2012.

[13] J. Yu, H. Hu, A. Agmon, and K. Svoboda, "Recruitment of GABAergic interneurons in the barrel cortex during active tactile behavior," *Neuron*, vol. 104, no. 2, pp. 412–427.e4, 2019.

[14] F. Najafi, G. F. Elsayed, R. Cao, E. Pnevmatikakis, P. E. Latham, J. P. Cunningham, and A. K. Churchland, "Excitatory and Inhibitory Subnetworks Are Equally Selective during Decision-Making and Emerge Simultaneously during Learning," *Neuron*, vol. 105, pp. 165–179.e8, Jan. 2020.

[15] J. Fritz, S. Shamma, M. Elhilali, and D. Klein, "Rapid task-related plasticity of spectrotemporal receptive fields in primary auditory cortex," *Nature Neuroscience*, vol. 6, pp. 1216–1223, Nov. 2003.

[16] K. V. Kuchibhotla, J. V. Gill, G. W. Lindsay, E. S. Papadoyannis, R. E. Field, T. A. H. Sten, K. D. Miller, and R. C. Froemke, "Parallel processing by cortical inhibition enables context-dependent behavior," *Nature Neuroscience*, vol. 20, pp. 62–71, Jan. 2017.

[17] C. C. Rodgers and M. R. DeWeese, "Neural correlates of task switching in prefrontal cortex and primary auditory cortex in a novel stimulus selection task for rodents," *Neuron*, vol. 82, no. 5, pp. 1157–1170, 2014.

[18] C. M. Niell and M. P. Stryker, "Modulation of visual responses by behavioral state in mouse visual cortex," *Neuron*, vol. 65, no. 4, pp. 472–479, 2010.

[19] R. S. Erzurumlu and P. Gaspar, "How the barrel cortex became a working model for developmental plasticity: a historical perspective," *Journal of Neuroscience*, vol. 40, no. 34, pp. 6460–6473, 2020.

[20] R. S. Williamson, K. E. Hancock, B. G. Shinn-Cunningham, and D. B. Polley, "Locomotion and task demands differentially modulate thalamic audiovisual processing during active search," *Current Biology*, vol. 25, no. 14, pp. 1885–1891, 2015.

[21] S. Jaramillo, K. Borges, and A. M. Zador, "Auditory thalamus and auditory cortex are equally modulated by context during flexible categorization of sounds," *Journal of Neuroscience*, vol. 34, no. 15, pp. 5291–5301, 2014.

[22] H. K. Kato, S. N. Gillet, and J. S. Isaacson, "Flexible sensory representations in auditory cortex driven by behavioral relevance," *Neuron*, vol. 88, no. 5, pp. 1027–1039, 2015.

[23] C.-T. Wang, C.-T. Lee, X.-J. Wang, and C.-C. Lo, "Top-down modulation on perceptual decision with balanced inhibition through feedforward and feedback inhibitory neurons," *PLoS ONE*, vol. 8, no. 4, p. e62379, 2013.

[24] C.-C. Lo and X.-J. Wang, "Cortico–basal ganglia circuit mechanism for a decision threshold in reaction time tasks," *Nature Neuroscience*, vol. 9, pp. 956–963, July 2006.

[25] K.-F. Wong and X.-J. Wang, "A Recurrent Network Mechanism of Time Integration in Perceptual Decisions," *J. Neurosci.*, vol. 26, pp. 1314–1328, Jan. 2006.

[26] G. Hennequin, T. Vogels, and W. Gerstner, "Optimal Control of Transient Dynamics in Balanced Networks Supports Generation of Complex Movements," *Neuron*, vol. 82, pp. 1394–1406, June 2014.

[27] D. N. Lyttle, J. P. Gill, K. M. Shaw, P. J. Thomas, and H. J. Chiel, "Neuromechanical bistability contributes to robust and flexible behavior in a model of motor pattern generation," *BMC Neuroscience*, vol. 16, p. P33, Dec. 2015.

[28] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *2016 IEEE European symposium on security and privacy (EuroS&P)*, pp. 372–387, IEEE, 2016.

[29] E. Galván and P. Mooney, "Neuroevolution in deep neural networks: Current trends and future challenges," *IEEE Transactions on Artificial Intelligence*, vol. 2, no. 6, pp. 476–493, 2021.

[30] P. Helfer, C. Teeter, A. Hill, C. M. Vineyard, J. B. Aimone, and D. Kudithipudi, "Context Modulation Enables Multi-tasking and Resource Efficiency in Liquid State Machines," in *Proceedings of the 2023 International Conference on Neuromorphic Systems*, ICONS '23, (New York, NY, USA), pp. 1–9, Association for Computing Machinery, Aug. 2023.

[31] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[32] L. Itti and C. Koch, "Computational modelling of visual attention," *Nature reviews neuroscience*, vol. 2, no. 3, pp. 194–203, 2001.

[33] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the National Academy of Sciences*, vol. 79, no. 8, pp. 2554–2558, 1982.

[34] L. J. Herstel and C. J. Wierenga, "Network control through coordinated inhibition," *Current Opinion in Neurobiology*, vol. 67, pp. 34–41, Apr. 2021.

[35] S. Sadeh and C. Clopath, "Inhibitory stabilization and cortical computation," *Nature Reviews Neuroscience*, vol. 22, pp. 21–37, Jan. 2021.

[36] G. Mongillo, S. Rumpel, and Y. Loewenstein, "Inhibitory connectivity defines the realm of excitatory plasticity," *Nature Neuroscience*, vol. 21, pp. 1463–1470, Oct. 2018.

[37] F. C. Hoppensteadt and E. M. Izhikevich, *Weakly Connected Neural Networks.* Applied Mathematical Sciences, New York: Springer-Verlag, 1997.

[38] B. Liu, A. J. White, and C.-C. Lo, "Augmenting flexibility: Mutual inhibition between inhibitory neurons expands functional diversity," *bioRxiv*, 2022.

[39] F. C. Hoppensteadt and E. M. Izhikevich, "Local Analysis of Weakly Connected Maps," in *Weakly Connected Neural Networks* (F. C. Hoppensteadt and E. M. Izhikevich, eds.), Applied Mathematical Sciences, pp. 209–218, New York, NY: Springer New York, 1997.

[40] R. D. Beer, "On the Dynamics of Small Continuous-Time Recurrent Neural Networks," *Adaptive Behavior*, 1995.

[41] A. K. Singh, *Digital VLSI Design.* Prentice Hall India, v ed., 2010.

[42] C.-C. Lo, C.-T. Wang, and X.-J. Wang, "Speed-accuracy tradeoff by a control signal with balanced excitation and inhibition," *Journal of Neurophysiology*, vol. 114, no. 1, pp. 650–661, 2015.

[43] C.-T. Wang, C.-T. Lee, X.-J. Wang, and C.-C. Lo, "Top-down modulation on perceptual decision with balanced inhibition through feedforward and feedback inhibitory neurons," *PLoS One*, vol. 8, no. 4, p. e62379, 2013.

[44] D. Durstewitz, J. K. Seamans, and T. J. Sejnowski, "Neurocomputational models of working memory," *Nature Neuroscience*, vol. 3, no. 11, pp. 1184–1191, 2000.

[45] K. A. Ferguson and J. A. Cardin, "Mechanisms underlying gain modulation in the cortex," *Nature Reviews Neuroscience*, vol. 21, no. 2, pp. 80–92, 2020.

[46] T. Fukuda and T. Kosaka, "Ultrastructural study of gap junctions between dendrites of parvalbumin-containing GABAergic neurons in various neocortical areas of the adult rat," *Neuroscience*, vol. 120, pp. 5–20, Aug. 2003.

[47] T. Fukuda, T. Kosaka, W. Singer, and R. A. W. Galuske, "Gap Junctions among Dendrites of Cortical GABAergic Neurons Establish a Dense and Widespread Intercolumnar Network," *Journal of Neuroscience*, vol. 26, pp. 3434–3443, Mar. 2006.

[48] Y.-C. Huang, C.-T. Wang, T.-S. Su, K.-W. Kao, Y.-J. Lin, C.-C. Chuang, A.-S. Chiang, and C.-C. Lo, "A single-cell level and connectome-derived computational model of the drosophila brain," *Frontiers in Neuroinformatics*, vol. 12, p. 99, 2019.

[49] E. M. Izhikevich, *Dynamical Systems in Neuroscience.* MIT Press, 2007.

1 figures

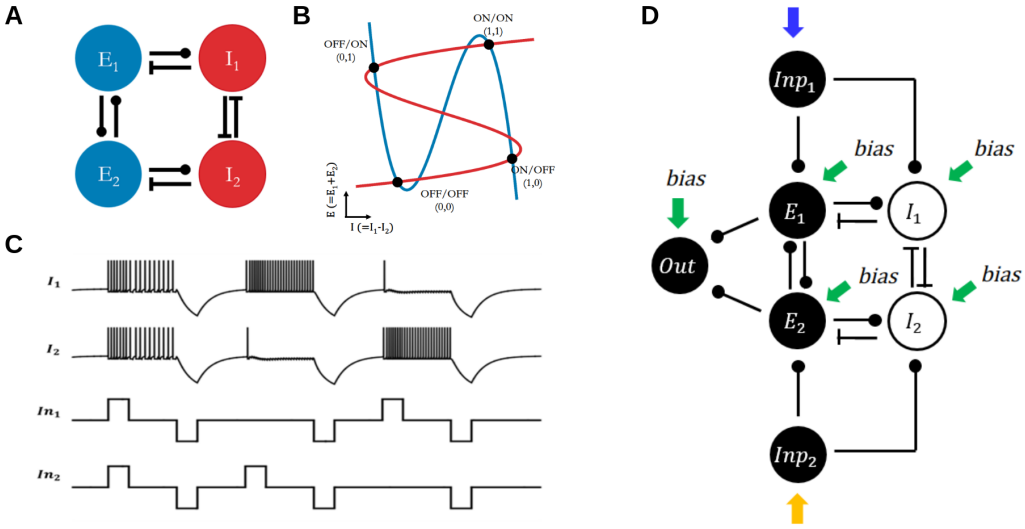


Figure 1: The CRIREL circuit. A) The circuit consists of two mutually coupled excitatory neurons connected to two mutually coupled inhibitory neurons. B) The 2-D reduced phase plane of the firing rate version of the network. The blue nullcline is the average excitation, and the red nullcline is the difference between inhibitory neurons. The particular setup is capable of storing 2-bit working memory C) An LIF example of the 2-Bit Memory storage. Depending on the input into the network, the network can either store ON/ON $(1, 1)$, ON/OFF $(1, 0)$, OFF/ON $(0, 1)$, or OFF/OFF $(0, 0)$. The network can store this information until a negative reset signal. D) The full circuit diagram to test the logic gates. Two input neurons are used to convert square pulses into spike trains. The output neuron is used to unambiguously map a 2-dimensional input to a 1-dimensional output. The input and output neurons are not essential for the computation of the CRIREL circuit.

AND			OR			NAND			XNOR		
A	**B**	**Out**	**A**	**B**	**Out**	**A**	**B**	**Out**	**A**	**B**	**Out**
0	0	0	0	0	0	0	0	1	0	0	0
0	1	0	0	1	1	0	1	1	0	1	1
1	0	0	1	0	1	1	0	1	1	0	1
1	1	1	1	1	1	1	1	0	1	1	0
NOR			**XNOR**			**1 IMPLY 2**			**2 IMP 1**		
A	**B**	**Out**	**A**	**B**	**Out**	**A**	**B**	**Out**	**A**	**B**	**Out**
0	0	1	0	0	1	0	0	1	0	0	1
0	1	0	0	1	0	0	1	1	0	1	0
1	0	0	1	0	0	1	0	0	1	0	1
1	1	0	1	1	1	1	1	1	1	1	1
ON			**OFF**			**1 NIMP 2**			**2 NIMP 1**		
A	**B**	**Out**	**A**	**B**	**Out**	**A**	**B**	**Out**	**A**	**B**	**Out**
0	0	1	0	0	0	0	0	0	0	0	0
0	1	1	0	1	0	0	1	0	0	1	1
1	0	1	1	0	0	1	0	1	1	0	0
1	1	1	1	1	0	1	1	0	1	1	0
A			**B**			**NOT A**			**NOT B**		
A	**B**	**Out**	**A**	**B**	**Out**	**A**	**B**	**Out**	**A**	**B**	**Out**
0	0	1	0	0	0	0	0	0	0	0	0
0	1	1	0	1	0	0	1	0	0	1	1
1	0	1	1	0	0	1	0	1	1	0	0
1	1	1	1	1	0	1	1	0	1	1	0

Table 6: The truth table for all 16 classical logic gates. Six are trivial: always off (false), always on (true), A, B, Not A and Not B gates. For the remaining ten gates, six are symmetric and four are asymmetric. Here we consider AND, OR, XOR, XNOR, NOR, and NAND as the six unique non-trivial symmetric gates. NIMP and IMP constitute the four unique non-trivial asymmetric gates.

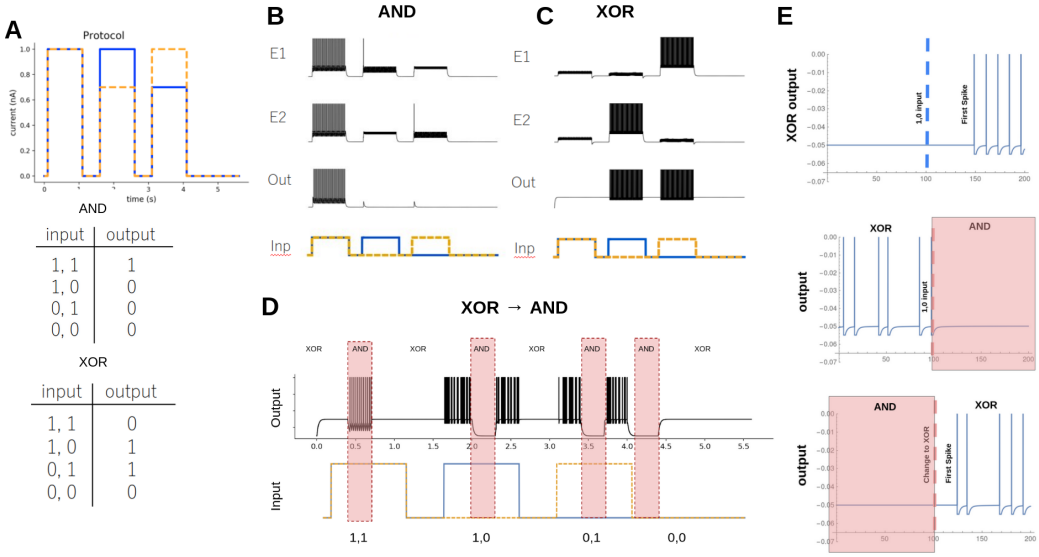


Figure 2: Magnitude Logic. A) Input protocol for the Magnitude Logic Gate. The difference between the magnitude of the square pulses determines the $(1,1)$, $(1,0)$, $(0,1)$ and $(0,0)$. The Truth tables for AND and XOR are shown. B) The LIF version of an AND gate. The activity of the excitatory neurons (E1 & E2) and the output neuron (Out) are shown. A cartoon representation of the input protocol (Inp) is also shown. C) Same as B but for a XOR gate. D) An example of the CRIREL being converted back and forth between XOR and AND gates on the fly during the signal input. E) The latency of switching output for the input a $(0,0)$ to a $(1,0)$ in an XOR gate is 47 mS or about 5 spikes. Latency of gate switching which occurs on the scales of 20 ms or about 2 spikes.

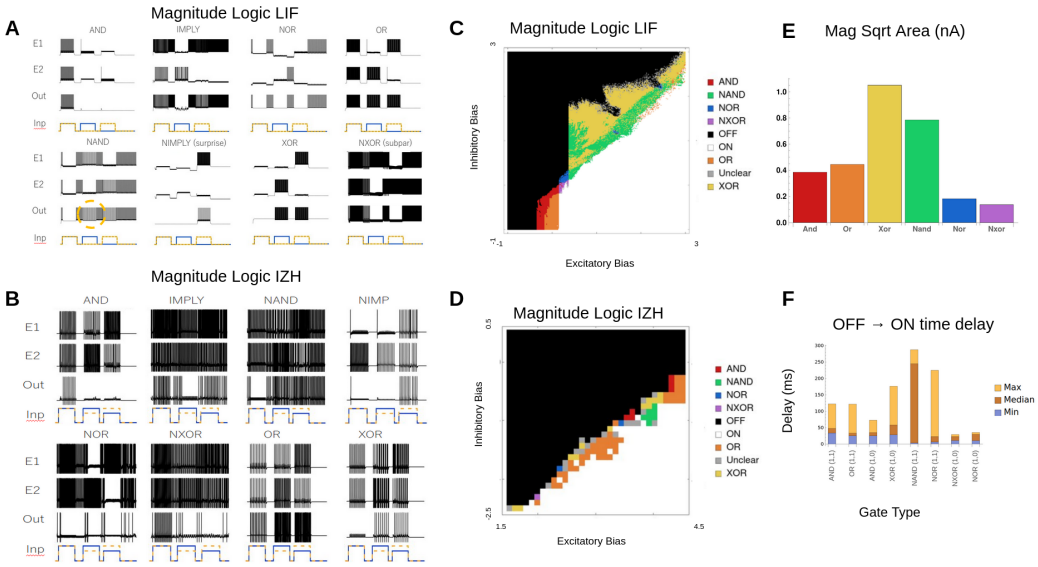


Figure 3: All Magnitude gates. A) All 8 Magnitude Logic gates for the LIF version of the CRIREL. Both IMP and NIMP have a mirror image that is not shown. B) All 8 Magnitude Logic gates for the IZH version of the CRIREL. Both IMP and NIMP have a mirror image that is not shown. C) A sweep through the inhibitory bias and excitatory bias for the LIF model. Each gate is labeled based on its color. Left has a output bias of 1 nA while right has an output bias of 0 nA. D) A sweep through the inhibitory bias and excitatory bias for the IZH model. Each gate is labeled based on its color. E) The robustness of each gate in terms of bias current. It is calculated as the square root of the area for each gate in figure C. Larger values represent more robust gates. F) The average time from a silent OFF state for the output neuron to the first spike of an ON state for the output neuron. We sampled different biases from the sweep in C.

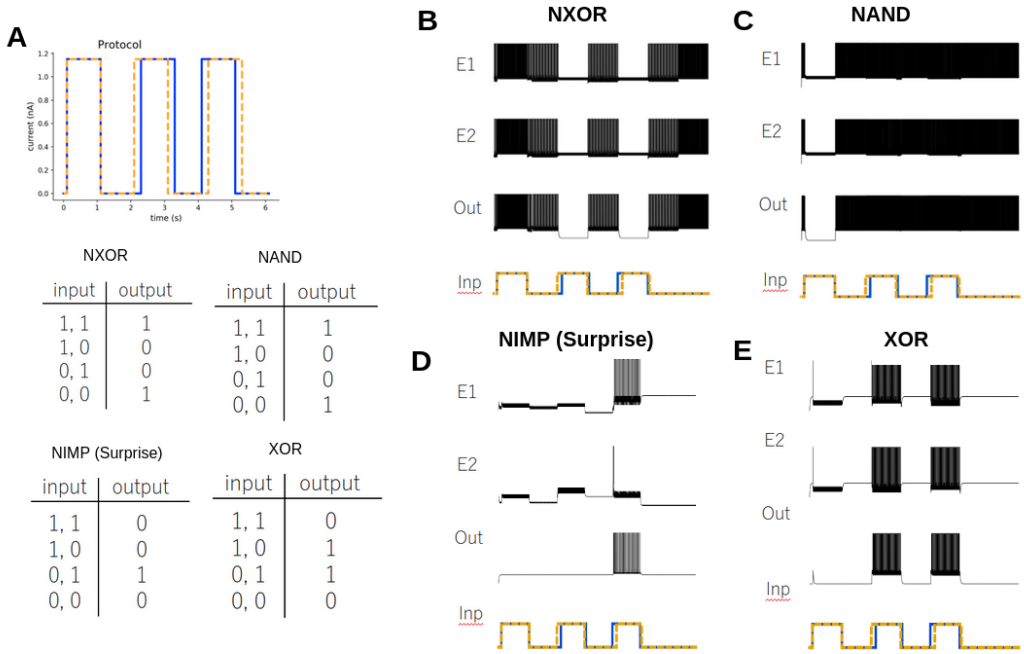


Figure 4: Temporal Logic. A) Protocol for the Temporal Logic Gate. The difference between the timing of the square pulses determines the $(1,1)$, $(1,0)$, $(0,1)$ and $(0,0)$. The difference in timing is 10 ms. The difference between signals is exaggerated for clarity. The Truth tables for XNOR and NAND are shown. B) The LIF version of an XNOR gate. The activity of the excitatory neurons (E1 & E2) and the output neuron (Out) are shown. A cartoon representation of the input protocol is shown. C) The LIF version of a NAND gate. The activity of the excitatory neurons and the output neuron are shown. D) The LIF version of the NIMP gate. This gate is the logical negation of the Imply gate. E) The LIF version of the XOR gate.

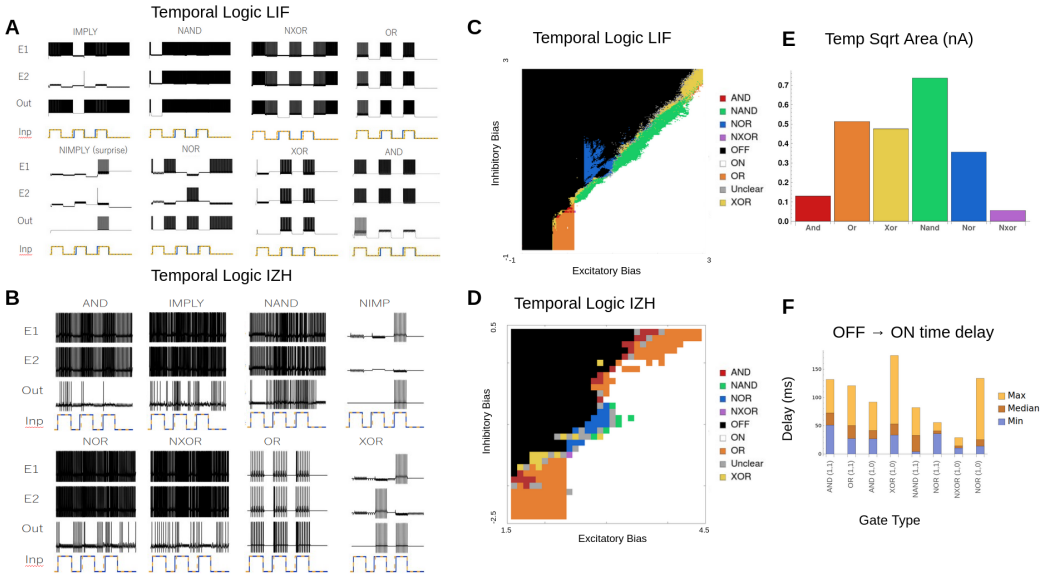


Figure 5: All Temporal gates. A) All 8 Temporal Logic gates for the LIF version of the CRIREL. Both IMP and NIMP have a mirror image that is not shown. B) All 8 Temporal Logic gates for the IZH version of the CRIREL. Both IMP and NIMP have a mirror image that is not shown. C) A sweep through the inhibitory bias and excitatory bias for the LIF model. Each gate is labeled based on its color. Left has an output bias of 1 nA while right has an output bias of 0 nA. D) A sweep through the inhibitory bias and excitatory bias for the IZH model. Each gate is labeled based on its color. E) The robustness of each gate in terms of bias current. It is calculated as the square root of the area for each gate in figure C. Larger values represent more robust gates. F) The average time from a silent OFF state for the output neuron to the first spike of an ON state for the output neuron. We sampled different biases from the sweep in C.

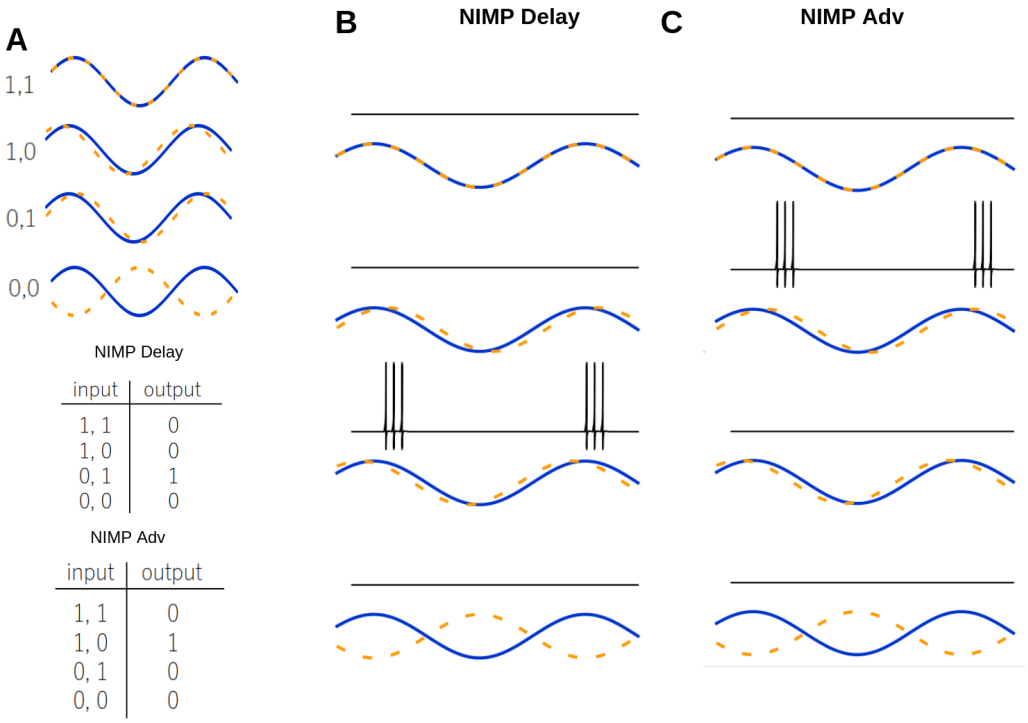


Figure 6: Phase Logic. A) Protocol for the Phase Logic Gate. The difference between the phase of the sine waves determines the $(1,1)$ synchrony, $(1,0)$ phased advanced, $(0,1)$ phase delayed and $(0,0)$ anti-synchronized. The Truth tables for NIMP delayed and NIMP advanced are shown. B) The IZH version of an NIMP delayed gate. The activity of the output neuron is shown. A cartoon representation of the input protocol is shown. C) The IZH version of a NIMP advanced gate. The activity of the excitatory neurons and the output neuron are shown.

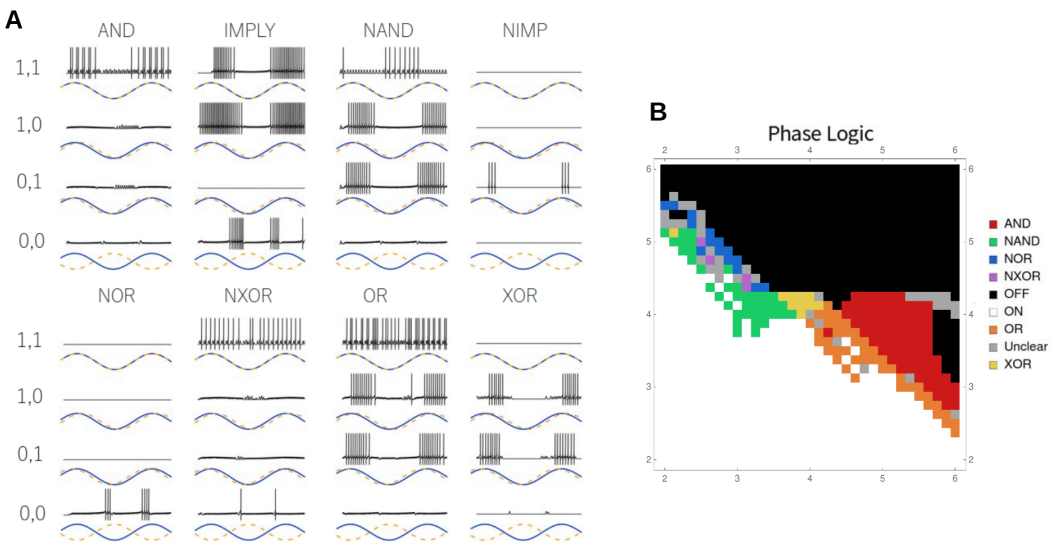
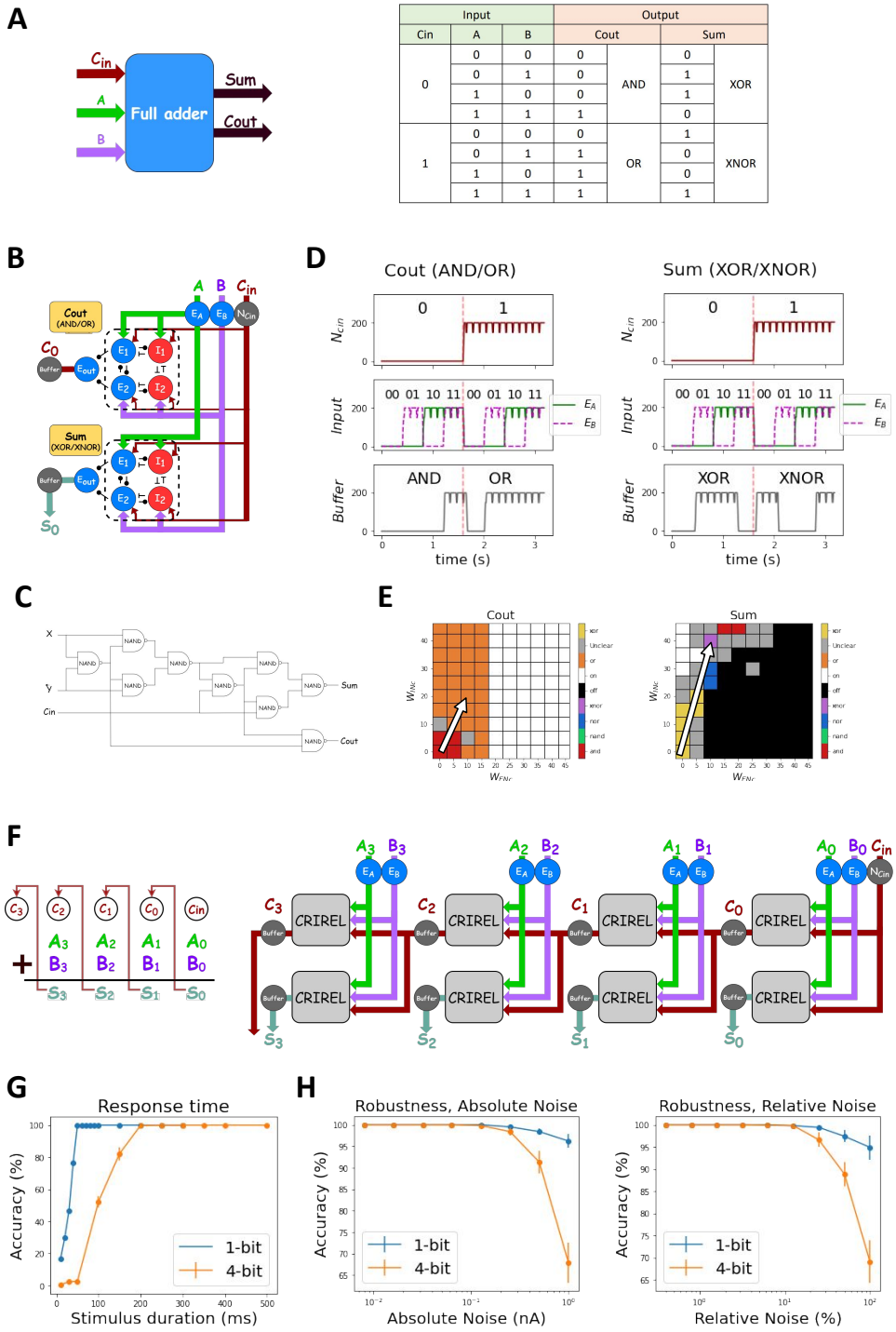


Figure 7: All Phase gates. A) All 8 Magnitude Logic gates for the IZH version of the CRIREL. Both IMP and NIMP have a mirror image that is not shown. B) A sweep through the inhibitory bias and excitatory bias for the IZH model. Each gate is labeled based on its color.



Input			Output			
Cin	A	B	Cout		Sum	
0	0	0	0	AND	0	XOR
	0	1	0		1	
	1	0	0		1	
	1	1	1		0	
1	0	0	0	OR	1	XNOR
	0	1	1		0	
	1	0	1		0	
	1	1	1		1	

Figure 8: An Adder Application for CRIRELs. (A) The schematic and the truth table of full adder.Carry-out (Cout) toggle between AND and OR gates, while the summation (Sum) toggle between XOR and XNOR gates. (B) The networks of 1-bit full adder. (C)A classical 9-gate 1-bit adder, not that it has 7 more required logic gates than the gate switching adder. (D) The output firing rates of two CRIRELs. (E) Parameter space for the control neurons, when the neuron is off the network switches from the starting gate AND and XOR to XNOR and OR. The synaptic weight of the control neuron is tuned such that the resulting constant current changes the bias current to switch to the correct gate. (F) The schematic and network of 4-bit ripple carry adder (G) The response time of adder. A 1-bit adder requires a response time of 50 ms and a 4-bit ripple carry adder needs 200 ms. (H) The adder composed of CRIRELs can tolerate absolute noise of 0.1 nA and relative noise of 10% in its bias.

2 Supplementary Tables

Supplementary Table 1

Parameter for Magnitude Logic Gates(LIF)					
Logic Gate	b_{E_1}	b_{E_2}	b_{I_1}	b_{I_2}	b_{out}
AND	0	0	-0.2	-0.2	0
NAND	0.75	0.75	1.7	1.7	0.5
OR	0.2	0.2	-0.5	-0.5	0.5
NOR	0.65	0.65	2.1	2.1	0.5
XOR	0.3	0.3	1.4	1.4	0.5
XNOR	0.9	0.9	0.5	0.5	-0.7
IMP	1	0.7	0.8	1	0
NIMP	0.2	0	0.85	0.85	0

Table 7

Parameter for Magnitude Logic Gates(IZH)					
Logic Gate	b_{E_1}	b_{E_2}	b_{I_1}	b_{I_2}	b_{out}
AND	1.25	1.25	-1	-1	-3.5
NAND	3.5	3.5	1.0	1.0	-3.0
OR	1.0	1.0	-1.0	-1.0	3.0
NOR	5.05	5.05	2.6	2.6	-5.0
XOR	1.55	1.55	0	0	-2.1
XNOR	3.0	3.0	-1.0	-1.0	-7.25
IMP	3.6	3.7	-.75	-.7	-7.22
NIMP	1.5	0.5	1	0.5	-1.5

Table 8

Supplementary Table 1

Supplementary Table Phase (IZH)

Parameter for Temporal Logic Gates(LIF)					
Logic Gate	b_{E_1}	b_{E_2}	b_{I_1}	b_{I_2}	b_{out}
AND	0	0	0	0	0
NAND	1.5	1.5	1.0	1.0	0
OR	0	0	-0.2	-0.2	0
NOR	0.9	0.9	1.25	1.25	0
XOR	0	0	0.85	0.85	0
XNOR	1.25	1.25	1.15	1.15	0
IMP	1.2	0.7	0.9	1	0
NIMP	0.5	0	0.85	0.85	0

Table 9

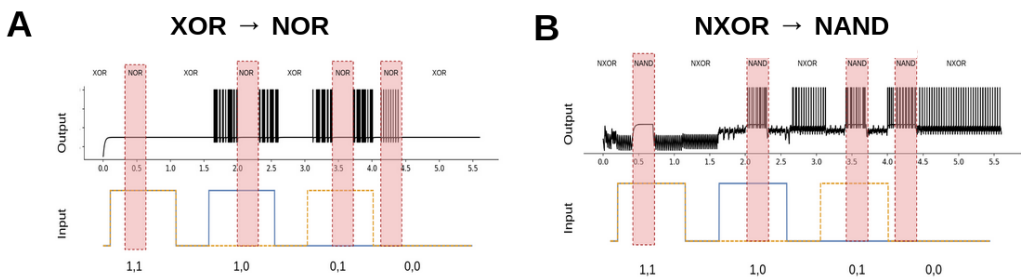
Parameter for Temporal Logic Gates(LIF)					
Logic Gate	b_{E_1}	b_{E_2}	b_{I_1}	b_{I_2}	b_{out}
AND	3.35	3.35	0.35	0.35	-5.45
NAND	3.5	3.5	1.0	1.0	-3.6
OR	1.0	1.0	-1.0	-1.0	3.0
NOR	4.0	4.0	-0.35	0.35	-7.65
XOR	0.5	0.5	2.0	2.0	3.0
XNOR	3.75	3.75	-1.0	-1.0	-8.0
IMP	3.6	3.7	-.75	-.7	-7.22
NIMP	0.5	-0.5	1.0	1.1	2

Table 10

3 supplemental fig

Parameter for Phase Logic Gates (IZH)					
Logic Gate	b_{E_1}	b_{E_2}	b_{I_1}	b_{I_2}	b_{out}
AND	3.5	3.5	3.1	3.1	-2.1
NAND	4	4	3.75	3.75	-2.1
OR	4	4	3.5	3.5	-2.1
NOR	4	4	4.5	4.5	-2
XOR	2	2	2.5	2.5	1.1
XNOR	3.6	3.6	3.3	3.3	-2.05
IMP	2.99	3.13	3.4	3.5	0
NIMP	0	1.5	2.5	2.5	2

Table 11



C

Robust Gate Switching Possible

	AND	IMP	NAND	NIMP	NOR	NXOR	OR	XOR
AND		O	O	O	X	X	O	O
IMP	O		O	O	O	X	O	X
NAND	O	O		O	O	O	O	O
NIMP	O	O	O		O	O	O	O
NOR	X	O	O	O		O	X	O
NXOR	O	O	O	O	O		X	O
OR	O	O	O	O	X	X		O
XOR	O	O	X	O	O	O	O	

Figure 9: sFig1. ate switch test. A) A sample gate change between XOR and NOR. B) A sample gate change between XNOR and NAND. C) A systemetic test for all possible gate changes (row: original gates, column: switched gate). All "O" represent a valid gate. X represents a gate that needs a global reset. All gates use the parameters from sTable **??**. Given the computational complexity, we did not do an exhaustive sweep.

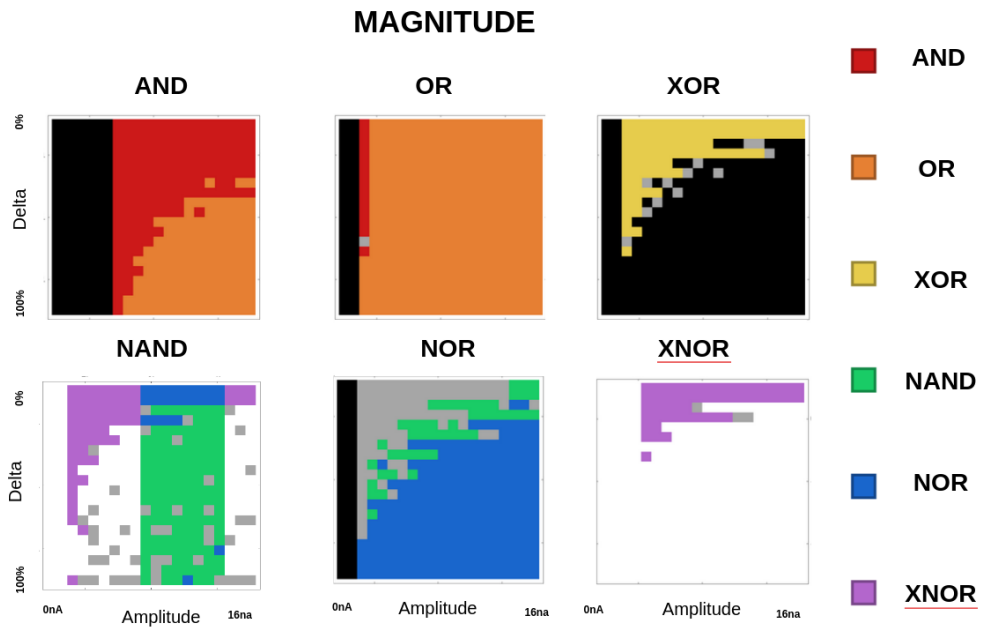


Figure 10: sFig2. A graph of robustness for the 6 symmetric magnitude logic gates as a function of amplitude and the difference in amplitude. Most gates are very robust, except for XNOR.

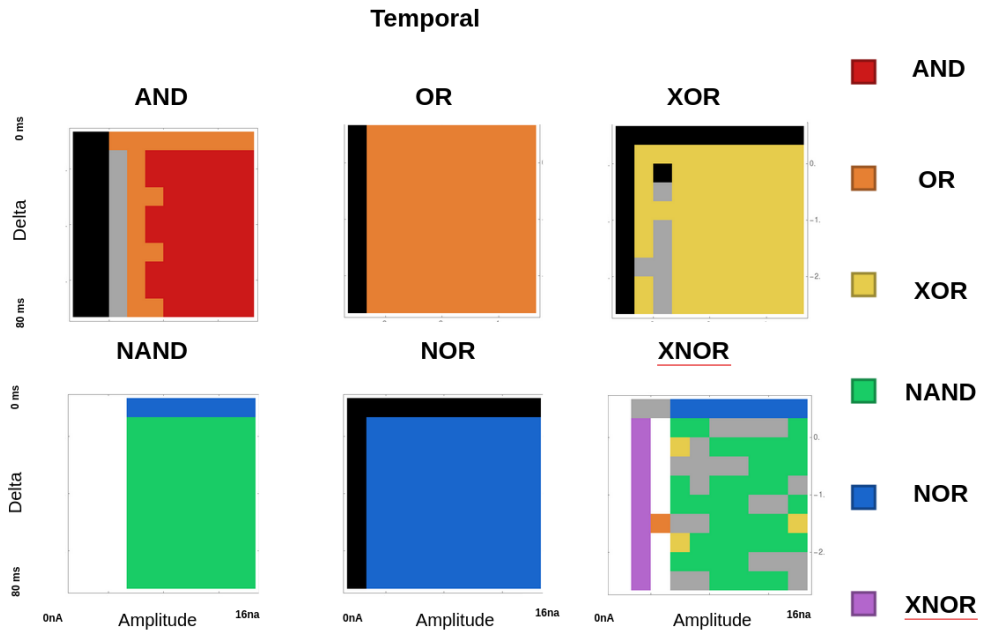


Figure 11: sFig3. A graph of robustness for the 6 symmetric temporal logic gates as a function of amplitude and the timing difference. Most gates are very robust, except for XNOR.

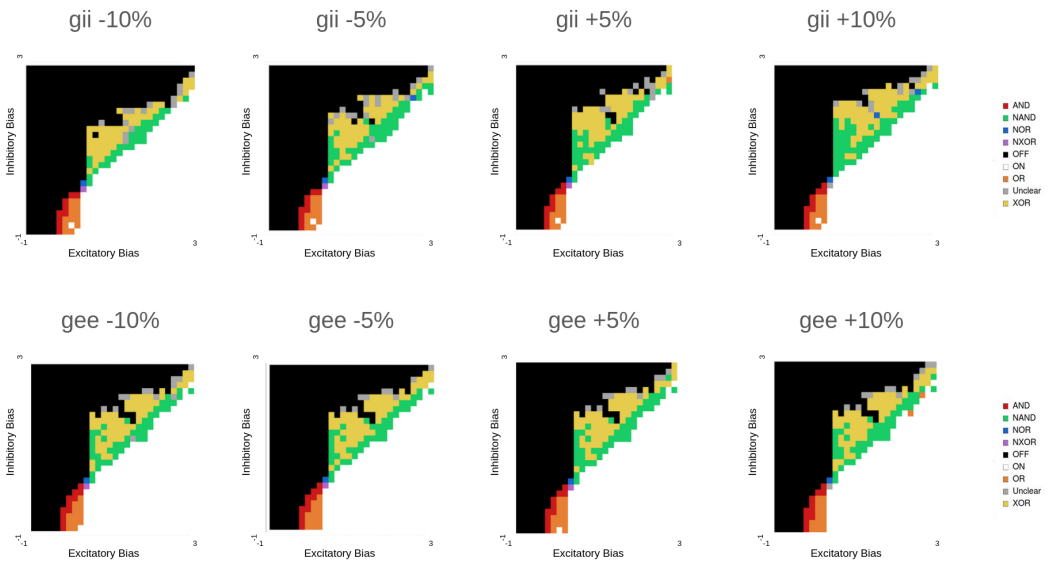


Figure 12: sFig4. Modification of synaptic weights inhibitory to inhibitory (inh) and excitatory to excitatory gee with up to a 10% perturbation both up and down. All gates exist for -10% synaptic weight perturbation. 5 out of 6 gates exist for $+10\%$ synaptic weight perturbation. XNOR is the only exception and exists up to 5% pertubration.