

Oscillatory Threshold Logic

Jon Borresen*, Stephen Lynch

School of Computing, Mathematics and Digital Technology, Manchester Metropolitan University, Manchester, United Kingdom

Abstract

In the 1940s, the first generation of modern computers used vacuum tube oscillators as their principle components, however, with the development of the transistor, such oscillator based computers quickly became obsolete. As the demand for faster and lower power computers continues, transistors are themselves approaching their theoretical limit and emerging technologies must eventually supersede them. With the development of optical oscillators and Josephson junction technology, we are again presented with the possibility of using oscillators as the basic components of computers, and it is possible that the next generation of computers will be composed almost entirely of oscillatory devices. Here, we demonstrate how coupled threshold oscillators may be used to perform binary logic in a manner entirely consistent with modern computer architectures. We describe a variety of computational circuitry and demonstrate working oscillator models of both computation and memory.

Citation: Borresen J, Lynch S (2012) Oscillatory Threshold Logic. PLoS ONE 7(11): e48498. doi:10.1371/journal.pone.0048498

Editor: Teresa Serrano-Gotarredona, National Microelectronics Center, Spain

Received: May 11, 2012; **Accepted:** October 1, 2012; **Published:** November 16, 2012

Copyright: © 2012 Borresen, Lynch. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Funding: The authors have no support or funding to report.

Competing Interests: International patent published 5th January 2012 which underpins the preliminary work. J. Borresen and S. Lynch, International patent no: Binary half-adder using oscillators WO2012/001372 A1 (PDR/CLR/PM328208WO) Pages 1–57. This does not alter the authors' adherence to all the PLOS ONE policies on sharing data and materials.

* E-mail: J.Borresen@mmu.ac.uk

Introduction

In 2005, the US National Security Agency [1] concluded that transistors were rapidly approaching the limits of functionality and that new technologies needed to be developed in order to overcome this.

Recent advances in novel computation suggest a variety of new computing technologies which may be applicable and have far reaching applications. Of particular note are: quantum computing [2]–[6], all-optical computing [7]–[11], spin computing [12]–[15], chaos computing [16]–[18], and DNA computing [19]–[23]. Each of these suggests a technology which is in many respects fundamentally different from current computing systems and in order for their implementation to be successful, new programming structures and techniques would be required, in addition to any development of reliable working prototypes.

One technology which is considerably more advanced from an implementation perspective is superconductive computing [24,25], based on Rapid Single Flux Quantum (RSFQ) technology [26]–[30] which effectively uses Josephson Junctions (JJs) to replace transistors as the fundamental active element in any circuit. This technology is more mature than the others mentioned above and has already produced practical digital and mixed-signal circuits with world record breaking processing speeds at exceptionally low power [31]–[36]. The RSFQ-based digital receivers were demonstrated in the field, converting high-frequency wide-band analog communications signals to digital domain taking advantage of extreme sampling speeds of RSFQ circuits [37]–[41]. The JJ RSFQ circuits are fabricated using a relatively simple thin-film process developed in several places [42]–[46]. Recently, a new energy-efficient generation of RSFQ circuits has been introduced [47] complemented with energy-efficient cryogenic memory [48] leading to the implementation of energy-efficient computing.

Neurons can also be simulated using simple JJ circuits [49]–[53]. These circuits could then be connected to form logic gates similar to those appearing in this paper. The NSA report [1] also concluded that the most likely successor to transistor technology would be JJ circuitry. The authors are currently investigating implementation of binary logic using neuronal JJ circuitry and the results will be published at a later date.

One further technology which is somewhat less advanced from an implementation perspective is neuronal computing (or wetware), which uses artificially grown neurons as the processing units. Again such technology would require a considerable shift in how computers were designed and programmed and is unlikely to prove to be a successor to Complementary Metal-Oxide-Semiconductors (CMOS) at any time in the foreseeable future. What neuronal computing does provide is a deeper understanding of brain functionality and possible associated medical benefits that would therefore follow. Recently, it has been reported that scientists were able to grow brain nerve cells affected by Parkinson's disease using human skin cell samples [54]. These neurons could be connected to form logic circuits similar to those reported in this paper. Degradation of logical functionality could then be used as an assay to determine the effect of drugs or physical damage on neuronal circuitry.

Zanin et al. [55], show that computation can emerge from collective dynamics of an ensemble of networking neurons. Synchronization and desynchronization of neurons using a dynamic weighted network is used to codify binary information. Additionally, neural encoding using conjugate symmetries [56] and the concept of 'winnerless competition' in coupled oscillator networks [57] have also been demonstrated as viable methods of oscillatory computation.

Threshold logic [58,59] and null convention logic [60] have also been considered, particularly with respect to neural network

computing. Beiu et al. [61] provide a review of commercial VLSI hardware implementation of threshold logic up to 2003. With the emergence of nanotechnologies; resonant tunneling, single electron and memristor implementation are also looking promising, especially with respect to energy delay efficiency and reconfigurable circuits [62]–[67]. Additionally, Modified Variable Threshold Logic (MVTTL) using JJs has demonstrated high speed, low power processing [68].

In this paper, we demonstrate a novel computational concept using both inhibitory and excitatory connections between threshold oscillators. The use of inhibitory connections in this way draws on ideas from biological neural encoding and has not previously considered as a tool in implementing binary computation. The use of threshold logic in oscillator circuits presents numerous opportunities for novel circuit design. The inherent richness of oscillator dynamics allows for excitatory and inhibitory connections which may be in many respects different from standard threshold weights used in traditional threshold logic: Neuronal circuits are known to display phase and anti-phase synchronization [69] which may have applications in clocking and error correction. Bursting type behavior [70] could be utilized to prevent signal degradation and improve noise resilience. Connections may induce a variety of responses [71] - all or nothing, additive, amplitude or frequency, which may have specific uses in circuit design.

The method by which inhibition may be caused to occur would be specific to the type of oscillator or oscillatory circuit in question. Neural inhibition has been widely studied [72]. In biological neural circuits, excitation and inhibition occur via diffusion of neurotransmitters across a synaptic gap. For inhibition, this has a temporal effect which permits suppression of post synaptic neural activity for specific time intervals (known as the *refractory period*). In this paper, we use a method of inhibition using negative connection weights similar to that employed in standard threshold logic design, however, this is to demonstrate proof of principle and is not the only possible method by which logic gate connections could be made. Electrical circuit oscillators can likewise demonstrate inhibition [73]. Optical oscillator circuits offer the possibility of interference based inhibition [74] and perhaps most importantly, given recent advances in JJ technology, JJ oscillator circuits have also been designed with inhibitory characteristics very similar to those in neural circuitry [52].

The underlying concept is not technology specific and, given recent advances in JJ circuitry and neuronal computing, could be readily implemented [75,76]. It is also considerably less disruptive in that the underlying binary logic is identical to that employed in CMOS and implementation in the JJ form would be compatible with traditional computing architectures.

We present schematics for simple arithmetic and memory operations and describe these operations as the solution set of a system of linear inequalities. Simulations using a neural oscillator model demonstrate how such a model may be implemented.

Methods

Computing using oscillators is not a new concept, indeed the first modern computers were made using vacuum tube oscillators, and oscillators in a variety of forms are integral components in many devices. The use of neural oscillators has also been widely studied, however, in all cases the method of computation is derived from concepts of biological neural encoding. Current research into encoding using neural oscillators is therefore spatio-temporal, rate, or more usually synchronization based [77–79].

What has not previously been considered is using oscillators as the fundamental components of computing devices (with all the inherent dynamical richness that this provides) and designing them in such a way as to perform binary logic in an equivalent manner to standard transistor logic - that is the oscillator will provide a binary output (1 equivalent to an oscillator firing or 0 where the oscillator does not fire) and the output from a single oscillator can be interpreted in exactly the same way as that of a transistor.

Threshold logic

Threshold logic has been studied as an alternative to Boolean logic for some time. For many implementations this is advantageous, allowing for reduced component counts and/or number of logic levels, as the implementation of complex logical operations may be achieved using a single gate [80].

Threshold logic gates [81] have a set of inputs $\{I_1, I_2, \dots, I_n\}$, weights $\{w_1, w_2, \dots, w_n\}$ and a binary output y . The output y is typically described by:

$$y = \phi \left(\sum_{i=1}^n I_i w_i \right),$$

where the function ϕ is an activation function (eg Heaviside, tanh, sigmoid, piecewise linear, low gain saturation [82]) and the binary output 1 is defined at some threshold $y > T$, say.

Threshold logic implementation has not supplanted standard logic implementation in CMOS due to sensitivity to parameter changes and variable connection weights requiring very low tolerance engineering. Recent advances in nanotechnology, in particular, Resonant Tunneling Devices (RTD) [83] and memristor devices [84] have the potential to overcome such concerns.

Generic threshold oscillator model

A threshold oscillator is an oscillatory device that will begin oscillating when the input to the device is above a certain threshold. Below this level the oscillator remains in a resting state and gives no output. It is possible to use the output of one threshold oscillator as the input of another oscillator to cause the second oscillator to operate (excitation) and under certain circumstances, it is also possible to cause the input of one oscillator to suppress the output of another oscillator (inhibition).

There are numerous viable methods for implementing binary computation using threshold oscillators. In order to perform the logical operations it is necessary that either oscillators with differing thresholds be used or the connections to the oscillators be of differing weights. In our modeling we shall use the latter method as this mimics more closely biological neural systems, from where the idea originated.

Logical operations

Logical operations can be performed in a similar manner to standard logic circuits, however, due to the threshold nature it is possible to formulate logical operations as solutions of sets of linear inequalities. For instance, the AND function can be replicated by a threshold oscillator with two inputs, where the input strengths are scaled such that the total input is only above threshold if both the inputs are on. For a single input or for no input the total input would be below threshold. Defining the inputs to the logical circuits in vector form and scaling the input strength to a binary 1 or 0, we write $\sum I = I_{1,1} + I_{2,1}$ as the total input to the circuit. The threshold equations may be thus written as:

$$\begin{aligned} \text{for } I = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad \sum Iw < T \\ \text{for } I = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad \sum Iw > T, \end{aligned} \tag{1}$$

where T is the oscillator threshold and w the coupling weight between the inputs and the oscillator performing the AND operation. Clearly the solution to the above system Eq. (1) is $\frac{T}{2} < w < T$. For the logical OR operation, the solution $w > T$ would suffice.

Using threshold oscillators in this manner it is straightforward to implement the logical NOT operation using a negative coupling strength, however, as the logical NOT is effectively redundant in more complex logically complete circuit design where NAND and XOR operations are used, we will present all models using the latter formulations.

Binary half adder

One of the simplest computing circuits is the binary half adder. The binary half adder gives the sum of two binary inputs as a two bit binary output. The truth table for the binary half adder is given in Fig. 1A.

Standard transistor implementation of a binary half adder uses one XOR gate (to give the sum) and one AND gate (to give the carry). Implementation of this circuit using threshold oscillators can be achieved via a similar design, with two oscillators replicating the logical functions. The AND operation is implemented as described above and the XOR operation can be achieved using an OR operation (as above) with an additional connection from the AND oscillator, which in some way inhibits the operation of the OR oscillator if the AND oscillator is active. The method by which inhibition occurs would be dependent upon the oscillators being used to form the circuitry.

Fig. 1B demonstrates a viable circuit schematic for half adder implementation using two oscillators O_1 and O_2 and two inputs I_1 and I_2 , which may themselves be the output from other oscillators in a more complex circuit. Schematically, the circuit design is not dissimilar to standard threshold logic half adders [58], however, due to the nature of the connections between oscillators, implementation may be markedly different. If we consider oscillators with identical thresholds we will require that the coupling strength, w_1 , say, from I_1 and I_2 to O_1 be sufficient to cause O_1 to oscillate for only one input and for the coupling strength, w_2 , say, from I_1 and I_2 to O_2 to

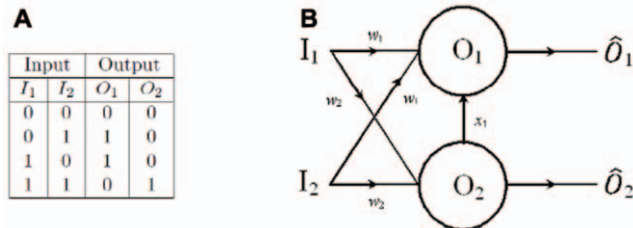


Figure 1. Binary half adder. (A) Truth table for a binary half adder. (B) Oscillator circuit diagram for a binary half adder comprising two inputs I_1 and I_2 and two oscillators O_1 and O_2 . The sum oscillator O_1 will oscillate if either I_1 or I_2 are active. The carry oscillator O_2 will oscillate if both I_1 and I_2 are active. An inhibitory connection from O_2 to O_1 suppresses oscillator O_1 if O_2 is active. doi:10.1371/journal.pone.0048498.g001

be sufficient for it to oscillate for two inputs. The additional connection x_1 , say, from O_2 to O_1 is inhibitory such that if O_2 is oscillating it suppresses O_1 . Denoting the output from O_2 as \hat{O}_2 , the total input to O_1 and O_2 are thus given by:

$$\begin{aligned} O_1 &= \sum Iw_1 - \hat{O}_2x_1 \\ O_2 &= \sum Iw_2. \end{aligned} \tag{2}$$

We can consider such a system as a set of linear inequalities with normalized input vectors I and threshold T requiring solutions of the form:

$$\begin{aligned} \text{for } I = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad & \begin{cases} \sum Iw_1 - \hat{O}_2x_1 < T \\ \sum Iw_2 < T \end{cases} \\ \text{for } I = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad & \begin{cases} \sum Iw_1 - \hat{O}_2x_1 > T \\ \sum Iw_2 < T \end{cases} \\ \text{for } I = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad & \begin{cases} \sum Iw_1 - \hat{O}_2x_1 < T \\ \sum Iw_2 > T. \end{cases} \end{aligned} \tag{3}$$

Thus, for instance, for a total input of $\sum I = 1$, only O_1 will be above threshold causing oscillation giving a binary equivalent output of 1. If both I_1 and I_2 are active, O_2 will oscillate but O_1 is suppressed if $\hat{O}_2x_1 > T/2 + w_1$, giving a binary output $1 + 1 = 10$, as required.

Two-oscillator full adder

In order to demonstrate how oscillatory threshold logic scales for operations on larger numbers of bits we will consider the next simplest arithmetic circuit - the binary adder (or full adder). The binary adder has a truth table as in Fig. 2A.

Using traditional circuitry the binary adder requires five logic gates to perform such an operation. It is possible to formulate the adder circuit using oscillators as a solution set of two linear inequalities and as such only two oscillators are required to perform the operation. The oscillator equations are the same as Eq. (??) and the threshold inequalities are given by:

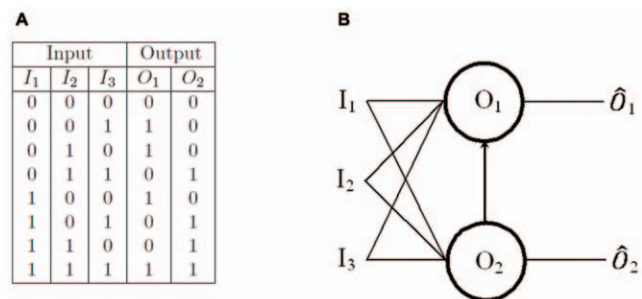


Figure 2. Binary full adder. (A) Truth table for a binary full adder. (B) Oscillator circuit diagram for a binary full adder comprising three inputs I_1, I_2 and I_3 and two oscillators O_1 and O_2 . Oscillator O_1 will oscillate if either I_1, I_2 or I_3 are active. Oscillator O_2 will oscillate if any two of I_1, I_2 and I_3 are active. An inhibitory connection from O_2 to O_1 suppresses oscillator O_1 if O_2 is active, however, the inhibition is only sufficient to suppress O_1 for $\sum I = 2$. For inputs of $\sum I = 3$ the total input to O_1 is still sufficient to induce oscillation. doi:10.1371/journal.pone.0048498.g002

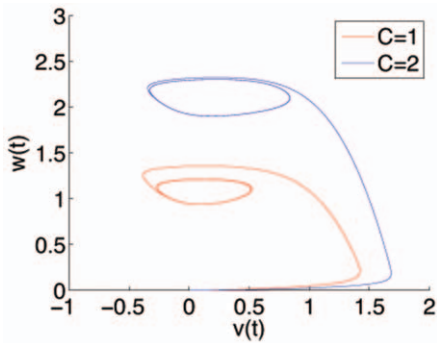


Figure 3. Phase portrait for Fitzhugh-Nagumo model with $\alpha=0.1, \epsilon=0.1$ and $\gamma=0.1$ shown for two input strengths $C=1$ and $C=2$. From an initial resting state at $(0,0)$, the transient dynamics is of a wide trajectory before settling onto a limit cycle. doi:10.1371/journal.pone.0048498.g003

$$\begin{aligned}
 &\text{for } I = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad \begin{cases} \sum Iw_1 - \hat{O}_2x_1 < T \\ \sum Iw_2 < T \end{cases} \\
 &\text{for } I = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad \begin{cases} \sum Iw_1 - \hat{O}_2x_1 > T \\ \sum Iw_2 < T \end{cases} \\
 &\text{for } I = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \quad \begin{cases} \sum Iw_1 - \hat{O}_2x_1 < T \\ \sum Iw_2 > T \end{cases} \\
 &\text{for } I = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \quad \begin{cases} \sum Iw_1 - \hat{O}_2x_1 > T \\ \sum Iw_2 > T \end{cases}
 \end{aligned} \tag{4}$$

In keeping with other full adder threshold logic designs [58,85], the full adder schematic (Fig. 2B) is essentially identical to those of the half adder except for the additional input I_3 .

It is possible to use additional oscillators to give the the binary sum for $2^n - 1$ inputs using only n oscillators. Each additional oscillator acts as the next binary digit of the required output and

the respective input weights are adjusted thusly. Inhibitory connections from the additional oscillator to all other oscillators are required such that the inhibition strength from each new oscillator is scaled accordingly. Thus an exponential increase in computational power could (theoretically) be provided by a linear increase in the number of fundamental components. For larger circuits, the number of interconnections within the circuit would increase considerably, as the computation is effectively encoded in the connections rather than the switches themselves. Given sufficient engineering capability to provide the necessary connectivity, the number of components in any circuit and the time required to perform calculations - as a function of the required switching time of the oscillators - could be considerably reduced in this way.

Results

The Fitzhugh-Nagumo model

The Fitzhugh-Nagumo system [86,87] is one of the more well known oscillator models. It is essentially a reduction of the Hodgkin-Huxley equations [88] which describe the action potential of a spiking neuron. The model is fairly straightforward to implement as an electrical circuit, as demonstrated by Binczak et al. [73].

The describing equations are:

$$\begin{aligned}
 \frac{dv}{dt} &= C + v(v - \alpha)(1 - v) - w \\
 \frac{dw}{dt} &= \epsilon(v - \gamma w),
 \end{aligned} \tag{5}$$

where v is a fast variable (in biological terms - the action potential) and w represents a slow variable (biologically - the sodium gating variable). The parameters α , γ and ϵ dictate the threshold, oscillatory frequency and the location of the fixed points for v and w . The model will begin to oscillate when the input current C is above a critical threshold C_c . For all the following simulations, the threshold $T \approx 0.5$.

It is possible to couple the oscillators together via various methods. For biological neural systems, where there is synaptic coupling between neurons the coupling function is complex, relying on diffusion of neurotransmitters across a synaptic gap. The connections between neurons may either depolarize (excite) or hyperpolarize (inhibit) the post synaptic neuron.

Crucially, the hyperpolarizing inhibitory effect has a temporal component such that if inhibition occurs, the post synaptic neuron remains inhibited for some period of time after the pre-synaptic

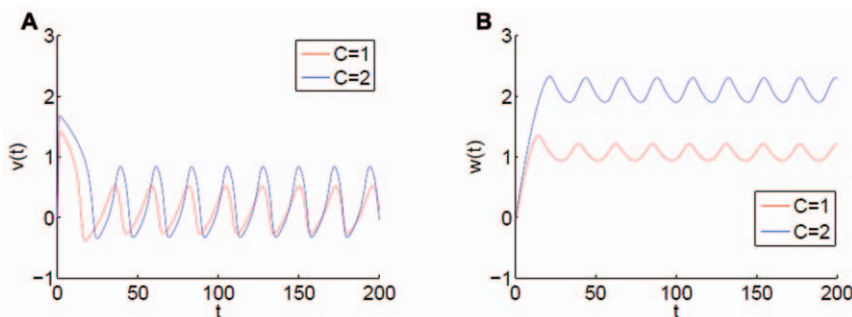


Figure 4. Associated time series of v and w for corresponding Fig. 3. Note that after transients, (A) $v(t) \in [-1,1]$ for $C=1$ and $C=2$, (B) whereas $w(t) \approx C$. doi:10.1371/journal.pone.0048498.g004

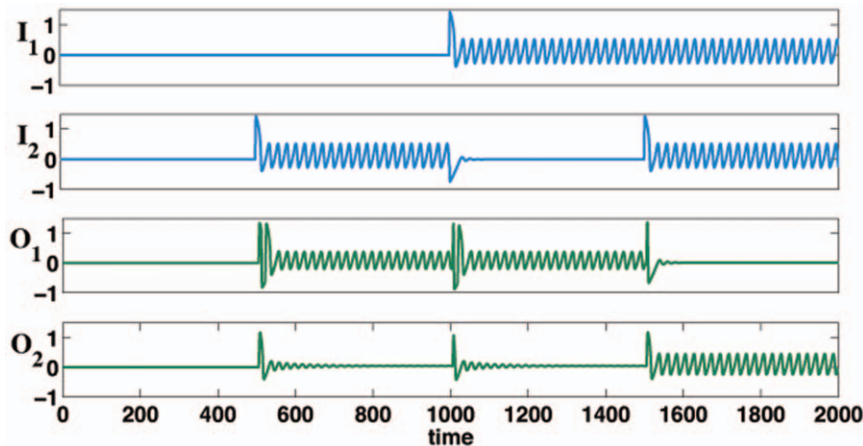


Figure 5. Time series for Fitzhugh-Nagumo binary half adder. The parameter values taken for the transfer function $S(x)$, see Eq. (5), are $m=100$ and $c=0.6$. All binary combinations of oscillatory inputs I_1 and I_2 give the required binary outputs O_1 and O_2 (see Fig. 1A). Here O_1 represents the sum and O_2 the carry in terms of standard logical circuitry. The observed pulse like behavior after switching is due to the wide trajectories taken by the input oscillators after being perturbed from the resting state.
doi:10.1371/journal.pone.0048498.g005

neuron fires. It is not straightforward to simulate such a system using the Fitzhugh-Nagumo model without either integration of the signal pulse or introducing arbitrary conditions on oscillators receiving an inhibitory pulse - which would not be viable from an implementation perspective. As such we will employ a method which is phenomenologically similar to neural hyperpolarization but is not necessarily consistent with any biological process.

Implementation by coupling through either the fast v variable or the slow w variable are equally viable. As can be seen from Fig. 3 and Fig. 4 for varying inputs C , the fast voltage oscillates with a fairly constant maximum and minimum whilst the slow variable w oscillates around a fixed point at approximately C . Any coupling function to be used must take into account the specific dynamics of whichever variable is used.

As is common in such biologically inspired models we will use a sigmoidal transfer function between oscillators of the form:

$$S(x) = \frac{1}{1 + e^{m(-x+c)}}, \quad (6)$$

where c is the threshold at which the output begins to rise and m denotes the steepness of the curve of the function $S(x)$. In biological systems, neural connections can exhibit plastic responses and become ‘tuned’ (via some Hebbian learning rule [89]) allowing for more reliable excitation and inhibition. Choosing suitable values of m and c would in many respects replicate such a process.

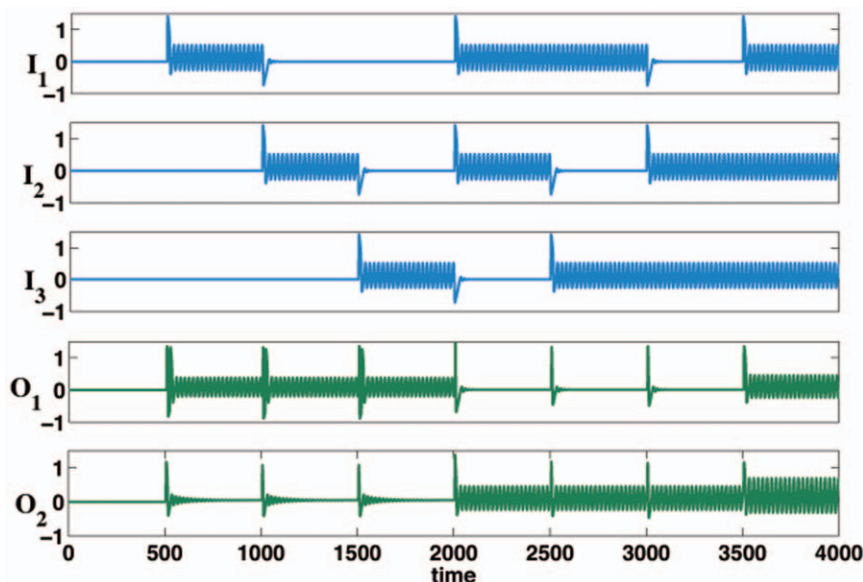


Figure 6. Time series for a Fitzhugh-Nagumo two oscillator full adder. All binary combinations of oscillatory inputs I_1 , I_2 and I_3 give the required binary outputs for O_1 and O_2 (see Fig. 2A). The weights used in this example were again $w_1=0.8$, $w_2=0.4$ and $x_1=1.5$ (see Eq. (4)).
doi:10.1371/journal.pone.0048498.g006

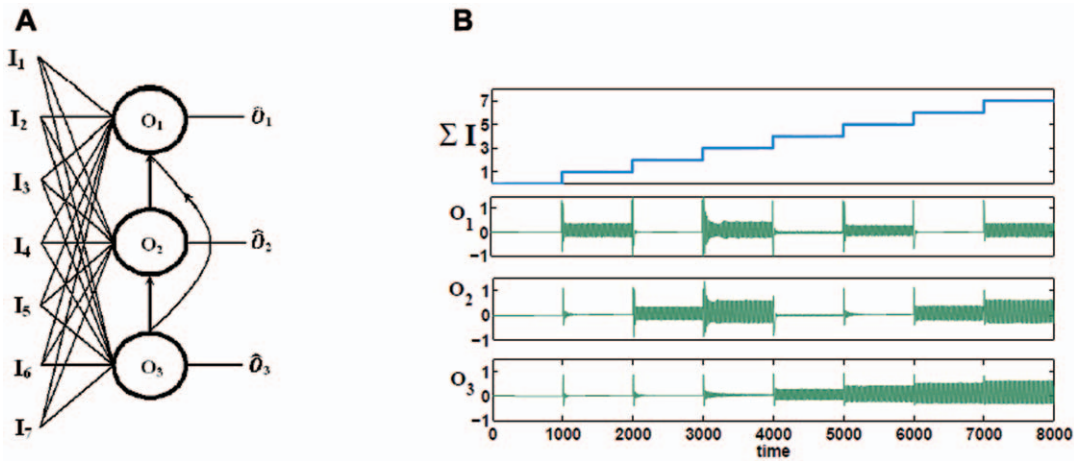


Figure 7. Seven input, three oscillator full adder. (A) Block diagram for seven input, three oscillator full adder. (B) Time series for a Fitzhugh-Nagumo three-oscillator full adder. The total input strength $\sum I$ is shown in the first row. All binary combinations of oscillatory inputs I_1 to I_7 give the required binary outputs for O_1 , O_2 and O_3 . The weights used in this example were $w_1=0.8$, $w_2=0.385$, $w_3=0.175$, $x_1=1.5$, $x_2=1.5$ and $x_3=3.3$, where x_1 is the inhibitory connection from O_2 to O_1 , x_2 is from O_3 to O_2 and x_3 is from O_3 to O_1 . Note that the amplitude of oscillation increases when the sum of the input to any oscillator is greatly above threshold, however, the binary on/off response is as required. doi:10.1371/journal.pone.0048498.g007

Fitzhugh Nagumo half adder and full adders

Numerical simulations for systems of Fitzhugh Nagumo oscillators coupled as in Figs. 1B and 2B will now be discussed. It is also shown that by adding one more oscillator to the two-oscillator full adder it is possible to construct a three-oscillator seven-input full adder.

The inputs to the logical circuits are oscillatory, being provided by Fitzhugh-Nagumo oscillators with similar coupling and parameter values to the computational oscillators. Oscillatory inputs of this form have been chosen over continuous inputs, as this demonstrates the necessary robustness of signal integrity which would be required for larger computational circuits. Continuous inputs to the computational oscillators would be equally viable and present no difficulties in implementation. As such the matrix form for the input weights for each oscillator is 4×4 rather than 2×2 as two additional oscillators are used as inputs.

One solution, in matrix form, to the inequalities Eq. (2) and Eq. (3) for the binary half adder, would be:

$$W = \begin{pmatrix} 0 & 0 & 0.8 & 0.45 \\ 0 & 0 & 0.8 & 0.45 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1.5 & 0 \end{pmatrix}, \tag{7}$$

where $C=0.5$ in Eq. (5) for the inputs I_1 and I_2 . This would give the parameter values in Fig. 1B as $w_1=0.8, w_2=0.45$ and $x_1=1.5$. The time series for such is shown in Fig. 5.

A two-oscillator binary full adder can be constructed by simply introducing another input, I_3 , say, as in Fig. 2B. Fig. 6 shows the time series for the Fitzhugh-Nagumo two-oscillator full adder.

To conclude this section, we consider a three-oscillator seven input full adder (see Fig. 7). Fig. 7A shows a schematic of the three-oscillator seven input adder comprising twenty one excitatory

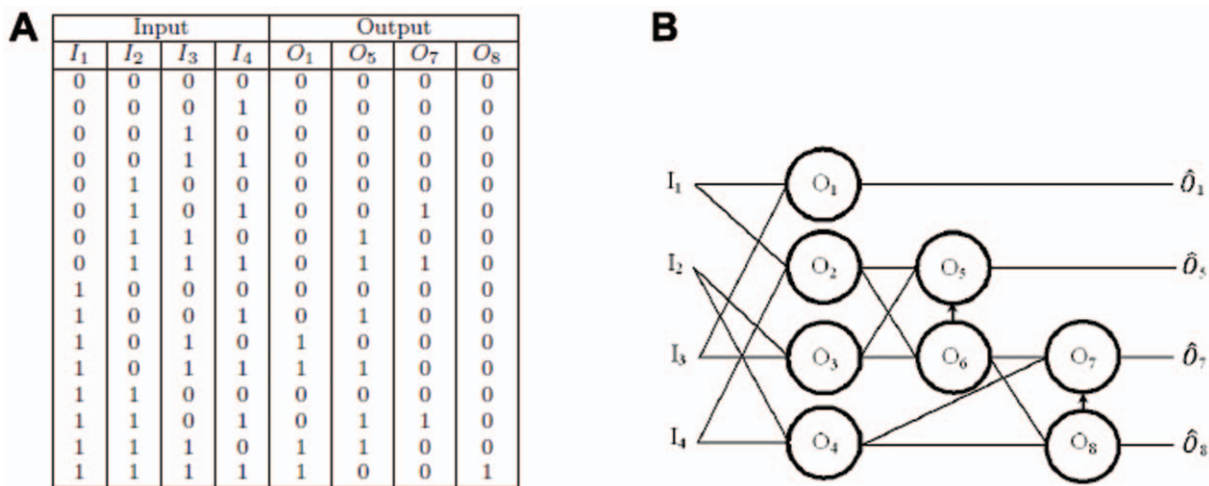


Figure 8. A 2×2 bit multiplier. (A) Truth table for 2×2 bit binary multiplier. (B) Schematic of the binary oscillator 2×2 bit multiplier using oscillators based on standard circuitry. doi:10.1371/journal.pone.0048498.g008

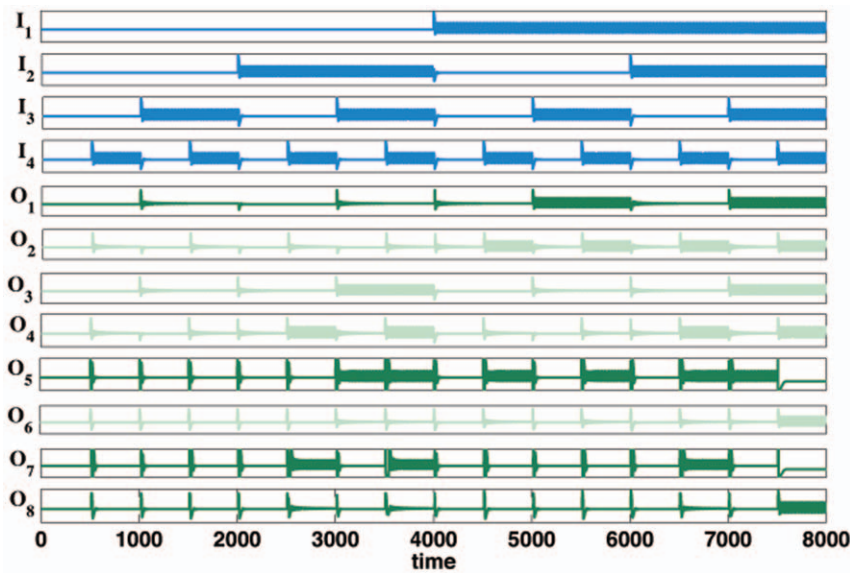


Figure 9. Time series of a 2×2 bit multiplier using oscillators based on standard circuitry. All possible binary combinations for inputs I_1, I_2, I_3 and I_4 are shown. The oscillators O_1, O_5, O_7 and O_8 represent binary digit representation of the powers of 2. Oscillators O_2, O_3, O_4 and O_6 are used in the internal circuitry only and their time traces have been made fainter to emphasize the binary output oscillators. For example, at time $t \in [3500, 4000]$ the inputs $I_1 = 0, I_2 = 1$ representing the decimal 2 and $I_3 = 1, I_4 = 1$, representing the decimal 3, giving an output where O_5 and O_7 are oscillating, being the binary equivalent ($10 + 100 = 110$) of the decimal 6. Note O_1 and O_8 are not oscillating. This is equivalent to column eight above and row eight in the truth table Fig. 8A. doi:10.1371/journal.pone.0048498.g009

connectors and three inhibitory connectors. Fig. 7B shows the time series of the seven input three-oscillator full adder.

Fitzhugh Nagumo 2×2 bit binary multiplier

In order to more fully demonstrate the applicability of such a model we have also simulated a more complex circuit, namely a 2×2 bit binary multiplier. Such a circuit outputs the binary multiple of two, 2 bit binary inputs. Although it is possible to perform such a calculation using only four oscillators we have used a standard circuit implementation of a binary multiplier to demonstrate how oscillators could be used to replace transistors as the fundamental units of computing devices without the need for architectural redesign.

Fig. 8A shows the logic table for a 2×2 bit binary multiplier and Fig. 8B shows the corresponding schematic of the oscillator circuit. The time series of the 2×2 bit binary multiplier is displayed in Fig. 9.

Fitzhugh Nagumo set reset flip-flop

Fig. 10 shows a schematic of a Set-Reset (SR) flip-flop circuit, the input I_1 is commonly referred to as the Set and input I_2 is referred to as the Reset. Output \hat{O}_2 is the complement of output \hat{O}_1 . Note that both oscillators require a constant input I_C , say, for the circuit to function properly. This circuit acts as a memory, storing a bit and presenting it on its output \hat{O}_1 , as can be seen in Fig. 11.

The SR flip-flop described here is an application of the ‘winnerless competition’ principle [57]. In the absence of coupling between the oscillators, both will remain active. However, a symmetric inhibitory coupling between them ensures that from an initial state, where only one oscillator is active, the other will remain suppressed in the absence of any external perturbation. When an input is given to the inactive oscillator this is switched on, simultaneously suppressing the previously active oscillator. When

the external input is turned off, the system remains in the switched state. Note that for a switch to occur, an input pulse of only one period is required (see Fig. 11). Switching using a single pulse in this way can open an opportunity to use ballistic propagation of signals between gates and memory cells, which could significantly reduce the energy required to operate memory circuits, where currently power intensive line charging is required to initiate memory switches.

One important consideration, particularly with respect to flip-flop circuits is the ability to switch accurately in the presence of noise [90]. A detailed examination of error rates arising from such noise would be specific to the oscillator and the underlying circuit implementation. For the general case considered here (a Fitzhugh-Nagumo implementation) repeated simulations have demonstrated that the system is particularly resistant to such noise (see Fig. 12) as

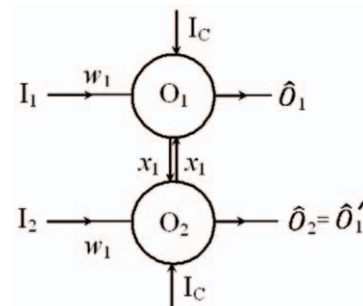


Figure 10. Schematic of an SR flip flop using oscillators. Both oscillators O_1 and O_2 receive an external input current I_c and there are inhibitory connections x_1 between the oscillators. From an initial state where one oscillator is active the other will remain suppressed. External inputs I_1 or I_2 to the inactive oscillator will induce a switch. The oscillators are effectively performing NOR logical operations as in a conventional SR NOR latch. doi:10.1371/journal.pone.0048498.g010

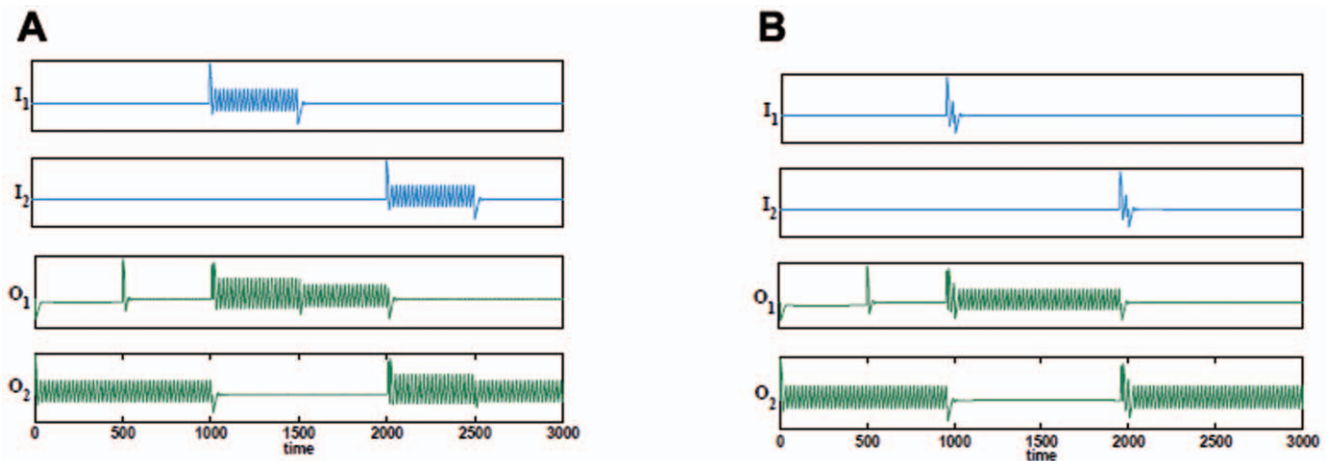


Figure 11. Time series of an SR flip-flop using oscillations to switch. (A) Simulation with $w_1 = 0.5$ and $x_1 = -1$. The simulation is initialized using a single external current to O_2 for $t \in [0, 500]$. At $t = 500$, oscillator O_1 also receives an external input current, however, it is suppressed by the output from O_2 . The initial state has O_2 active and O_1 inactive. A continuous switching pulse is provided by I_1 at $t = 1000$. At $t = 1500$, this switching pulse is turned off, but O_1 and O_2 remain in the switched state (as required). A further switch is performed at $t = 2000$ using a continuous pulse to O_2 . (B) Time series of an SR flip-flop using single input pulses to switch, with $w_1 = 0.5$ and $x_1 = -1$. The switching is performed as for case A, however, only one pulse cycle (ballistic propagation) is used. Note the switching occurs as required and the system remains switched once the pulse has been received.

doi:10.1371/journal.pone.0048498.g011

is often the case with coupled oscillator dynamics [91,92]. As the oscillators in the flip-flop circuit will only 'switch' if provided with a pulse above threshold, for a noise induced switch to occur (an error) it would be required that the noise amplitude itself be above this threshold. Although such an error is conceivable, no such error has been observed for Gaussian white noise with a standard deviation below 5% of the oscillator threshold. It should also be noted that in any JJ implementation of an oscillatory SR flip-flop, noise is effectively reduced to zero due to the required supercooling.

Discussion

Here we have demonstrated how coupled threshold oscillators may be used as the principle components of the next generation of computers. Such implementation using binary logic is not

disruptive, in that, from a programming and architectural perspective no significant changes are required.

There are clearly considerations concerning the accurate functioning of any such highly connected circuits, particularly with respect to multiple inputs to a single gate [93,94], however, it should not be overly problematic to reduce the component numbers significantly given modern engineering capabilities.

The Fitzhugh-Nagumo simulations of the half adder, two-oscillator full adder, three-oscillator full adder, and 2×2 bit multiplier demonstrate threshold oscillators performing all the necessary components of arithmetic logic, while the SR flip-flop demonstrates the potential for very low power memory, particularly when ballistic propagation is considered.

Although the Fitzhugh-Nagumo models demonstrated here are in many ways phenomenologically similar to neural dynamics, we are not attempting to make any inference as to neural dynamics

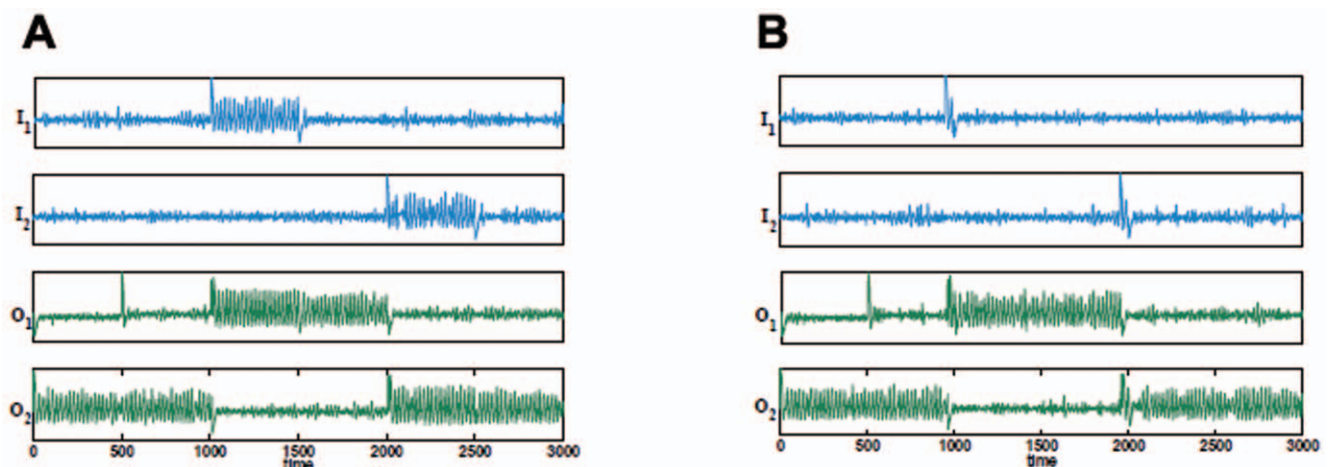


Figure 12. Controlled switching in the presence of noise. (A) and (B) Time series of SR flip-flop as in Fig. 11 with additional Gaussian noise, mean=0, standard deviation=0.05, added to all oscillators. In both cases the switching occurs as required.

doi:10.1371/journal.pone.0048498.g012

themselves. Moreover, we use the Fitzhugh-Nagumo system as an exemplification of the idea. In practice, implementation via Fitzhugh-Nagumo circuitry would be unlikely as this would offer very little in terms of speed or power consumption, however, any implementation via JJs or optical oscillators could be achieved in a very similar manner to that described for the Fitzhugh-Nagumo model whilst providing exceptional processing speed for minimal power usage.

There is currently a drive to low power exascale supercomputing (a computer which performs more than 10^{18} floating point operations per second (FLOPS)). The previous world's fastest supercomputer, the K computer, operates at a maximum performance of 10.51×10^{15} FLOPS and requires approximately 12 MW of power (excluding the power for the cooling system which is typically in excess of 25% of power). This has recently been overtaken by the Sequoia IBM BlueGene/Q operating at 16.32×10^{15} FLOPS and using 7.89 MW [95]. Even with such continued improvements in power consumption it is clear that

References

- National Security Agency (2005) Superconducting Technology Assessment report. Available: www.nitrd.gov/PUBS/nsa/sta.pdf. Accessed 2012 June 27.
- Deutsch D (1985) Quantum-theory, the Church-Turing principle and the universal quantum computer. *Proc Roy Soc A* 400: 97–117.
- Lloyd S (1993) A potentially realizable quantum computer. *Science* 261: 1569–1571.
- Majer J, Chow JM, Gambetta JM, Koch J, Johnson BR, et al. (2007) Coupling superconducting qubits via a cavity bus. *Nature* 449: 443–447.
- Morton JLL, McCamey DR, Eriksson MA, Lyon SA (2011) Embracing the quantum limit in silicon computing. *Nature* 479: 345–353.
- Weber B, Mahapatra S, Ryu H, Lee S, Fuhrer A, et al. (2012) Ohm's law survives to the atomic scale. *Science* 335: 64–67.
- Szöke A, Daneu V, Goldhar J, Kirit NA (1969) Bistable optical element and its applications. *Appl Phys Lett* 15: 376.
- Smith SD (1984) Towards the optical computer. *Nature* 307: 315–316.
- Arsenault A, Fournier-Bidoz SB, Hutton B, Miguez H, Tétraud N, et al. (2004) Towards the synthetic all-optical computer: science fiction or reality? *Journal of Materials Chemistry* 14: 781–794.
- Pahari N, Das DN, Mukhopadhyay S (2004) All optical methods for the addition of binary data by nonlinear materials. *Applied Optics* 43: 6147–6150.
- Lynch S, Steele AL (2011) Nonlinear optical fibre resonators with applications in electrical engineering and computing, Applications of Nonlinear Dynamics and Chaos in Engineering. Santo Banerjee, Mala Mitra, Lamberto Rondoni (Eds.), Springer 1: 65–84.
- Julliere M (1975) Tunneling between ferromagnetic-films. *Phys Lett A* 54: 225–226.
- Datta S, Das B (1990) Electronic analog of the electro-optic modulator. *Appl Phys Lett* 56: 665–667.
- Meservey R, Tedrow PM (1994) Spin-polarized electron tunneling. *Physics Reports* 238: 173–243.
- Dery H, Dalal P, Cywinski L, Sham LJ (2007) Spin-based logic in semiconductors for reconfigurable large-scale circuits. *Nature* 447: 573–576.
- Kia B, Dari A, Ditto WL, Spano ML (2011) Unstable periodic orbits and noise in chaos computing. *Chaos* 21: 047520.
- Kia B, Spano ML, Ditto WL (2011) Chaos computing in terms of periodic orbits. *Phys Rev E* 84: 036207.
- Zanin M, Papo D, Sendina-Nadal I, Boccaletti S (2011) Computation as an emergent feature of adaptive synchronization. *Phys Rev E* 84: 060102.
- Adleman LM (1994) Molecular computation of solutions to combinatorial problems. *Science* 266:1021–1024.
- Boneh D, Dunworth C, Lipton RJ, Jiri S (1996) On the computational power of DNA. *Discrete Applied Mathematics* 71: 79–94.
- Amos M, Paun G, Rozenberg G, Salomaa A (2002) Topics in the theory of DNA computing. *Theoretical Computer Science* 287:1, 3–38.
- Kari L, Gloor G, Yu S (2000) Using DNA to solve the Bounded Post Correspondence Problem. *Theoretical Computer Science* 231: 192–203.
- Xu J, Tan GJ (2007) A review on DNA computing models. *Journal of Computational and Theoretical Nanoscience* 4: 1219–1230.
- Dorojevets M, Bunyk P, Zonoviev D, Likharev K (1999) COOL-0: Design of an RSFQ subsystem for petaflops computing. *IEEE Trans Appl Supercond* 9: 3606–3614.
- Takagi N, Murakami K, Fujimaki A, Yoshikawa N, Inoue K, et al. (2008) Proposal of a desk-side supercomputer with reconfigurable data-paths using Rapid Single-Flux-Quantum circuits. *IEICE Trans Electron E91-C*: 350–355.
- Mukhanov OA, Semenov VK, Likharev K (1987) Ultimate performance of the RSFQ Logic Circuits. *IEEE Trans Mag* 23: 759–762.
- Kaplunenko VK, Khabipov MI, Koshelets VP, Likharev KK, Mukhanov OA, et al. (1989) Experimental study of the RSFQ logic elements. *IEEE Trans Mag* 25: 861–864.
- Likharev K, Semenov V (1991) RSFQ logic/memory family: A new Josephson-junction technology for sub-terahertz clock-frequency digital systems. *IEEE Trans Appl Supercond* 1: 3–28.
- Mukhanov OA (1994) Superconductive single flux quantum technology. *Solid-State Circuits Conference, Digest of Technical Papers*. 126–127, doi: 10.1109/ISSCC.1994.344703
- Bunyk P, Likharev K, Zonoviev D (2001) RSFQ technology: Physics and devices. *International Journal of High Speed Electronics and Systems* 11: 257–306.
- Mukhanov OA, Rylov SV (1997) Time-to-Digital converters based on RSFQ digital counters. *IEEE Trans Appl Supercond* 7: 2669–2672.
- Kirichenko AF, Sarwana S, Gupta D, Rochwarger I, Mukhanov O (2003) Multi-channel time digitizing system. *IEEE Trans Appl Supercond* 13: 454–458.
- Mukhanov O, Gupta D, Kadin AM, Semenov VK (2004) Superconductor analog-to-digital converters. *Proc of the IEEE* 92: 1564–1584.
- Filippov T, Dorojevets M, Sahu A, Kirichenko A, Ayala C, et al. (2011) 8-bit asynchronous wave-pipelined RSFQ arithmetic-logic unit, *IEEE Trans on Applied Superconductivity* 21: 847–851.
- Filippov T, Sahu A, Kirichenko AF, Vernik IV, Dorojevets M, et al. (2012) 20 GHz operation of an asynchronous wave-pipelined RSFQ arithmetic-logic unit, *Physics Procedia* 36:59–65.
- Ortlepp T, Wetzstein O, Engert S, Kunert J, Toepfer H (2011) Reduced power consumption in superconducting electronics, *IEEE Trans. on Applied Superconductivity* 21: 770–775.
- De Escobar LA, Mukhanov O, Hitt R, Littlefield W (2006) High performance HF-UHF all digital RF receiver tested at 20 GHz clock frequencies. *Military Communications Conference, MILCOM 2006*. IEEE, 1–6, doi: 10.1109/MILCOM.2006.302479.
- Vernik V, Kirichenko D, Dotsenko VV, Miller R, Webber RJ, et al. (2007) Cryocooled wideband digital channelizing RF receiver based on low-pass ADC. *Superconductor Science and Technology* 20: S323–S327.
- Wong J, Dunneagan R, Deepnarayan G, Kirichenko D, Dotsenko V, et al. (2007) High performance, all digital RF receiver tested at 7.5 GigaHertz. *Military Communications Conference, MILCOM 2007*. IEEE, 1–5. doi: 10.1109/MILCOM.2007.4455052.
- Mukhanov OA, Kirichenko D, Vernik IV, Filippov TV, Kirichenko A, et al. (2008) Superconductor digital-RF receiver systems, *IEICE Trans Electron E91-C*: 306–317.
- Littlefield W, Mukhanov O, Gupta D, Hitt D (2009) Ultra high speed ADCs and DSP brings direct digital RF beam forming to MILSATCOM phased array apertures. *Military Communications Conference, 2009. MILCOM 2009*. IEEE, 1–7, doi: 10.1109/MILCOM.2009.5379946.
- Brock DK, Kadin AM, Kirichenko AF, Mukhanov OA, Sarwana S, et al. (2001) Retargeting RSFQ cells to a submicron fabrication process. *IEEE Trans Appl Supercond*, 11: 369–372.
- Tolpygo SK, Yohannes D, Hunt RT, Vivalda JA, Donnelly D, et al. (2007) 20 kA/cm² process development for superconducting integrated circuits with 80 GHz clock frequency. *IEEE Trans Appl Supercond* 17: 946–951.
- Abelson LA, Kerber GL (2004) Superconductor integrated, circuit fabrication technology, *Proc of the IEEE* 92: 1517–1533.
- Satoh T, Hinode K, Akaike H, Nagasawa S, Kitagawa Y, et al. (2005) Fabrication process of planarized multi-layer Nb integrated circuits. *IEEE Trans Appl Supercond* 15: 78–81.

46. Andres S, Blamire MG, Buchholzet FI, Crété DG, Cristiano R, et al. (2010) European roadmap on superconductive electronics - status and perspectives. *Physica C* 470: 2079–2126.
47. Mukhanov O (2011) Energy-efficient Single Flux Quantum technology. *IEEE Trans Appl Super-cond.* 21: 760–769.
48. Ryazanov VV, Bol'ginov VV, Sobanin DS, Vernik IV, Tolpygo SK, et al. (2012) Magnetic Josephson junction technology for digital and memory applications, *Physics Procedia* 36: 35–41.
49. Mukhanov OA, Semenov VK (1988) Reproduction of the Single Flux Quantum pulses in Josephson junction systems. II. Neuristor lines and logic elements. *Mikroelektronika [Sov. Microelectronics]* 17: 96–102.
50. Mizugaki Y, Nakajima K, Sawada Y, Yamashita T (1994) Implementation of new superconducting neural circuits using coupled SQUIDS. *IEEE Trans Appl Supercond* 4: 1–8.
51. Dana SK, Sengupta DC, Hu CK (2006) Spiking and bursting in Josephson junction, *IEEE Transactions on Circuits and Systems 11-Express Briefs* 10: 1031–1034.
52. Crotty P, Schult D, Segall K (2010) Josephson junction simulation of neurons, *Phys Rev E* 82: 011914.
53. Onomi T, Maenami Y, Nakajima K (2011) Superconducting neural network for solving a combi-natorial optimization problem, *IEEE Trans Appl Supercond* 21: 701–704.
54. Jiang H, Ren Y, Zhong P, Ghaedi M, Hu Z, et al. (2012) Parkin controls dopamine utilization in human midbrain dopaminergic neurons derived from induced pluripotent stem cells, *Nature Communications* 3: 688.
55. Zanin M, Del Pozo F, Boccaletti S (2011) Computation emerges from adaptive synchronization of networking neurons, *PLoS ONE* 11: e26467.
56. Ashwin P, Borresen J (2004) Encoding via conjugate symmetries of slow oscillations for globally coupled oscillators. *Phys Rev E* 70:026203.
57. Afraimovich V, Rabinovich M, Varona P (2004) Heteroclinic contours in neural ensembles and the winnerless competition principle. *Int. Journal of Bifurcation and Chaos* 14: 1195–1208.
58. Muroga S (1971) *Threshold Logic and Its Applications* Wiley.
59. Siu KY, Roychowdhury VP (1994) On optimal depth threshold circuits for multiplication and related problems. *SIAM Journal on Discrete Mathematics* 7: 284–292.
60. Fant KM, Brandt SA (1997) *Null Convention Logic* Theseus Logic Inc.
61. Beiu V, Quintana JM, Avedillo MJ (2003) VLSI implementation of threshold logic: a survey. *IEEE Trans on Neural Networks, Special Issue on Hardware Implementations* 14: 1217–1243.
62. Gosser KF, Pacha C, Kanstein A, Rossmann ML (1997) Aspects of system and circuits for nano-electronics. *Proc IEEE* 85: 558–573.
63. Likharev K (2003) Electronics below 10 nm. in Greer J, Korkin A, Labanowsky J (eds.): *Nano and Giga Challenges in Microelectronics*. Elsevier 27–68.
64. Wei YI, Shen J (2011) Novel universal threshold logic gate based on RTD and its application. *Microelectronics Journal* 42: 851–854.
65. Chua L (1971) Memristor-the missing circuit element. *IEEE Trans Circuit Theory* 18: 507–519.
66. Strukov DB, Snider GS, Stewart DR, Stanley W (2008) The missing memristor found. *Nature* 453: 80–83.
67. Rajendran J, Manem H, Karri R, Rose GS (2012) An energy-efficient memristive threshold logic circuit. *IEEE Trans on Computers* 61: 474–487.
68. Inoue A, Kotani S, Imamura T, Hasuo S (1993) Niobium based Josephson circuit technology. *Superconductivity* 1: 1863–1877.
69. Han SK, Kurrer C, Kuramoto Y (1995) Dephasing and bursting in coupled neural oscillators. *Phys Rev Lett* 75:3190–3193.
70. Hindmarsh JL, Rose RM (1984) A model of neuronal bursting using three coupled first order differential equations. *Proc Roy Soc B* 221:87–102.
71. Boucsein C, Tetzlaff T, Meier R, Aertsen A, Naundorf B (2009) Dynamical response properties of neocortical neuron ensembles: Multiplicative versus additive noise. *J Neuroscience* 29: 1006–1010.
72. Selverston AL, Moulins M (1985) Oscillatory neural networks. *Annual Review of Physiology* 47: 29–48.
73. Binczak S, Jacquir S, Bibault J-M, Kazantsev VB, Nekorkin VI (2006) Experimental study of electrical Fitzhugh-Nagumo neurons with modified excitability. *Neural Networks* 19: 684–693.
74. Kiesel N, Schmid C, Weber U, Ursin R, Weinfurter H (2005) Linear optics controlled-phase gate made simple. *Phys Rev Letts* 95:210505.
75. Borresen J, Lynch S (2012) Binary half-adder using oscillators. *International Patent Number WO 2012/001372 A1 (PDR/CLR/PM328208WO)* 1–57.
76. Borresen J, Lynch S (2012) Binary half-adder and other logic circuits. *UK Patent Number GB 2481717 A, Priority Data 1011110.2*, 1–57.
77. Smeal RM, Ermentrout GB, White JA (2010) Phase response curves and synchronized neural networks. *Phil Trans R Soc B.* 365 (1551): 2402–2407.
78. Barreiro AK, Shea-Brown E, Thilo EL (2010) Time scales of spike-train correlation for neural oscillators with common drive. *Phys Rev E* 81: 011916.
79. Zhang X, Wang R, Zhang Z, Qu J, Cao J, et al. (2010) Dynamic phase synchronization characteristics of variable high-order coupled neuronal oscillator population. *Neurocomputing* 73 (13–15): 2665–2670.
80. Waser R (2012) *Nanoelectronics and Information Technology*. Wiley.
81. McCulloch W, Pitts W (1943) A logical calculus of the ideas imminent in nervous activity. *Bull Math Biophys* 5:115–133.
82. Lynch S (2010) *Dynamical Systems with Applications using Maple*. Springer.
83. Wei Y, Shen J (2011) Novel universal threshold logic gate based on RTD and its application. *Microelectronics Journal* 42: 851–854.
84. Rajendran J, Manem H, Karri R, Rose GS (2012) An energy efficient memristive threshold logic circuit. *IEEE Trans on Computers* 61: 474–487.
85. Current KW (1994) Current-hyphen mode CMOS multiple-valued logic-circuits. *IEEE J Solid-State Circuits* 29: 95–107.
86. Fitzhugh R (1961) Impulses and physiological states in theoretical models of nerve membranes. *Biophys* 1182: 445–466.
87. Nagumo J, Arimoto S, Yoshizawa S (1970) An active pulse transmission line simulating 1214-nerve axons. *Proc. IRL* 50: 2061–2070.
88. Hodgkin AL, Huxley AF (1952) A qualitative description of membrane current and its application to conduction and excitation in nerve. *J Physiol* 117: 500–544. Reproduced in *Bull Math Biol* (1990) 52: 25–71.
89. Hebb DO (1949) *The organization of behaviour: A Neurophysiological Theory*. Wiley.
90. Elgamel MA, Faisal MI, Bayoumi MA (2005) Noise metrics in flip-flop designs. *IEICE Trans Inf & Syst* E88-D 7: 1501–1505.
91. Ashwin P, Orosz G, Wordsworth J, Townley S (2007) Dynamics on networks of cluster states for globally coupled phase oscillators. *SIAM J on Applied dynamical systems* 6:728–758.
92. Ashwin P, Borresen J (2005) Discrete computation using a perturbed heteroclinic network. *Phys Letts A* 347: 208–214.
93. Reischuk R (2000) Can large fanin circuits perform reliable computations in the presence of faults? *Theoretical Computer Science* 240: 319–335.
94. Kuo PY, Wang CY, Huang CY (2011) On rewiring and simplification for canonicity in threshold logic circuits. *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)* 396–403.
95. Top 500 supercomputer sites. Available www.top500.org. Accessed 2012 June 27.