Dynamics Based Computation

Sudeshna Sinha[1,*] and William L. Ditto[2,†]

[1]*C.I.T. Campus, The Institute of Mathematical Sciences, Madras, India 600 113*

[2]*Applied Chaos Laboratory, School of Physics, Georgia Institute of Technology, Atlanta, Georgia 30332*

(Received 20 April 1998)

We demonstrate the ability of lattices of coupled chaotic maps to perform simple computations. This dynamical system is shown to emulate logic gates, encode numbers, and perform specific arithmetic operations on those numbers such as addition and multiplication. We also demonstrate the ability of this dynamical system to perform the more specialized operation of determining the least common multiplier of a sequence of integers. [S0031-9007(98)07021-5]

PACS numbers: 05.45.+b, 89.70.+c

A recurring theme of research into chaotic systems has been that chaos provides "flexibility" in the performance of natural systems and provides such systems with a rich repertoire of behaviors that can be utilized for "improved" performance [1]. We demonstrate the ability of coupled chaotic dynamical systems to perform a variety of computations.

Chaotic lattices.—Our computer hardware is a network of chaotic elements. It is described by discrete space i, discrete time τ, and continuous state variable. The individual elements (indexed by their spatial location i) evolve under a suitable nonlinear map $f(x_\tau(i))$. We have taken f to be the logistic map: $f(x) = ax(1 - x), x \in [0, 1]$ with the nonlinearity parameter a chosen to make the system chaotic ($a = 4$ throughout this work). In this chaotic lattice a *self-regulatory threshold dynamics* is incorporated to provide adaptation [2]. The adaptive mechanism is triggered when a site in the lattice exceeds the critical value x_*, i.e., when a certain site $x_\tau(i) > x_*$. The supercritical element then relaxes (or avalanches) by transporting its excess $\Delta = (x_\tau(i) - x_*)$ to its neighbor(s). In particular, we consider *unidirectional* transport in one-dimensional lattices of coupled logistic maps, which behave as follows: When a response (a relaxation) is triggered, the signal (excess of threshold) is transferred to one neighbor: $x_\tau(i) \rightarrow x_*, x_\tau(i + 1) \rightarrow x_\tau(i + 1) + \Delta$.

The relaxation continues synchronously until all $x(i) \leq x_*$, after which the next iteration of the maps takes place. The dynamics then induces a unidirectional nonlinear transport down the lattice by initiating a domino effect (reminiscent of the avalanches arising in self-organized sandpiles [3]). The boundary (henceforth *edge*) is open so that the excess is conducted out of the system. Our basic unit of time (henceforth a *dynamical update*) consists of one synchronous forward iteration of the maps in the lattice followed by relaxation of all lattice sites to their final (relaxed) state [all $x(i) \leq x_*$].

The threshold coupling governs the dynamics of the lattice, showing the presence of many *phases* in x_* space [2]. The excess emitted from the open boundary of the system, as well as the lattice configurations of N threshold coupled elements $\{x(1), x(2), \ldots, x(N)\}$ for all finite N, evolve in cycles of varying orders with period depending on the value of x_*. For example [2]: For $0 < x_* < 0.75$ we get a spatiotemporal fixed-point region (where all lattice sites relax to x_*, i.e., $x(i) = x_*$ for every element i after threshold coupling). Further, the excess is emitted at the rate of one unit per dynamical update. This is followed by regions of threshold parameter space where the system emits excess (and lattice configurations $\{x(1), x(2), \ldots, x(N)\}$ repeat) in periodic sequences of higher order. For instance, for $0.75 < x_* < 0.905$, we get cycles of order 2, for $0.905 < x_* < 0.925$ we get order 4, for $x_* \sim 0.93$ we get order 6, and so forth. Thus from this *single* spatially distributed chaotic system, we can deterministically [2] extract with a *single* parameter (in this case the threshold value) an infinite variety of dynamical behavior and periods.

The properties that underscore the significance of having chaotic elements in the lattice are: (a) If the same threshold dynamics was imposed on a random lattice, we would not recover any of the above periodicities. For these periodicities to occur we require deterministic dynamics; (b) The *ergodic* properties of chaotic systems guarantee that the system always falls into the desired periodicities and it will not get trapped in any restricted corner of phase space; and (c) chaotic elements will yield an infinite number of periodicities under variation of the threshold.

Construction of gates.—First, we will demonstrate that our dynamical system can emulate a NOR logic gate. One can interpret the state of an element as follows: $x_i = x_*$ is state 1 and $x_i < x_*$ is state 0. The response of the lattice used to characterize the output is the excess transported out of the edge of the lattice as a result of the relaxation. Thus one can imagine a readout at the open end of the lattice which registers the excess signal which represents the output (the answer of the logical operation).

To achieve this logic operation, we operate in the x_* regime where the chaotic elements emit excess as a sequence of period 2 when threshold coupled. Here a two-element unit can have two possible states (note that we always consider attractor states, not transients): (1) The coherent state: This occurs in the range $0.750 < x_* < 0.905$, and emits excess from the open edge in the periodic sequence $0 \rightarrow 2\Delta_1 \rightarrow 0 \rightarrow 2\Delta_1 \ldots$, where

$\Delta_1 = f^2(x_*) - x_*$ and (2) the out-of-phase state which occurs in the range $0.835 < x_* < 0.905$ and which emits excess from the open edge in the periodic sequence $0 \to \Delta_2 \to 0 \to \Delta_2 \ldots$ where $\Delta_2 = f(f(x_*) + \Delta_1) - x_c$.

Thus a particular realization of a NOR gate is achieved as follows: We work in a parameter window around $x_* \sim 0.84$ where the coherent and out-of-phase states are coexisting attractor states and $\Delta_2 \ll \Delta_1$ such that the difference between the coherent and out-of-phase states is clearly discernible. Now the inputs of our logic gate are two elements in specified states. Their collective response after a dynamical update is the excess signal ejected from the open edge of the two-element lattice (which here is the second lattice element, i.e., element two). The collective response should emulate the output of a NOR gate. We obtain this input-output association as follows: If the inputs are $I_1 = 0$ and $I_2 = 0$ we select an attractor state consisting of two elements, both with $x < x_*$. This coherent two-element lattice, $(0, 0)$, after dynamical update, emits $2\Delta_1$ from the open edge. If the inputs are $I_1 = 0$ and $I_2 = 1$, we select the out-of-phase lattice state $(0, 1)$, whose response, after a dynamical update, is to eject a total excess of 0 from the open edge. If the inputs are $I_1 = 1$ and $I_2 = 0$ we select the out-of-phase lattice $(1, 0)$ whose response, after a dynamical update, is to eject a total excess of $\Delta_2 \sim 0$ from the open edge. Finally, if the inputs are $I_1 = I$ and $I_2 = I$ we again choose the coherent lattice state $(1, 1)$ whose response after a dynamical update is to eject a total excess of 0 from the open edge.

Now if we define the output from the open edge of the lattice as: 1 if the ejected amount is $\gg 0$ and 0 if the ejected amount is ~ 0, it is clear that the input-to-output association corresponds to that of a NOR gate. Consequently, any Boolean operation or circuit can be constructed by a suitable coupling of this basic two-element lattice NOR gate. Note that one requires sufficiently strong nonlinearities in the local map in order to obtain the emission pattern necessary for the construction of gates. In our system of coupled logistic maps, only values of $a > 3$ can yield the required attractor states. Intriguingly, logic operations in dynamical systems have been seen before, as demonstrated by Toth and Showalter, who were able to demonstrate logic in a spatially extended chemical dynamical system [4].

Arithmetic operations.——When our lattice elements are not being used in computations they have a default threshold of $x_* = 1$ (i.e., they are effectively decoupled). Specification of the input of an arithmetic operation consists of providing threshold parameters $x_* < 1$ for some elements. This induces an avalanche of excess providing communication of information among these elements. The collective excess from a specified open edge yields the answer.

Encoding and addition scheme 1.—In the threshold range $0 < x_* < 0.75$, a chaotic element under adaptive threshold response emits excess after each dynamical update in order to relax back to x_*. The amount of excess emitted per dynamical update is a unimodal nonlinear function of the threshold, over the range $0 < x_* < 0.75$, going from 0 at $x_* = 0$ to a maximum value $E_{\max} = 9/16$ at $x_* = 3/8$ and then back to 0 again at $x_* = 3/4$. In our encoding scheme the amount of excess emission (in specified units) directly gives the value of the integer. We define the unit of excess emission to be $\delta - E_{\max}/N$, where N is the largest integer we wish to encode. Then an integer m is encoded by an excess emission of $m\delta$. In order to encode integers 0 to N, the necessary capacity of resolution of emitted excess must be $E_{\max}/N = \delta$. Clearly, greater precision in measuring the excess and threshold setting allows larger numbers to be encoded.

Since the map is deterministic, one can determine exactly the threshold which yields a given excess (where excess varies from 0 to N units) and this gives a lookup table associating the value of the threshold to the value of an integer (see Fig. 1). Thus the same element can encode an arbitrarily large set of numbers, under varying threshold (with the threshold levels being sent to it as part of the software or programming).

Typically stronger nonlinearities yield a larger range of excess emission. For instance the parabolic form of the logistic map, at $a = 4$, has the highest maxima and thus yields the largest difference between the map and the effective truncated map after adaptive response as shown in the inset of Fig. 1. The range of excess emission $[0, E_{\max}]$ is determined by the above-mentioned difference with $E_{\max} = (a - 1)^2/4a$ for the logistic map. This range decreases with decreasing strength of the nonlinearity parameter a.

To perform addition on m numbers, we set the threshold of m connected elements such that each encodes a term in the sum. The excess emitted from an element drives its neighboring element, with the element encoding the last term of the sum having the open edge where we register

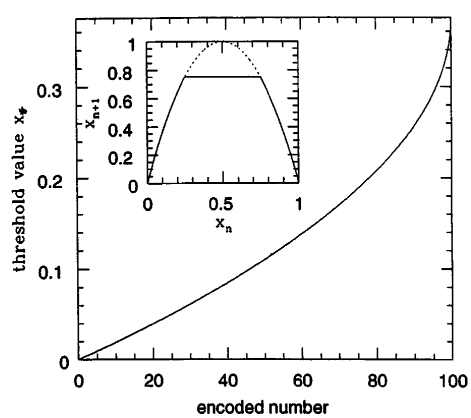

FIG. 1. A lookup graph of encoded number vs threshold value x_*. The encoded number is given by the emitted excess Δ (which is a function of x_c) through the relation: Encoded number $= \Delta/\delta$, where $\Delta = f(x_*) - x_*$ and $\delta = E_{\max}/N$, with $f(x) = 4x(1 - x^2)$ and the largest number encoded ($N = 100$ here). Inset: Return map of a single chaotic element under adaptive threshold response (here $x_* = 3/4$). The difference between the solid and dotted lines is the amount of excess emitted in the dynamical update.

the output. After a dynamical update, an avalanche sweeps across the lattice as shown in Fig. 2. This avalanche gives rise to an excess emission from the open edge which can be directly associated with the result. The addition operation is then achieved simply as follows: *Input the threshold values from the lookup table to encode the numbers to be added and then register the emitted excess from the open edge at the end of one dynamical update.* The dynamics of the lattice is such that this emitted excess is the required answer.

Parallel operations.—Finally, the operation can be done in parallel (synchronously/concurrently) by having a branching topology of the lattice. To add several numbers a branched lattice is employed where each branch is an element encoding a term in the sum. Now the computation time is not proportional to the number of terms in the sum, as in serial addition. Instead, the computation time is independent of the number of terms in the addition, and is always equal to two: In one avalanche step all the branch elements relax and then in a second step the element with the open edge leading to output relaxes. Consequently, this dynamical system, consisting of a highly branching lattice, can serve as a massively parallel computer, with several inputs flowing concurrently into an element, from whose open edge one collects the answer.

Encoding and addition scheme 2.—Now we describe an alternate encoding scheme that exploits the ability of the dynamical system to operate at various periodicites. We denote the threshold yielding excess emission at

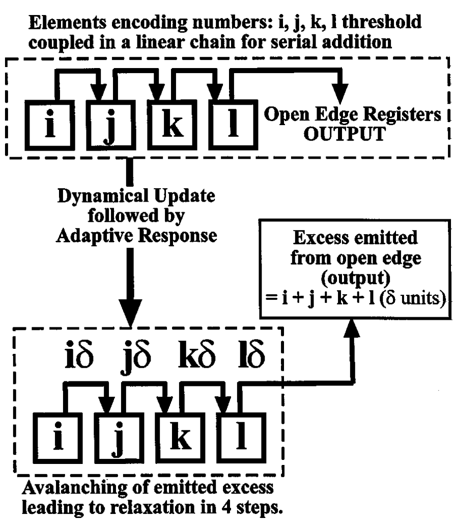

FIG. 2. Threshold coupled chaotic elements emulating an adding machine: Here we are adding four numbers, i, j, k, and l each encoded by an element with threshold set such that it emits i, j, k, l units of excess, respectively (where the unit of excess is δ). These elements are threshold coupled in a chain, with the ejected excess from element $i(= i\delta)$ driving element j, etc., onto element l, from whose open boundary the collective excess is emitted to the output lead. The emitted excess is exactly the sum $i + j + k + l$ in units of δ. The computing time, equal to the duration of the adaptive avalanching process, is equal to the number of terms in the sum (which is four here).

periodicity k as x_*^k. Now in order to encode an N bit binary number whose representation is $a_N a_{N-1} \cdots a_2 a_1$, we use N chaotic elements, each encoding a bit. If the value of the kth bit is 1 (i.e., $a_k = 1$) its threshold is set at $x_*^{2^{N-k}}$, such that it emits excess periodically with period 2^{N-k}. For instance, if the bit farthest from the decimal point $a_N = 1$ it will be encoded by an element whose period is 1, while if $a_1 = 1$ it will be encoded by an element whose periodicity 2^{N-1}. If the value of a bit is 0, then the element representing the bit has its threshold set at 0 resulting in zero emission.

To obtain the value of the number $a_N \cdots a_1$, we have to threshold couple the N elements representing the bits, with a_N having the open edge to the readout. The excess emitted by this N element over one period of the longest period 2^{N-1} (i.e., over 2^{N-1} dynamical updates for a N-bit number) gives the value of the number, namely $\sum_{k=1,N} a_k 2^{k-1}$. This encoding scheme exploits chaos as it employs many different periods and only a chaotic element can yield all of them under varying threshold. The scheme can be easily modified to encode any other base expansions (such as decimals) as well. It should be noted that one has to take care in choosing the same unit of excess emission for all cycles in that the amount an element with threshold x_*^k emits after k steps should be the same for all k. The threshold values for which a requisite set of cycles emits the same excess can be determined exactly. The number of bits that can be encoded is limited by the resolution of excess emission and threshold setting.

For addition, we again threshold couple the lattice of elements representing the terms in the sum. After evolution over 2^{N-1} dynamical updates the coupled elements will eject from the open boundary an amount equal to the result of the addition. This operation commutes and any number of terms can be threshold coupled together (i.e., added) in series (linear lattice configuration) or parallel (branching lattice configuration). The relaxation time (which determines the computing speed) for serial addition of m N-bit numbers is $\leq m + N - 1$, while for parallel addition it is $\leq N + 1$. A specific example of the parallel addition operation is demonstrated in Fig. 3.

Multiplication.—Multiplication can be performed (as an extension of addition) by invoking the same parallel computational approach through branching. For instance, to multiply m by n, we have a lattice with n branches, each branch being a copy of the lattice element encoding m (via encoding scheme 1 or 2). The total ejected excess will be the answer: $m \times n$. Alternately, we can take the lattice element encoding m and collect the emitted excess over n dynamical updates. The collected excess, equal to $m \times n$, yields the answer. This has the advantage of requiring only one element of the lattice in order to calculate the product (in contrast to the former method which requires n elements) while costing n times more than the former method (n dynamical updates are required instead of 1). Depending on the resources available, one could either operate with many elements or with fewer elements with

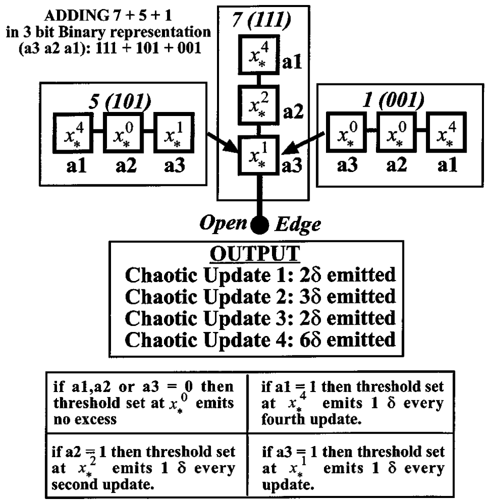

FIG. 3. Parallelized addition operation of three integers: 7, 5, and 1, where the terms of the addition are encoded by a chain of three elements each. These are threshold coupled in a branching configuration. Now after four dynamical updates (since the longest period $= 2^{3-1} = 4$) the entire branching lattice emits a total excess of 13 units, which is the result of the operation: $7 + 5 + 1 = 13$.

quicker dynamics, in order to achieve the operation in the same amount of time.

Least common multiple.—We can also devise dynamical algorithms, which exploit the dynamics to perform other, more specialized operations. For instance, we have realized a dynamical algorithm for finding a least common multiple (LCM) of a sequence of integers. To find the LCM of a sequence of n integers: $k_1, k_2, \ldots, k_n$ we use n chaotic elements as the input. These n input elements have their threshold fixed such that they emit excess cyclically with periods equal to the values of the integers they represent, namely $k_1, k_2, \ldots, k_n$. The periodicity of the excess emitted thus represents the value of the terms of the LCM.

The deterministic dynamics of the local elements allows one to obtain exact generating equations for windows of threshold values supporting a certain periodicity [2]. Thus one can obtain a lookup table relating periodicity of excess emission to threshold in order to represent any positive integer. Again, only local chaos can provide all the periodicities necessary to represent the whole range of inputs via the same single element.

Now these input elements are coupled in parallel to one master element whose threshold is fixed at $x_* < 0.75$ and which has the open edge whose excess provides the final answer. The excess ejected from the input elements synchronously stimulates the master element, which in turn emits excess from its open edge with periodicity equal to the LCM of all the input stimuli periods. Thus one obtains the LCM of the terms by simply measuring the period of the master element's response. Note that one can handle many terms in parallel by stimulating the master element synchronously with different periodic impulses. This simple but intriguing example demonstrates that dynamic algorithms hold the potential for computing a range of specialized mathematical operations. Thus we begin to see the first glimpse that dynamics can perform computation not just by emulating logic gates or simple arithmetic operations, but by performing more sophisticated operation through self-organization rather than composites of simpler operations.

Obviously it remains to be seen how to match dynamical systems with specific computational problems. A key feature of dynamical computing is its ability to handle general computational tasks. This appears to be in contrast with DNA [5,6] and quantum computers [7] which seem to be geared to handle *specific* problems suited specially to their physical properties. Additionally, we can demonstrate [8] that the methods for dynamical computation outlined in this paper work (even in the presence of noise) with continuous nonlinear differential equations, such as those which model coherently pumped far-infrared NH_3 lasers [9]. Thus, applications of this technique might prove useful for high speed optical computing. Further, while it is known that coupled chaotic maps [10] can, *in principle*, be viewed as universal computers, we have shown *in practice* that the generic chaotic properties of nonlinear dynamical systems can perform a variety of computations. Thus the potential of dynamical computation lies in the possibility of designing a *single* spatially extended dynamical system to perform a *variety* of computational tasks by exploiting the rich and complex dynamics and pattern formation of spatially extended nonlinear systems.

We would like to thank Kurt Wiesenfeld for many helpful discussions. We acknowledge support from ONR and Control Dynamics, Inc.

*Electronic address: sudeshna@imsc.ernet.in
†Electronic address: wditto@acl.gatech.edu

[1] W. L. Ditto and L. M. Pecora, Sci. Am. **269**, No. 2, 62 (1993); S. Hayes, C. Grebogi, E. Ott, and A. Mark, Phys. Rev. Lett. **73**, 1781 (1994); G. D. VanWiggeren, and R. Roy, Science **279**, 1198 (1997).
[2] S. Sinha and D. Biswas, Phys. Rev. Lett. **71**, 2010 (1993); S. Sinha, Phys. Rev. E **49**, 4832 (1994); S. Sinha, Phys. Lett. A **199**, 365 (1995).
[3] P. Bak, C. Tang, and K. Wiesenfeld, Phys. Rev. A **38**, 364 (1988).
[4] A. Toth and K. Showalter, J. Chem. Phys. **103**, 2058 (1995).
[5] L. M. Alderman, Science **266**, 1021 (1994).
[6] R. Pool, Science **268**, 498 (1995).
[7] Andrew Steane, Rep. Prog. Phys. **61**, 117–173 (1998).
[8] S. Sinha and W. L. Ditto (to be published).
[9] C. O. Weiss *et al.*, Appl. Phys. B **61**, 223–242 (1995).
[10] A. V. Holden *et al.*, Chaos **2**, 367 (1992).