Continuous-Time Symmetric Hopfield Nets Are Computationally Universal

Jiří Šíma*
Institute of Computer Science,
Academy of Sciences of the Czech Republic,
P. O. Box 5, 182 07 Prague 8, Czech Republic, *sima@cs.cas.cz*

Pekka Orponen†
Laboratory for Theoretical Computer Science,
Helsinki University of Technology,
P. O. Box 5400, FIN-02015 HUT, Finland, *orponen@tcs.hut.fi*

Abstract

We establish a fundamental result in the theory of computation by continuous-time dynamical systems, by showing that systems corresponding to so called continuous-time symmetric Hopfield nets are capable of general computation. As is well known, such networks have very constrained, Liapunov-function controlled dynamics. Nevertheless, we show that they are universal and efficient computational devices, in the sense that any convergent synchronous fully parallel computation by a recurrent network of n discrete-time binary neurons, with in general asymmetric coupling weights, can be simulated by a symmetric continuous-time Hopfield net containing only $18n+7$ units employing the saturated-linear activation function. Moreover, if the asymmetric network has maximum integer weight size $w_{\max}$ and converges in discrete time t^*, then the corresponding Hopfield net can be designed to operate in continuous time $\Theta(t^*/\varepsilon)$, for any $\varepsilon > 0$ such that $w_{\max}2^{12n} \leq \varepsilon 2^{1/\varepsilon}$. In terms of standard discrete computation models, our result implies that any polynomially space-bounded Turing machine can be simulated by a family of polynomial-size continuous-time symmetric Hopfield nets.

1 Introduction

In recent years, a number of studies have sought to understand the computational characteristics of "natural" dynamical systems. The results achieved include, e.g. universal computation results for several types of ODE's (Asarin & Maler, 1994; Branicky, 1994), PDE's (Omohundro, 1984), and discrete iterations (Koiran, Cosnard, & Garzon, 1994; Moore, 1990; Siegelmann & Sontag, 1995). One much studied class of systems are those that are defined by various *neural network* models (Golden, 1996; Haykin, 1999; Siegelmann, 1999). This interest is motivated partly by the quest to understand the fundamental limits and possibilities of practical neurocomputing, and partly by the realization that despite their formal simplicity,

*Research partially supported by grants GA AS CR B2030007, GA CR No. 201/02/1456.
†Research partially supported by grant 81120 from the Academy of Finland.

neural networks are computationally quite powerful, and thus may serve as a useful reference model for investigating more complicated systems. In general, the computational capabilities of discrete-time systems are by now fairly well understood, but in the area of continuous-time systems much work remains to be done. To get an overview of the issues, see the survey papers by Moore (1998) and Orponen (1997a).

Continuous-time recurrent neural networks are an attractive class of computational models with applications in, e.g., control, optimization, and signal processing (Cichocki & Unbehauen, 1993; Medsker & Jain, 2000). Probably the best-known, and most widely-used continuous-time recurrent network model is that popularized by John Hopfield (1984), and often called the "continuous-time Hopfield model". The dynamics of this model were actually already analyzed earlier by Cohen and Grossberg (1983) in a more general setting, but because of the affinity to the very influential discrete-time binary-state version of the model (Hopfield, 1982), this additive special case of the Cohen-Grossberg equations has become associated to Hopfield's name. As practical neural networks, proposed uses of Hopfield nets include associative memory (Hopfield, 1984) and fast approximate solution of combinatorial optimization problems (Hopfield & Tank, 1985), and designs exist for implementing them in analog electrical (Hopfield, 1984) and optical (Stoll & Lee, 1988) hardware.

In this paper, we prove a fundamental result concerning the computational power of continuous-time symmetric Hopfield nets. It is well known (Cohen & Grossberg, 1983; Hopfield, 1984) that the dynamics of any Hopfield net with a *symmetric* coupling weight matrix is governed by a *Liapunov*, or *energy function*. This is a bounded function defined on the state space of a network, whose values are properly decreasing along any nonconstant trajectory of the network's dynamics. In particular, such a symmetric network always converges from any initial state towards some stable equilibrium state. This is a very useful property for obtaining guaranteed behavior in practical applications, but would at first sight seem to severely limit the networks' general dynamical capabilities. For instance, nondamping oscillations of the network state, which seem to be an essential prerequisite of general computation, obviously cannot be created under this constraint (i.e. strictly speaking continuous-time Hopfield nets cannot simulate even a single alternating bit), whereas such oscillations are easily obtained in networks with *asymmetric* coupling weights.

Nevertheless, we shall show that infinite oscillations are the *only* feature of general-purpose digital computation that cannot be reproduced in continuous-time symmetric Hopfield nets. More precisely, we prove that any *converging* discrete-time computation described by a recurrent network of $n \geq 1$ binary neurons working under synchronous fully parallel mode and with in general asymmetric coupling weights, can be embedded in a continuous-time Hopfield net of $18n + 7$ units with a symmetric coupling weight matrix, and using the saturated-linear activation function. In particular, the simulation of t^* discrete update steps can be achieved in continuous time $\Theta(t^*/\varepsilon)$, for any $\varepsilon > 0$ such that $w_{\max} 2^{12n} \leq \varepsilon 2^{1/\varepsilon}$ where $w_{\max} \geq 1$ is the maximum integer weight size of the discrete network. An experimental validation of this result appeared previously in the extended abstract (Šíma & Orponen, 2000).

A key observation is that any terminating computation by a discrete-time deterministic network with state space $\{0,1\}^n$ must converge within 2^n steps. A basic technique used in our proof is then the construction of a symmetric continuous-time *clock* network—a simulated $(n+2)$-bit binary counter—that, using $8n+7$ units, produces a sequence of 2^n well-controlled oscillations (generated by the simulated second least significant counter bit) before it converges. This sequence of clock pulses is used to drive the rest of the network where each discrete neuron is simulated by a symmetrically coupled subnetwork of 10 continuous-time units. The continuous-time clock is already by itself of some interest from a dynamical systems perspective, because it provides to our knowledge the first known example of a continuous-time Liapunov system whose convergence time grows exponentially in the system dimension (Šíma & Orponen, 2001a).

A similar, but considerably simpler, construction was used in the *discrete-time* setting (Orponen, 1996) to prove the computational equivalence of symmetric and convergent asymmetric binary neural networks. Our present construction also provides the technique for improving this discrete-time simulation by reducing its overhead from a quadratic number of units to the asymptotically optimal linear size (Šíma, Orponen, & Antti-Poika, 2000). The original idea for the discrete-time clock network used by Orponen (1996), and on which our current construction is based, stems from the result by Goles and Martínez (1989). Another related work (Orponen, 1997b) concerns the simulation of discrete-time binary networks by continuous-time *asymmetric* networks which, however, do not adhere to the Liapunov property.

It is quite easy to see (Lepley & Miller, 1983; Orponen, 1996) that families of polynomial-size discrete neural networks are computationally equivalent to (nonuniform) polynomially space-bounded Turing machines; more precisely, they compute the complexity class PSPACE/poly (Balcázar, Díaz, & Gabarró, 1995, p. 105). By the result in the present paper, we now know that continuous-time symmetric Hopfield nets are also at least as powerful, i.e. given any polynomially space-bounded Turing machine, we can construct a family of polynomial-size continuous-time symmetric Hopfield nets (one network for each input length) for simulating it. This is to our knowledge the first result concerning the computational power of continuous-time symmetric networks, and it is somewhat surprising that they turn out to be computationally universal in this complexity-limited sense. It remains an open question whether this is also an upper bound on the power of such networks.

A related line of study concerns the computational capabilities of finite *discrete-time analog-state* neural networks (Siegelmann, 1999). Here it is known that the computational power of *asymmetric* networks using the saturated-linear activation function increases with the Kolmogorov complexity of the real weight parameters (Balcázar, Gavaldà, & Siegelmann, 1997). With integer weights such networks coincide with binary networks, and so are equivalent to finite automata (Horne & Hush, 1996; Indyk, 1995; Šíma & Wiedermann, 1998), while with rational weights arbitrary Turing machines can be simulated (Indyk, 1995; Siegelmann & Sontag, 1995). With arbitrary real weights the networks can even have "super-Turing" computational capabilities (Siegelmann & Sontag, 1994). To some extent similar results also apply to discrete-time *symmetric* analog networks that are computationally equivalent to their asymmetric counterparts when an external clock pulse sequence is provided (Šíma, Orponen, & Antti-Poika, 2000). On the other hand, it is known that any amount of analog noise reduces the computational power of discrete-time analog recurrent networks to that of finite automata (Casey, 1996; Maass & Orponen, 1998) or even less (Maass & Sontag, 1999; Siegelmann, Roitershtein, & Ben-Hur, 2000).

A preliminary version of this paper appeared as an extended abstract (Šíma & Orponen, 2001b). The present article is organized as follows. After a brief review of the basic definitions in section 2, our main construction of the symmetric continuous-time Hopfield net simulating a given discrete-time binary neural network is outlined in section 3 where its dynamics is also informally explained. The formal verification of this construction, which has the form of a rather tedious case analysis, is given in section 4. In section 5 a numerical simulation example witnessing the validity of the construction is presented. Section 6 concludes with some open problems.

2 Preliminaries

We will first specify the model of a finite *discrete-time binary-state recurrent neural network (DRNN)*. Such a network consists of n simple computational binary *units* or *neurons*, indexed as $1,\ldots,n$, that are connected into a generally cyclic directed graph or *architecture*. Each edge (i,j) in this graph, leading from neuron i to j, is labeled with an integer coupling *weight* $w(i,j) = w_{ji} \in \mathbf{Z}$. Note that for binary neurons *integer* weights can be assumed without loss of generality (see e.g. Parberry, 1994). The absence of an edge in the architecture indicates a zero weight between the respective neurons, and vice versa.

An instantaneous *state* of a DRNN at time $t \geq 0$ is described by a vector $\mathbf{y}^{(t)} = (y_1^{(t)},\ldots,y_n^{(t)}) \in \{0,1\}^n$ composed of current binary *outputs* (*states*) $y_j^{(t)}$ from particular neurons $j = 1,\ldots,n$. We consider here only the synchronous *fully parallel* dynamics in which the evolution of the network state $\mathbf{y}^{(t)}$ is determined for discrete time instants $t = 0,1,\ldots$, as follows. At the beginning of a computation the network is placed in some *initial state* $\mathbf{y}^{(0)}$, which may include an external input. At discrete time $t \geq 0$, each neuron $j = 1,\ldots,n$ collects its local binary *inputs* from the *states* (*outputs*) $y_i^{(t)} \in \{0,1\}$ of incident neurons i and determines its integer *excitation*

$$\xi_j^{(t)} = \sum_{i=0}^{n} w_{ji} y_i^{(t)} \,, \quad j = 1,\ldots,n \tag{2.1}$$

as the respective weighted sum of these inputs. Moreover, sum 2.1 includes an integer *bias* $w_{j0} \in \mathbf{Z}$ local to each neuron j, which is formally modeled as a coupling weight from an additional neuron with index 0 and constant unit output $y_0^{(t)} \equiv 1$. At the next instant $t+1$, the new network state $\mathbf{y}^{(t+1)}$ is computed by applying an *activation function* to all excitations componentwise:

$$y_j^{(t+1)} = H\left(\xi_j^{(t)}\right) \,, \quad j = 1,\ldots,n \tag{2.2}$$

where the *Heaviside* or *threshold* function

$$H(\xi) = \begin{cases} 1 & \text{for } \xi \geq 0 \\ 0 & \text{for } \xi < 0 \end{cases} \tag{2.3}$$

is employed here. We also denote by

$$w_{\max} = \max_{j=1,\ldots,n;\, i=0,\ldots,n} |w_{ji}| \tag{2.4}$$

the *maximum weight* size in the network.

Similarly, a finite *continuous-time analog neural network* is composed of m analog units which operate (in our case) with the *saturated-linear* activation function

$$\sigma(\xi) = \begin{cases} 1 & \text{for } \xi \geq 1 \\ \xi & \text{for } 0 < \xi < 1 \\ 0 & \text{for } \xi \leq 0\,. \end{cases} \tag{2.5}$$

Hence, the states $y_p(t)$ of analog units $p = 1,\ldots,m$ at time $t \geq 0$ are real numbers within the interval $[0,1]$, and the coupling weights (including biases), denoted by $v(p,q) \in \mathbf{R}$ for edges from p to q, are reals as well. The computational dynamics of a continuous-time network is defined for every real $t > 0$, e.g. by the following system of ordinary differential equations in real variables $y_1,\ldots,y_m \in [0,1]$:

$$\frac{dy_p}{dt}(t) = -y_p(t) + \sigma(\xi_p(t)) \,, \quad p = 1,\ldots,m \tag{2.6}$$

where

$$\xi_p(t) = \sum_{q=0}^{m} v(q,p) y_q(t) \tag{2.7}$$

is the real-valued excitation for unit p at time $t \geq 0$ (cf. equation 2.1) including bias $v(0,p) \in \mathbf{R}$ which is associated with an additional formal variable y_0 whose value is constant $y_0(t) \equiv 1$ in time. The initial network state $\mathbf{y}(0) \in [0,1]^m$ then determines the boundary conditions for system 2.6.

In particular, we shall consider the continuous-time *(symmetric) Hopfield networks*, whose architecture is an undirected graph with symmetric weights

$$v(p,q) = v(q,p) \quad \text{for every } \; 1 \leq p,q \leq m\,. \tag{2.8}$$

The dynamics of a continuous-time *symmetric* Hopfield net described by system 2.6 is controlled by the following *Liapunov* or *energy* function, introduced by Cohen and Grossberg (1983):

$$E(\mathbf{y}) = -\frac{1}{2} \sum_{p=1}^{m} \sum_{q=1}^{m} v(q,p) y_q y_p - \sum_{p=1}^{m} v(0,p) y_p + \sum_{p=1}^{m} \int_0^{y_p} \sigma^{-1}(y) dy\,. \tag{2.9}$$

The characteristic properties of function $E(\mathbf{y})$ are that it is bounded on the system's state space $[0,1]^m$, and that it is properly decreasing (i.e. $dE/dt < 0$) along any nonconstant trajectory of the network's dynamics. It then follows that the continuous-time Hopfield net always converges, from any initial state, towards some stable equilibrium state with $dy_p/dt = 0$ for all $p = 1,\ldots,m$.

3 Constructing the Continuous-Time Hopfield Net

We shall now show how to simulate the convergent computations of a given DRNN $\mathcal{N}$ on a somewhat bigger continuous-time Hopfield network $\mathcal{H}$. In our simulation, the binary output values 0 and 1 of discrete neurons in $\mathcal{N}$ will be represented by excitations 2.7 of corresponding analog units in $\mathcal{H}$ that are below the lower saturation threshold of 0 or above the upper saturation threshold of 1, respectively, for activation function 2.5. For brevity, we shall simply say that a unit p is *saturated* at 0 or 1 at time t if its excitation satisfies $\xi_p(t) \leq 0$ or $\xi_p(t) \geq 1$, respectively. We also say that p is *unsaturated* when $0 < \xi_p(t) < 1$.

Note that we use the *excitations* $\xi_p(t)$ of continuous-time units $p \in \mathcal{H}$ rather than their actual *states* $y_p(t)$ to represent the binary values since starting at any point within the open interval $(0,1)$, the outputs $y_p(t)$ of the units saturated at 0 or 1 only converge to limit values 0 or 1, respectively, and never reach these boundaries (see lemma 2.1 below). In fact, this is the main difference between the discrete dynamics 2.2 and its simulation within the continuous dynamics 2.6. The following theorem summarizes the result:

Theorem 1 *Any fully parallel computation by a recurrent neural network $\mathcal{N}$ of $n \geq 1$ binary neurons with asymmetric integer weights of maximum size $w_{\max} \geq 1$, converging within t^* discrete update steps, can be simulated by a continuous-time symmetric Hopfield network $\mathcal{H}$ with $m = 18n + 7$ analog units employing the saturated-linear activation function, within continuous time $\Theta(t^*/\varepsilon)$ for any real $\varepsilon > 0$ such that*

$$w_{\max} 2^{12n} \leq \varepsilon 2^{1/\varepsilon}\,. \tag{3.1}$$

Proof: Since the DRNN $\mathcal{N}$ defines a deterministic dynamical system over the state space $\{0,1\}^n$, any converging computation by $\mathcal{N}$ must terminate within $t^* \leq 2^n$

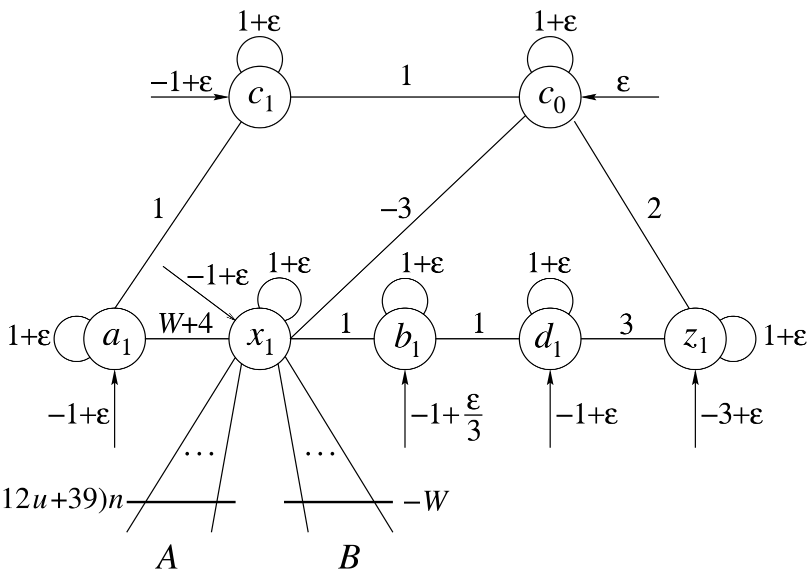


Figure 1: A 2-bit continuous-time counter $\mathcal{C}_1$.

discrete steps. To achieve the simulation of $\mathcal{N}$ by Hopfield net $\mathcal{H}$ we first construct in paragraph 3.1 a continuous-time symmetric *clock subnetwork* $\mathcal{C} = \mathcal{C}_{n+1}$ with $8n+7$ units that simulates a discrete $(n+2)$-bit binary counter. When the network $\mathcal{C}$ is initialized in the zero state, its *interface unit* x_1 will exhibit a sequence of 2^n well-controlled oscillations before $\mathcal{C}$ converges. Each "clock pulse" from $x_1 \in \mathcal{C}$ is then exploited to "drive" a simulation of one parallel discrete update iteration by a *simulating subnetwork* $\mathcal{S}$ of size $10n$ units that is constructed in paragraph 3.2. In particular, $\mathcal{S} = \cup_{j=1}^{n} \mathcal{G}_j$ consists of n continuous-time symmetric *gate subnetworks* $\mathcal{G}_j$ $(j = 1, \ldots, n)$ each having 10 units for simulating one discrete neuron $j \in \mathcal{N}$. We shall now first discuss the operation of the continuous-time Hopfield net $\mathcal{H} = \mathcal{C} \cup \mathcal{S}$ intuitively while its correctness will formally be verified later in section 4.

3.1 The Clock Network

The construction of the continuous-time symmetric clock network $\mathcal{C} = \mathcal{C}_{n+1}$ simulating an $(n+2)$-bit binary counter of "order (n+1)" will be described by induction on n. The induction starts with the 2-bit counter network $\mathcal{C}_1$ presented in Figure 1. The undirected edges connecting units in this graph are labeled with the corresponding symmetric weights whereas the oriented edges drawn without an originating unit correspond to the biases. Assume that initially all the states of the units indicated in this network are zero. The least significant counter unit c_0 of "order 0" has positive bias $v(0, c_0) = \varepsilon > 0$ corresponding to its initial positive excitation. Because of its feedback coupling $v(c_0, c_0) = 1 + \varepsilon > 1$ the state of c_0 gradually grows from initial 0 towards 1. Eventually c_0 saturates at 1, at which point we say that the unit c_0 becomes *active* or *fires*. Recall that we associate the simulated discrete counter behavior to the excitations of the units rather than their outputs. The external state of c_0 of course evolves smoothly converging to 1, and exhibits no abrupt "firing" transitions. Thus, c_0 simply implements counting from 0 to 1 as required. This trick of gradual transition from 0 to 1, formally described in lemma 3 below, is used repeatedly throughout our construction of $\mathcal{C}$.

The remaining six units $c_1, a_1, x_1, b_1, d_1, z_1$ in the 2-bit counter network $\mathcal{C}_1$ (see Figure 1) are of "order 1". They function similarly to the corresponding units of

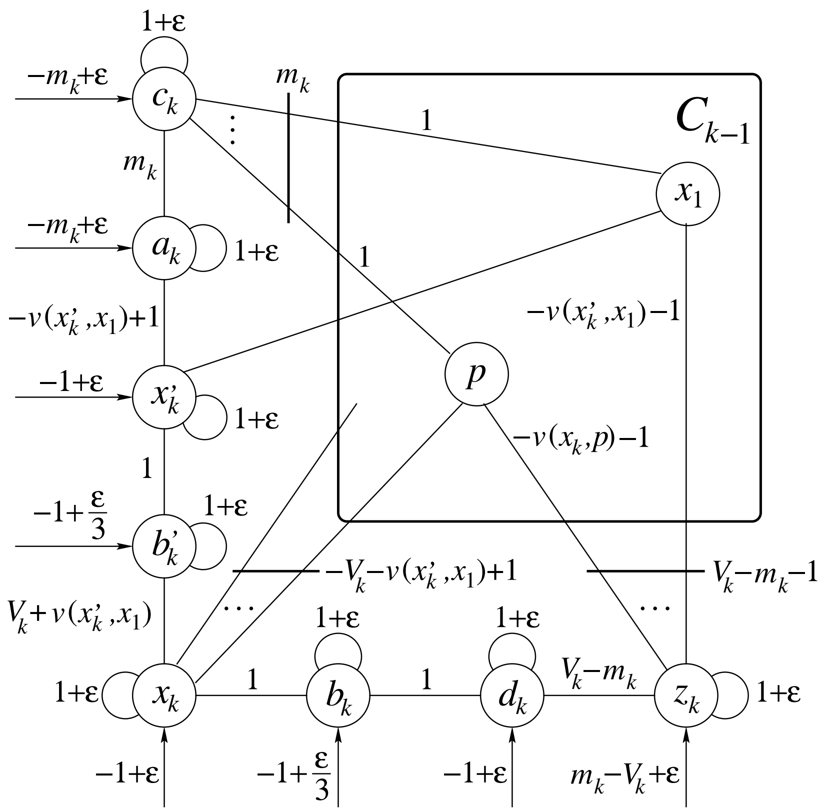


Figure 2: Inductive construction of $\mathcal{C}_k$.

higher orders, so we will describe below only the inductive construction for general order $k > 1$. The eventual interconnection of the clock network $\mathcal{C}$ to the simulating subnetwork $\mathcal{S}$, i.e. to the gate subnetworks $\mathcal{G}_j$ ($j = 1, \ldots, n$) is taken into account in the weight $v(a_1, x_1) = W + 4$, which includes a large positive integer parameter

$$W = \frac{3}{2}\left(nu + 28\left(n + \sum_{j=1}^{n} |w_{j0}|\right)\right) > 0\,, \tag{3.2}$$

where

$$u = 28 \max_{j=1,\ldots,n}\left(-\sum_{1\le i\le n;\, w_{ji}\le 0} w_{ji}\,,\ \sum_{1\le i\le n;\, w_{ji}\ge 0} w_{ji}\right) \ge 0\,, \tag{3.3}$$

and w_{ji} are the weights of the original DRNN $\mathcal{N}$ to be simulated. This large weight is needed because the interface unit x_1 transfers the pulses generated by the clock $\mathcal{C}$ to all of the gate subnetworks $\mathcal{G}_j$, and its operation within $\mathcal{C}$ must not be affected by feedback effects from $\mathcal{S}$ (see section 3.2).

For the induction step depicted in Figure 2, assume that an "order $(k-1)$" counter network $\mathcal{C}_{k-1}$ ($1 < k \le n+1$) has been constructed, containing the first k counter units $c_0, \ldots, c_{k-1} \in \mathcal{C}_{k-1}$, together with auxiliary units $a_\ell, x_\ell, b_\ell, d_\ell, z_\ell \in \mathcal{C}_{k-1}$ ($\ell = 1, \ldots, k-1$), and for $k > 2$ also x'_ℓ, b'_ℓ ($\ell = 2, \ldots, k-1$), for a total of

$$m_k = |\mathcal{C}_{k-1}| = k + 5(k-1) + 2(k-2) = 8k - 9 \tag{3.4}$$

units. Then the next counter unit c_k is connected to all the m_k units $p \in \mathcal{C}_{k-1}$ via unit weights $v(p, c_k) = 1$, which, together with its bias $v(0, c_k) = -m_k + \varepsilon$

and feedback weight $v(c_k, c_k) = 1 + \varepsilon$, cause c_k to fire shortly after all these units are active (lemma 3). This includes the first k active counter bits $c_0, \ldots, c_{k-1}$, which means that the simulated counting from 0 to $2^k - 1$ has been accomplished, and hence, the next counter bit c_k must fire. In addition, unit c_k is connected to a sequence of seven auxiliary units $a_k, x'_k, b'_k, x_k, b_k, d_k, z_k$ with feedback weights $v(a_k, a_k) = v(x'_k, x'_k) = v(b'_k, b'_k) = v(x_k, x_k) = v(b_k, b_k) = v(d_k, d_k) = v(z_k, z_k) = 1 + \varepsilon$, which are being, one by one, activated after c_k fires (lemma 3). This is implemented by the weights $v(c_k, a_k) = m_k$, $v(a_k, x'_k) = -v(x'_k, x_1) + 1$, $v(x'_k, b'_k) = v(x_k, b_k) = v(b_k, d_k) = 1$, $v(b'_k, x_k) = V_k + v(x'_k, x_1)$, $v(d_k, z_k) = V_k - m_k$, and biases $v(0, a_k) = -m_k + \varepsilon$, $v(0, x'_k) = v(0, x_k) = v(0, d_k) = -1 + \varepsilon$, $v(0, b'_k) = v(0, b_k) = -1 + \varepsilon/3$, $v(0, z_k) = m_k - V_k + \varepsilon$, where $v(x'_k, x_1) < 0$ is in its absolute value a sufficiently large, negative weight and $V_k > 0$ is a sufficiently large, positive parameter, whose values will be determined by formulas 3.5 and 3.8, respectively, so that units x'_k, x_k, z_k are not directly influenced by the computations occurring in units from $\mathcal{C}_{k-1}$ except via c_k.

The purpose of the auxiliary units a_k, b'_k, b_k, d_k is only to slow down the continuous-time state flow in order to synchronize the computation.

The units x'_k, x_k are used to reset all the lower-order units in $\mathcal{C}_{k-1}$ back to values near 0 by saturating them at 0 (lemma 2.2b) after c_k fires, which is consistent with the correct counter computation. The units that saturate at 0 are then called *passive*. To achieve this effect, x'_k is connected to x_1 via a large negative weight

$$v(x'_k, x_1) = -S_{kx_1} - (12u + 39)n \, , \tag{3.5}$$

and similarly, x_k is linked with each $p \in \mathcal{C}_{k-1} \setminus \{x_1\}$ via negative weight

$$v(x_k, p) = -S_{kp} \, , \tag{3.6}$$

where parameter

$$S_{kp} = \left\lceil v(c_k, p) + \sum_{q \in \mathcal{C}_{k-1};\, v(q,p) > 0} v(q, p) \right\rceil \, , \quad p \in \mathcal{C}_{k-1} \tag{3.7}$$

exceeds the positive influence of units in $\mathcal{C}_{k-1} \cup \{c_k\}$ on p, and term $-(12u+39)n$ in equation 3.5 balances the total positive influence on the interface unit x_1 originating from units A in the gate subnetworks $\mathcal{G}_j$ (see section 3.2). Now, the value of parameter

$$V_k = 1 - v(x'_k, x_1) - \sum_{p \in \mathcal{C}_{k-1} \setminus \{x_1\}} v(x_k, p) \tag{3.8}$$

is determined so that unit x_k fires after b'_k is activated in spite of the negative contributions through weights 3.6 from $p \in \mathcal{C}_{k-1} \setminus \{x_1\}$ to the excitation of x_k.

Note that the interface unit $x_1 \in \mathcal{C}_{k-1}$ is first suppressed separately by x'_k (lemma 4) while the remaining units in $\mathcal{C}_{k-1} \setminus \{x_1\}$ are active. In particular, units a_1 and b_1 incident on x_1 that are of the same order 1 remain activated after x_1 becomes passive because of active c_1 and d_1, respectively, and unit c_0 is connected with x_1 via a negative weight (see Figure 1). Furthermore, units in $\mathcal{C}_{k-1}$ of order greater than 1 are not directly influenced by passive x_1. Thus, only after x_1 becomes passive and b'_k, x_k are activated, the other units from $\mathcal{C}_{k-1} \setminus \{x_1\}$ are reset by x_k.

Finally, unit z_k balances the negative influence of x'_k, x_k on $\mathcal{C}_{k-1}$ so that the first k counter bits can again count from 0 to $2^k - 1$ but now with c_k being active. This is achieved by exact positive weights

$$v(z_k, p) = \begin{cases} -v(x'_k, x_1) - 1 & \text{for } p = x_1 \\ -v(x_k, p) - 1 & \text{for } p \in \mathcal{C}_k \setminus \{x_1\} \end{cases} \tag{3.9}$$

in which $-v(x'_k, x_1)$ and $-v(x_k, p)$ eliminate the influence of x'_k and x_k, respectively, on $p \in \mathcal{C}_{k-1}$, whereas -1 compensates for $v(c_k, p) = 1$ for each $p \in \mathcal{C}_{k-1}$. Clearly, units $p \in \mathcal{C}_{k-1}$ cannot reversely activate z_k since their maximal contribution to the excitation of z_k,

$$\sum_{p \in \mathcal{C}_{k-1}} v(p, z_k) = -m_k - v(x'_k, x_1) - \sum_{p \in \mathcal{C}_{k-1} \setminus \{x_1\}} v(x_k, p) = V_k - m_k - 1 \qquad (3.10)$$

according to equations 3.9, 3.8, cannot overcome its bias $v(0, z_k) = m_k - V_k + \varepsilon$. This completes the induction step of the counter network construction.

3.2 The Simulating Network

The high-level scheme of the simulating subnetwork $\mathcal{S}$ is depicted in Figure 3. Network $\mathcal{S}$ consists of a sequence of five layers, each layer being linked to the subsequent one. A signal is propagated through these layers (lemma 2.2b) only in one direction, i.e. from top to bottom in Figure 3, even though the respective connections are symmetric. For this purpose, a device used previously for symmetric implementation of feedforward networks (Parberry, 1994) is applied here. In particular, the absolute values of weights and biases in the symmetric network are arranged in a decreasing sequence, which prevents the signal from back-propagating. However, to make recurrent computation possible, the "bottom" layer is further connected to the "top" layer—but only via weights of small absolute values that need additional strong support from the clock $\mathcal{C}$ in order to transmit a signal.

Therefore, Figure 3 also shows the clock subnetwork $\mathcal{C}$ that has been constructed in paragraph 3.1. Particularly, interface unit x_1 of $\mathcal{C}$ is connected to the first layer denoted by $A = \{\alpha_j, \beta_j;\, j = 1, \ldots, n\}$ via large positive weights and at the same time x_1 is linked to the fourth layer $B = \{\phi_j, \chi_j;\, j = 1, \ldots, n\}$ via negative weights of large absolute values (cf. Figure 1). Within the simulated $(n+2)$-bit binary counter $\mathcal{C}_{n+1}$, unit x_1 is of the second least significant order 1 and thus fires 2^n times. To each discrete computational step of network $\mathcal{N}$ that converges after $t^* \leq 2^n$ updates then corresponds a two-phase continuous-time state dynamics of $\mathcal{S}$, controlled by one oscillation of $x_1 \in \mathcal{C}$.

In particular, the next simulated state of $\mathcal{N}$ is computed in the first phase when unit x_1 is passive. In this phase, the first layer A is *locked*, i.e. no signal can pass through A, since there is no support from $\mathcal{C}$. At the same time, the negative influence of x_1 on the fourth layer B is suppressed. The third layer of $\mathcal{S}$ serves as a memory that encodes the current binary state $\mathbf{y}^{(t)}$ of the simulated network $\mathcal{N}$ at discrete time instant t. This memory is properly initialized by the initial state $\mathbf{y}^{(0)}$ of $\mathcal{N}$. The new state $\mathbf{y}^{(t+1)}$ of $\mathcal{N}$ at each time instant $t+1$ is then computed from $\mathbf{y}^{(t)}$ in the subsequent fourth layer by using the original weights of $\mathcal{N}$ and it is further stored in the following fifth memory layer. This completes the first phase and $\mathcal{S}$ is stabilized until x_1 fires.

In the second phase, when unit x_1 is active, the new simulated state of $\mathcal{N}$ replaces the old one. The fourth layer B is now locked due to the large negative weights from x_1, while the first layer A is unlocked by the positive support from $\mathcal{C}$. The new state $\mathbf{y}^{(t+1)}$ is transmitted from the fifth memory layer back to the first layer in a cycle, and further transferred to the subsequent second layer that updates the contents of the third memory layer by new state $\mathbf{y}^{(t+1)}$. Thus, one simulation step is completed and the simulating subnetwork $\mathcal{S}$ is stabilized until x_1 becomes passive again.

Now, the detailed implementation of the simulating subnetwork $\mathcal{S}$ will be presented. As indicated in Figure 3 each of the five layers in $\mathcal{S}$ is composed of n pairs of continuous-time units, one pair for each discrete neuron j in the network $\mathcal{N}$ to be

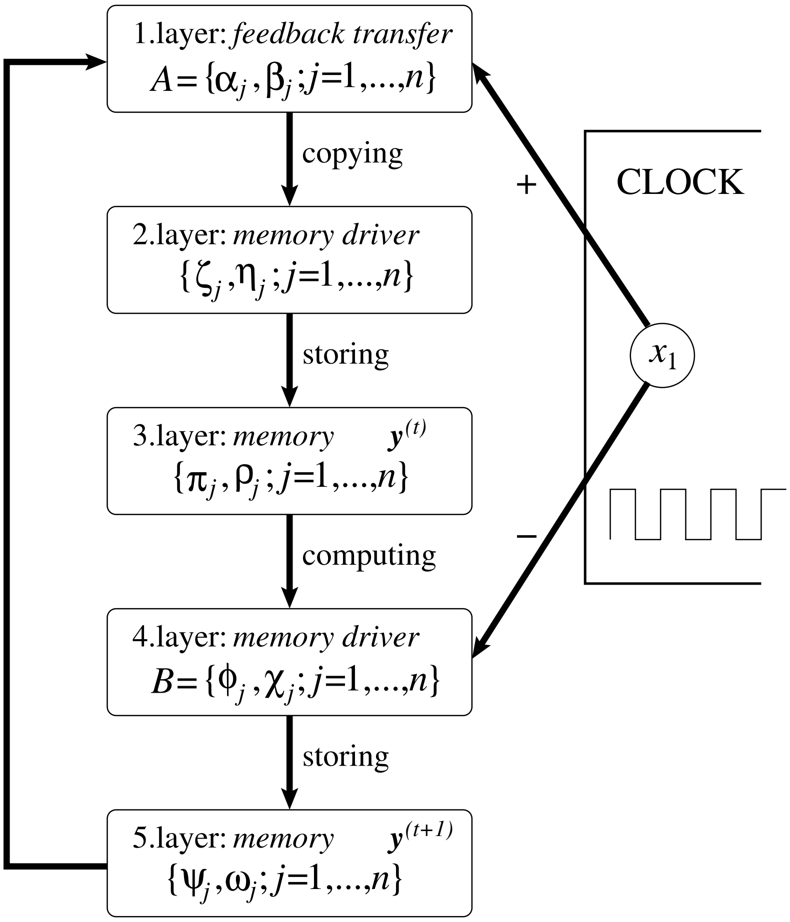


Figure 3: A high-level scheme of the simulation.

simulated. Each system of five unit pairs, one from each layer, that are associated with the same neuron j, create the symmetric gate subnetwork $\mathcal{G}_j$ with 10 units for simulating discrete neuron $j \in \mathcal{N}$ as depicted in Figure 4. Besides the definition of weights in $\mathcal{G}_j$, this figure again shows the interface clock unit $x_1 \in \mathcal{C}$ which is linked to each $\mathcal{G}_j$. In what follows we will informally describe the dynamics of $\mathcal{G}_j$ ($j = 1, \ldots, n$) that simulates the discrete-time updates of neuron $j \in \mathcal{N}$. For simplicity of presentation, we assume (see lemma 2.2b) that particular units in the continuous-time network $\mathcal{G}_j$ approximately follow the discrete update rule 2.1–2.3 when their inputs are saturated for the duration of a certain period.

At the beginning of the simulation, all units in $\mathcal{G}_j$ are placed in *zero* initial states, except possibly for units $\pi_j, \varrho_j \in \mathcal{G}_j$ that doubly encode the initial state of discrete neuron $j \in \mathcal{N}$:

$$y_{\pi_j}(0) = y_{\varrho_j}(0) = y_j^{(0)}\,, \tag{3.11}$$

which means that both units π_j, ϱ_j are initially saturated at value $y_j^{(0)} \in \{0, 1\}$.

The *first phase* of a general simulation step corresponding to the update of state of neuron j at discrete time instant $t+1$ starts when the interface unit $x_1 \in \mathcal{C}$ becomes passive. Consequently, units $\alpha_j, \beta_j \in A$ saturate at 0 and remain passive due to their negative biases (see Figure 4). Only then (lemma 4) the states of units $\phi_j, \chi_j \in B$ are allowed to evolve since the influence of negative weights from x_1

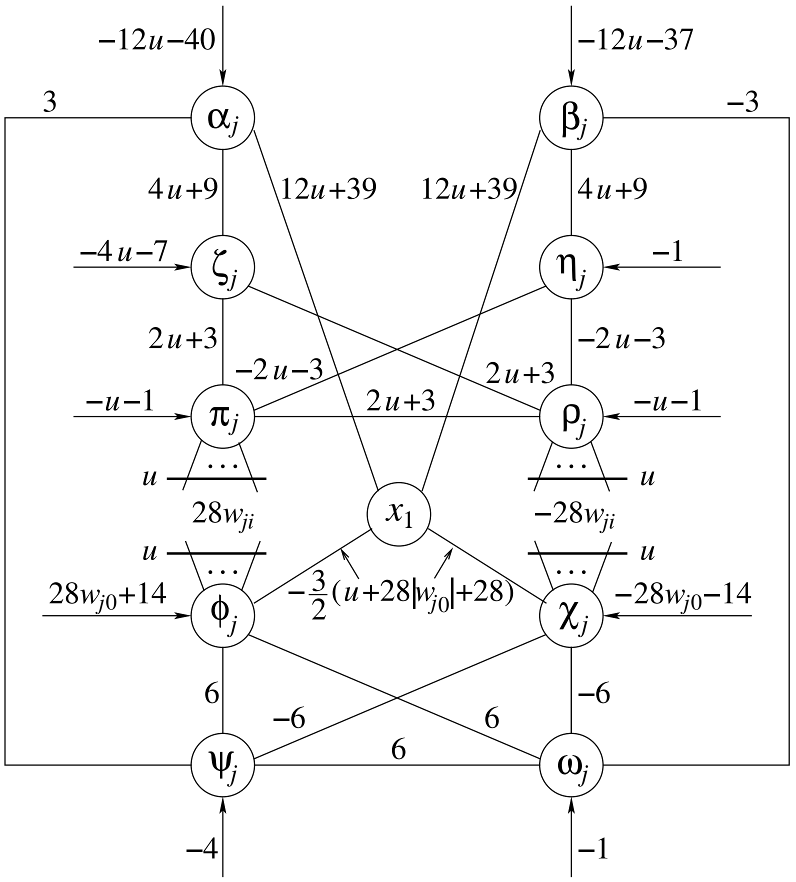


Figure 4: A continuous-time gate network $\mathcal{G}_j$ simulating the discrete neuron j.

disappears with passive x_1. Assume now that the simulated state $y_j^{(t)} \in \{0,1\}$ of neuron $j \in \mathcal{N}$ at discrete time instant t is doubly represented by continuous-time units $\pi_j, \varrho_j \in \mathcal{G}_j$ that are both saturated at value $y_j^{(t)}$. In particular, units π_j, ϱ_j are either both passive due to their negative biases and passive ζ_j, if $y_j^{(t)} = 0$, or they are saturating each other at 1 via their positive coupling weight $v(\pi_j, \varrho_j)$ while η_j is passive, if $y_j^{(t)} = 1$.

The value corresponding to the new state $y_j^{(t+1)} \in \{0,1\}$ of neuron j is computed by unit $\phi_j \in \mathcal{G}_j$, based on its connections to the units $\pi_i \in \mathcal{G}_i$ $(i = 1, \ldots, n)$ that represent the simulated states $y_i^{(t)}$ of neurons $i \in \mathcal{N}$ at the previous discrete time instant t. Therefore, these connections are labeled with symmetric weights $v(\pi_i, \phi_j) = 28w_{ji}$, and unit ϕ_j has bias $v(0, \phi_j) = 28w_{j0} + 14$, where w_{ji} are the original weights of $\mathcal{N}$. The large coefficients are used to guarantee that ϕ_j is not influenced by ψ_j, ω_j while the original function of j is preserved (Parberry, 1994). Similarly, unit χ_j computes the negation of $y_j^{(t+1)}$ from units $\varrho_i \in \mathcal{G}_i$ by using the opposite weights $v(\varrho_i, \chi_j) = -28w(i,j)$ and bias $v(0, \chi_j) = -28w_{j0} - 14$ which also prevent χ_j from being influenced by ψ_j, ω_j.

Note that because units π_i, ϱ_i have biases $v(0, \pi_i) = v(0, \varrho_i) = -u-1$, where u is the large nonnegative parameter defined by formula 3.3, the state evolution of ϕ_j, χ_j cannot reversely affect the dynamics of units π_i, ϱ_i, respectively. In addition, units $\phi_j, \chi_j \in \mathcal{G}_j$ are never simultaneously active, since either they are both passive, or each of them represents the negation of a binary state encoded by the other. Therefore, the positive parameter W introduced in equation 3.2 and implanted in weight $v(a_1, x_1)$ is sufficient for activating the clock unit $x_1 \in \mathcal{C}$ regardless of the total negative influence on x_1 from units in B (see Figure 1).

Furthermore, units ϕ_j, χ_j store the new state $y_j^{(t+1)}$ into both ψ_j and ω_j in such a way that either ϕ_j ensures activating ψ_j, ω_j through positive weights, if $y_j^{(t+1)} = 1$, or χ_j arranges for their reseting by means of negative weights, if $y_j^{(t+1)} = 0$. Consequently, continuous-time units ψ_j, ω_j store the simulated state $y_j^{(t+1)}$ even if ϕ_j, χ_j become later passive, since they are either both passive due to their negative biases, if $y_j^{(t+1)} = 0$, or they are saturating each other at 1 via their positive coupling weight $v(\psi_j, \omega_j)$, if $y_j^{(t+1)} = 1$. Since units $\alpha_j, \beta_j \in A$ are temporarily locked, the gate network $\mathcal{G}_j$ becomes stable until unit x_1 is activated.

In the *second phase*, interface unit $x_1 \in \mathcal{C}$ becomes active and locks units $\phi_j, \chi_j \in B$ by saturating them at 0 before releasing units $\alpha_j, \beta_j \in A$ (lemma 4). Unit α_j then receives $y_j^{(t+1)}$ from unit ψ_j, as the positive support from x_1 balances the negative bias of α_j so that the small positive weight $v(\psi_j, \alpha_j)$ can have an influence on α_j. Similarly, unit β_j computes the negation of the binary state $y_j^{(t+1)}$ received from ω_j by means of the support from x_1. State $y_j^{(t+1)}$ and its negation are further propagated in $\mathcal{G}_j$ from α_j, β_j to ζ_j, η_j, respectively, and these units then update the simulated state of neuron j stored in π_j, ϱ_j by its new value $y_j^{(t+1)}$. Since the negative weights $v(x_1, \phi_j) = v(x_1, \chi_j) = -3(u + 28|w_{j0}| + 28)/2$ from x_1 overcome the total positive influence on ϕ_j, χ_j that originates in their incident units, network $\mathcal{G}_j$ is temporarily stabilized until x_1 becomes passive again, and computing the new discrete state $y_j^{(t+2)}$ is initiated. Recall also that units $\alpha_j, \beta_j \in \mathcal{G}_j$ are never simultaneously active, which implies that the term $-(12u + 39)n$ in definition 3.5 of $v(x'_k, x_1)$ $(k > 1)$ suffices to suppress the total positive influence from A on x_1 (see Figure 1). This completes the construction of the simulating subnetwork $\mathcal{S} = \cup_{j=1}^{n} \mathcal{G}_j$.

4 Formal Verification

Now the correct state evolution of the continuous-time Hopfield network $\mathcal{H}$ described in section 3 needs to be verified. This is achieved by a sequence of lemmas analyzing the behavior of the corresponding system of differential equations 2.6. Lemma 1 first upper bounds the maximum sum of absolute values of weights incident on any unit in $\mathcal{H}$. Lemma 2 then describes explicitly the continuous-time state evolution for saturated units. An analysis of how the decreasing *defects*, i.e. distances from limit values in the states of saturated units, affect the excitation of any other unit reveals that the units in $\mathcal{H}$ actually approximate the discrete update rule 2.1–2.3 of corresponding binary neurons after a certain transient time, provided that the incident saturated units stay saturated. Furthermore, the transfer of the activity in the clock $\mathcal{C}$ from a unit to a subsequent one, when all the incident units are saturated, will be analyzed explicitly in lemma 3. (But note that the dynamics of unit c_0 at time $t = 0$ slightly differs from this analysis.) A crucial fact for the proof of theorem 1 is that the duration time of this transfer turns out to be *constant* and not affected by any initial defect. This introduces a "discrete time" into the clock operation and enables the clock to synchronize the simulation. In lemma 3.2 the result is also partially generalized to the case when some of the incident units may become unsaturated. Finally, lemma 4 will verify that the clock interface unit

$x_1 \in \mathcal{C}$ correctly synchronizes the simulation by network $\mathcal{S}$, i.e. locking the units in A and B always precedes unlocking the units in B and A, respectively.

Lemma 1 *For any unit $p \in \mathcal{H}$ in the Hopfield network constructed in section 3, the sum of absolute values of its incident weights (excluding its local bias) is upper bounded by*

$$\Xi_p = \sum_{q=1}^{m} |v(q,p)| < \varepsilon 2^{1/\varepsilon} \,. \tag{4.1}$$

Proof: Consider first the input weights to the clock interface unit $x_1 \in \mathcal{C}$ as indicated in Figures 1, 2, and 4:

$$\begin{aligned}
\Xi_{x_1} &= v(x_1,x_1) + |v(c_0,x_1)| + v(a_1,x_1) + v(b_1,x_1) \\
&\quad + \sum_{k=2}^{n+1} \left(v(c_k,x_1) + |v(x'_k,x_1)| + v(z_k,x_1) \right) \\
&\quad + \sum_{j=1}^{n} \left(v(\alpha_j,x_1) + v(\beta_j,x_1) + |v(\phi_j,x_1)| + |v(\chi_j,x_1)| \right)
\end{aligned} \tag{4.2}$$

which reduces to

$$\Xi_{x_1} = 3W + 2(12u+39)n + 9 + \varepsilon + 2\sum_{k=2}^{n+1} |v(x'_k,x_1)| \tag{4.3}$$

by using definitions 3.9, 3.2. Weights $v(x'_k,x_1)$ defined by formulas 3.5, 3.7 can be expressed recursively:

$$\begin{aligned}
v(x'_k,x_1) &= v(x'_{k-1},x_1) - \sum_{q \in \mathcal{C}_{k-1} \setminus \mathcal{C}_{k-2};\, v(q,x_1)>0} v(q,x_1) \\
&= v(x'_{k-1},x_1) - v(c_{k-1},x_1) - v(z_{k-1},x_1) = 2v(x'_{k-1},x_1)
\end{aligned} \tag{4.4}$$

for $k = 3, \ldots, n+1$ according to Figure 2 and equation 3.9, whereas

$$\begin{aligned}
v(x'_2,x_1) &= -\left\lceil v(c_2,x_1) + v(x_1,x_1) + v(a_1,x_1) + v(b_1,x_1) \right\rceil - (12u+39)n \\
&= -(W + (12u+39)n + 8)
\end{aligned} \tag{4.5}$$

follows from Figure 1. Hence,

$$v(x'_k,x_1) = -2^{k-2}(W + (12u+39)n + 8) \tag{4.6}$$

for $k = 2, \ldots, n+1$, which can be substituted into formula 4.3:

$$\Xi_{x_1} = 2^{n+1}(W + (12u+39)n + 8) + W - 7 + \varepsilon \,. \tag{4.7}$$

Furthermore, as indicated in Figure 2, the input weights to the clock unit $z_{n+1} \in \mathcal{C}$ of the highest order $n+1$ give:

$$\begin{aligned}
\Xi_{z_{n+1}} &= v(z_{n+1},z_{n+1}) + v(d_{n+1},z_{n+1}) + \sum_{p \in \mathcal{C}_n} v(p,z_k) \\
&= 2V_{n+1} - 2m_{n+1} + \varepsilon = 2V_{n+1} - 16n + 2 + \varepsilon
\end{aligned} \tag{4.8}$$

according to equations 3.10, 3.4. Parameter V_{n+1} will be computed recursively starting with

$$\begin{aligned}
V_2 &= 1 - v(x'_2,x_1) - \sum_{p \in \mathcal{C}_1 \setminus \{x_1\}} v(x_2,p) \\
&= 1 + \sum_{p \in \mathcal{C}_1} \left\lceil v(c_2,p) + \sum_{q \in \mathcal{C}_1;\, v(q,p)>0} v(q,p) \right\rceil + (12u+39)n \\
&= 2W + (12u+39)n + 48
\end{aligned} \tag{4.9}$$

which follows from equations 3.5–3.8 and Figure 1. For $2 < k \le n+1$ the definition 3.8 of V_k can be rewritten as follows:

$$V_k = 1 - v(x'_k, x_1) - \sum_{p \in \mathcal{C}_{k-2} \setminus \{x_1\}} v(x_k, p) - \sum_{p \in \mathcal{C}_{k-1} \setminus \mathcal{C}_{k-2}} v(x_k, p)\,. \tag{4.10}$$

Similarly as in equation 4.4, a recursive formula for the weights $v(x_k, p)$ for $p \in \mathcal{C}_{k-2} \setminus \{x_1\}$ $(k > 2)$ defined by formulas 3.6, 3.7 can be derived:

$$\begin{aligned} v(x_k, p) &= v(x_{k-1}, p) - \sum_{q \in \mathcal{C}_{k-1} \setminus \mathcal{C}_{k-2};\, v(q,p) > 0} v(q, p) \\ &= v(x_{k-1}, p) - v(c_{k-1}, p) - v(z_{k-1}, p) = 2v(x_{k-1}, p) \end{aligned} \tag{4.11}$$

according to Figure 2 and equation 3.9. By introducing formulas 4.4, 4.11 into equation 4.10 and using definition 3.8 the recursive formula for V_k is obtained:

$$V_k = 2V_{k-1} - 1 - \sum_{p \in \mathcal{C}_{k-1} \setminus \mathcal{C}_{k-2}} v(x_k, p)\,. \tag{4.12}$$

The weights $v(x_k, p)$ for $p \in \mathcal{C}_{k-1} \setminus \mathcal{C}_{k-2} = \{c_{k-1}, a_{k-1}, x'_{k-1}, b'_{k-1}, x_{k-1}, b_{k-1}, d_{k-1}, z_{k-1}\}$ in equation 4.12 can be calculated from Figure 2 and by definitions 3.6, 3.7 in which $\lceil v(c_k, p) + v(p, p) \rceil = 3$, as follows:

$$\begin{aligned} -v(x_k, c_{k-1}) &= 3 + v(a_{k-1}, c_{k-1}) + \sum_{q \in \mathcal{C}_{k-2};\, v(q, c_{k-1}) > 0} v(q, c_{k-1}) \\ &= 2m_{k-1} + 3 && (4.13) \\ -v(x_k, a_{k-1}) &= 3 + v(c_{k-1}, a_{k-1}) + v(x'_{k-1}, a_{k-1}) \\ &= -v(x'_{k-1}, x_1) + m_{k-1} + 4 && (4.14) \\ -v(x_k, x'_{k-1}) &= 3 + v(a_{k-1}, x'_{k-1}) + v(b'_{k-1}, x'_{k-1}) \\ &= -v(x'_{k-1}, x_1) + 5 && (4.15) \\ -v(x_k, b'_{k-1}) &= 3 + v(x'_{k-1}, b'_{k-1}) + v(x_{k-1}, b'_{k-1}) \\ &= V_{k-1} + v(x'_{k-1}, x_1) + 4 && (4.16) \\ -v(x_k, x_{k-1}) &= 3 + v(b'_{k-1}, x_{k-1}) + v(b_{k-1}, x_{k-1}) \\ &= V_{k-1} + v(x'_{k-1}, x_1) + 4 && (4.17) \\ -v(x_k, b_{k-1}) &= 3 + v(x_{k-1}, b_{k-1}) + v(d_{k-1}, b_{k-1}) = 5 && (4.18) \\ -v(x_k, d_{k-1}) &= 3 + v(b_{k-1}, d_{k-1}) + v(z_{k-1}, d_{k-1}) \\ &= V_{k-1} - m_{k-1} + 4 && (4.19) \\ -v(x_k, z_{k-1}) &= 3 + v(d_{k-1}, z_{k-1}) + \sum_{q \in \mathcal{C}_{k-2};\, v(q, z_{k-1}) > 0} v(q, z_{k-1}) \\ &= 2V_{k-1} - 2m_{k-1} + 2 && (4.20) \end{aligned}$$

where formula 3.10 has been employed in equation 4.20. Weights 4.13–4.20 are summed up as

$$- \sum_{p \in \mathcal{C}_{k-1} \setminus \mathcal{C}_{k-2}} v(x_k, p) = 5V_{k-1} + 31 \tag{4.21}$$

which is plugged in formula 4.12:

$$V_k = 7V_{k-1} + 30\,. \tag{4.22}$$

It follows from equations 4.9 and 4.22 that

$$V_{n+1} = 7^{n-1} \left(2W + (12u + 39)n + 53\right) - 5 \tag{4.23}$$

which gives

$$\Xi_{z_{n+1}} = 2 \cdot 7^{n-1} \left(2W + (12u + 39)n + 53\right) - 16n - 8 + \varepsilon \tag{4.24}$$

according to equation 4.8.

Observe from Figure 4 that the maximum value of Ξ_p among units $p \in \mathcal{S}$ within the simulating subnetwork is reached by $\alpha_j, \beta_j \in \mathcal{S}$, i.e.

$$\Xi_{\alpha_j} = \Xi_{\beta_j} = 16u + 51 < \Xi_{x_1}\,, \quad j = 1, \ldots, n \tag{4.25}$$

which is still less than Ξ_{x_1} associated with the clock interface unit $x_1 \in \mathcal{C}$ according to equation 4.7. On the other hand, the maximum value of Ξ_p among units $p \in \mathcal{C}$ within the clock subnetwork is reached by unit $z_{n+1} \in \mathcal{C}$ of the highest order $n+1$ for $n \geq 2$, e.g. compare formula 4.8 to

$$\begin{aligned} \Xi_{x_{n+1}} &= 2V_{n+1} - 2v(x'_{n+1}, x_1) + 1 + \varepsilon \\ &= 2V_{n+1} - 2^n(W + (12u + 39)n + 8) + 1 + \varepsilon < \Xi_{z_{n+1}} \end{aligned} \tag{4.26}$$

computed by formula 4.6, while $\Xi_{x_1} > \Xi_{z_2}$ dominates for $n = 1$ according to equations 4.7, 4.24. Hence,

$$\Xi_p \leq \max\left(\Xi_{z_{n+1}}, \Xi_{x_1}\right) \tag{4.27}$$

for every unit $p \in \mathcal{H}$. Clearly,

$$\varepsilon \leq 0.0625 \tag{4.28}$$

follows from assumption 3.1 in which $n \geq 1$, $w_{\max} \geq 1$, and $\varepsilon 2^{1/\varepsilon}$ is decreasing for $\varepsilon > 0$. In addition, the following two upper bounds

$$W \leq 42n^2 w_{\max} + 42n w_{\max} + 42n \tag{4.29}$$

$$u \leq 28n w_{\max} \tag{4.30}$$

are obtained from definitions 3.2 and 3.3, respectively, which together with equations 4.7, 4.24, and inequality 4.28 ensure that condition 4.27 implies

$$\Xi_p < 12 \cdot 7^n \left(10n^2 w_{\max} + 11n w_{\max} + 3n + 2\right) \tag{4.31}$$

which, for $n \geq 1$, gives

$$\Xi_p < w_{\max} 2^{12n} \leq \varepsilon 2^{1/\varepsilon} \tag{4.32}$$

according to assumption 3.1. $\square$

Lemma 2

1. *Let $p \in \mathcal{H}$ be a unit saturated at $b \in \{0, 1\}$ with a* defect

$$\delta_p(t) = |b - y_p(t)|\,, \tag{4.33}$$

for the duration of a continuous time interval $\tau = [t_0, t_f]$ for some $t_0 \geq 0$. Then the state dynamics of p converging towards value b can be explicitly solved as

$$y_p(t) = \left| b - \delta_p e^{-(t-t_0)} \right| \tag{4.34}$$

for $t \in \tau$, where $\delta_p = \delta_p(t_0)$ is p's initial defect.

2a. *Let $Q \subseteq \mathcal{H}$ be a subset of units saturated for the duration of time interval $\tau = [t_0, t_f]$. Then the dynamics of the excitation $\xi_p(t)$ for any unit $p \in \mathcal{H}$ can be described as*

$$\xi_p(t) = v(0,p) + \sum_{q \in Q;\, \xi_q(t) \geq 1} v(q,p) + \sum_{q \in \mathcal{H} \setminus Q} v(q,p) y_q(t) + \Delta_{pQ} e^{-(t-t_0)} \tag{4.35}$$

for $t \in \tau$, where

$$\Delta_{pQ} = \sum_{q \in Q;\, \xi_q(t_0) \leq 0} v(q,p)\delta_q - \sum_{q \in Q;\, \xi_q(t_0) \geq 1} v(q,p)\delta_q \tag{4.36}$$

is the initial total weighted defect *of Q affecting $\xi_p(t_0)$.*

2b. *In addition, let $t_f > t_0 + t_1$ where*

$$t_1 = \frac{\ln 2}{\varepsilon}\,, \tag{4.37}$$

and assume that the respective weights in $\mathcal{H}$ satisfy either

$$v(0,p) + \sum_{q \in Q;\, \xi_q(t_0) \geq 1} v(q,p) + \sum_{q \in \mathcal{H} \setminus Q;\, v(q,p) > 0} v(q,p) < -\varepsilon \tag{4.38}$$

or

$$v(0,p) + \sum_{q \in Q;\, \xi_q(t_0) \geq 1} v(q,p) + \sum_{q \in \mathcal{H} \setminus Q;\, v(q,p) < 0} v(q,p) > 1 + \varepsilon\,. \tag{4.39}$$

Then p is saturated at either 0 *or* 1*, respectively, for the duration of time interval* $[t_0 + t_1, t_f]$.

Proof:
1. According to dynamics 2.5 and 2.6 the state $y_p(t)$ of unit p saturated at $b \in \{0,1\}$ for $t \in \tau$ is independent of outputs from the remaining units and its continuous-time dynamics is described by a differential equation

$$\frac{dy_p}{dt}(t) = -y_p(t) + b \tag{4.40}$$

with a boundary condition

$$y_p(t_0) = |b - \delta_p| \tag{4.41}$$

obtained from formula 4.33 for initial defect δ_p. Hence, its explicit solution 4.34 follows.

2a. The excitation $\xi_p(t)$ of unit p defined by formula 2.7 is split among the contributions from units outside Q and from those in Q saturated at 0 and 1 whose dynamics for $t \in \tau$ is given by equation 4.34:

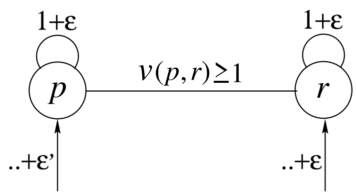
$$\begin{aligned} \xi_p(t) \quad = \quad & v(0,p) + \sum_{q \in \mathcal{H} \setminus Q} v(q,p) y_q(t) + \sum_{q \in Q;\, \xi_q(t) \leq 0} v(q,p)\delta_q e^{-(t-t_0)} \\ & + \sum_{q \in Q;\, \xi_q(t) \geq 1} v(q,p)\left(1 - \delta_q e^{-(t-t_0)}\right)\,. \end{aligned} \tag{4.42}$$

By introducing the initial total weighted defect 4.36 into formula 4.42 the dynamics 4.35 follows.

2b. The defect term $\Delta_{pQ} e^{-(t-t_0)}$ in equation 4.35 vanishes quickly as time proceeds and its absolute value can be bounded for $t \in [t_0 + t_1, t_f]$ as follows

$$\left|\Delta_{pQ} e^{-(t-t_0)}\right| \leq \Xi_p e^{-t_1} < \varepsilon \tag{4.43}$$

by using lemma 1 and equation 4.37. Hence, unit p is saturated at either 0 or 1 for the duration of time interval $[t_0 + t_1, t_f]$ when condition 4.38 or 4.39, respectively, is assumed in equation 4.35. □



o	p	r	unit order
z_1	c_0	c_k	$2 \leq k \leq n+1$
z_k	c_0	c_1	$2 \leq k \leq n+1$
c_0	c_k	a_k	$1 \leq k \leq n+1$
c_1	a_1	x_1	1
c_k	a_k	x'_k	$2 \leq k \leq n+1$
x'_k	b'_k	x_k	$2 \leq k \leq n+1$
x_k	b_k	d_k	$1 \leq k \leq n+1$
b_k	d_k	z_k	$1 \leq k \leq n+1$
d_k	z_k	c_0	$1 \leq k \leq n+1$

Figure 5: Activity transfer from unit p to unit r in the clock subnetwork $\mathcal{C}$.

Lemma 3
1. *Consider a situation as depicted in Figure 5, where a clock unit* $p \in \mathcal{C}$ *with fractional part of bias* $\varepsilon' \in \{\varepsilon, \varepsilon/3\}$ *and feedback weight* $v(p,p) = 1+\varepsilon$ *is supposed to receive a signal from preceding clock unit* $o \in \mathcal{C}$*, activate itself, and further transfer the signal to a subsequent clock unit* $r \in \mathcal{C}$ *with bias fraction* ε *and* $v(r,r) = 1+\varepsilon$ *via weight* $v(p,r) \geq 1$*. Let all the units incident on* p, r *excluding* p, r *be saturated for the duration of some sufficiently large time interval* $\tau = [t_0, t_f]$ *(e.g.* $t_f > t_0 + t_2$ *where* t_2 *is defined by formula 4.49 below), starting at a time* $t_0 > 0$ *when* $\xi_p(t_0) = 0$*. Assume that the initial defects meet*

$$\delta_p + \Delta_{rQ} < \varepsilon \tag{4.44}$$

for $Q = \mathcal{H} \setminus \{p\}$*. Further assume that the respective weights satisfy*

$$v(0,p) + \sum_{q \in Q;\, \xi_q(t_0) \geq 1} v(q,p) = \varepsilon' \tag{4.45}$$

$$v(0,r) + \sum_{q \in Q;\, \xi_q(t_0) \geq 1} v(q,r) = \varepsilon - v(p,r)\,. \tag{4.46}$$

Then p *is unsaturated with the state dynamics*

$$y_p(t) = \frac{\varepsilon' \left(e^{\varepsilon(t-t_0)} - 1\right)}{\varepsilon(1+\varepsilon)} - \frac{\varepsilon' + \Delta_{pQ} e^{-(t-t_0)}}{1+\varepsilon} \tag{4.47}$$

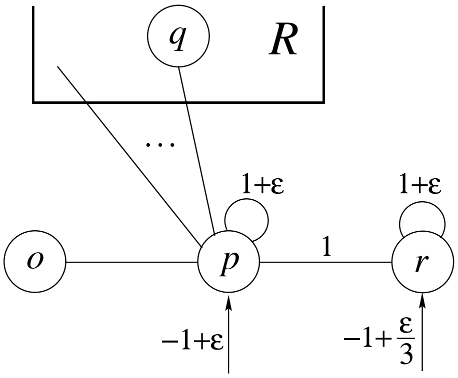
exactly for the duration of time interval $(t_0, t_0 + t'_1)$*, where*

$$t'_1 = \frac{\ln\left(1 + \frac{\varepsilon}{\varepsilon'}\right)}{\varepsilon} \tag{4.48}$$

(note $t'_1 = t_1$ *for* $\varepsilon' = \varepsilon$ *and* $t'_1 = 2t_1$ *for* $\varepsilon' = \varepsilon/3$*), and* r *is saturated at* 0*. In addition,* p *is saturated at* 1 *for the duration of time interval* $[t_0 + t'_1, t_0 + t_2]$ *and remains further saturated independently of the output from* r*, while* r *unsaturates from* 0 *at time* $t_0 + t_2$ *where*

$$t_2 = \ln \frac{v(p,r)\left(\left(\varepsilon + \varepsilon'\right)\left(1 + \frac{\varepsilon}{\varepsilon'}\right)^{1/\varepsilon} + \Delta_{pQ}\right) - (1+\varepsilon)\Delta_{rQ}}{\varepsilon(1+\varepsilon)} \geq t'_1\,. \tag{4.49}$$

2. *Consider a situation as depicted in Figure 6, where a clock unit* $p \in \mathcal{C}$ *with bias* $v(0,p) = -1+\varepsilon$ *and feedback weight* $v(p,p) = 1+\varepsilon$ *is supposed to receive a signal from preceding clock unit* $o \in \mathcal{C}$*, activate itself, and further transfer the signal to a subsequent clock unit* $r \in \mathcal{C}$ *with* $v(0,r) = -1+\varepsilon/3$ *and* $v(r,r) = 1+\varepsilon$*, via*



o	p	r	unit order	R
a_1	x_1	b_1	1	$\{c_0\} \cup A \cup B_{t_0}$
a_k	x'_k	b'_k	$2 \le k \le n+1$	$\{x_1\}$
b'_k	x_k	b_k	$2 \le k \le n+1$	$\mathcal{C}_{k-1} \setminus \{x_1\}$

Figure 6: Activity transfer from unit p to r when units $q \in R$ unsaturate.

weight $v(p,r) = 1$, while the units in a given subset $R \subseteq \mathcal{H}$, incident on p (with no connections to r) may unsaturate after p unsaturates from 0. Let all the units incident on p, r, excluding p, r, and R, be saturated for the duration of a sufficiently large time interval $\tau = [t_0, t_f]$ (e.g. at least until r unsaturates from 0) starting at a time $t_0 > 0$ when $\xi_p(t_0) = 0$. In Figure 6,

$$B_{t_0} = \{q \in B;\, \xi_q(t_0) \ge 1\} \subseteq \mathcal{S} \tag{4.50}$$

denotes the subset of units from simulating subnetwork $\mathcal{S}$ that are saturated at 1 at time t_0. Assume that the initial defects meet

$$\begin{aligned} \delta_r &< \varepsilon 2^{-1/\varepsilon} && (4.51) \\ \Delta_{rQ'} &< \varepsilon 2^{-1/\varepsilon} && (4.52) \end{aligned}$$

for $Q' = \mathcal{H} \setminus (R \cup \{p\})$, and

$$(1+\varepsilon)\delta_p + \sum_{q \in R;\, \xi_q(t_0) \le 0} v(q,p)\delta_q - \sum_{q \in R;\, \xi_q(t_0) \ge 1} v(q,p)\delta_q \le \varepsilon 2^{-1/\varepsilon} \tag{4.53}$$

outside Q'. Further, assume that the respective weights satisfy

$$\begin{aligned} v(0,p) + \sum_{q \in Q';\, \xi_q(t_0) \ge 1} v(q,p) + \sum_{q \in R;\, v(q,p) < 0} v(q,p) &= \varepsilon && (4.54) \\ \sum_{q \in Q';\, \xi_q(t_0) \ge 1} v(q,r) &= 0\,. && (4.55) \end{aligned}$$

Then p saturates at 1 in time at most $t_0 + 2t_1$, remaining then saturated until time at least t_f, and r unsaturates from 0 only after p is saturated at 1.

Proof:

1. A summary of the dynamics of units p, r under discussion here is presented in Table 1, which is verified step by step below.

Excitation

$$\xi_p(t) = \varepsilon' + (1+\varepsilon) y_p(t) + \Delta_{pQ} e^{-(t-t_0)} \tag{4.56}$$

<table>
<tr><th>t</th><th>$\xi_p(t)$</th><th>$y_p(t)$</th><th>$\frac{dy_p}{dt}(t)$</th><th>$\xi_r(t)$</th><th>$y_r(t)$</th></tr>
<tr><td>$= t_0$</td><td>$= 0$</td><td>$= \delta_p$</td><td rowspan="2">< 0</td><td rowspan="6">< 0</td><td>$= \delta_r$</td></tr>
<tr><td>$\in (t_0, t_0 + t_g)$</td><td rowspan="3">> 0</td><td rowspan="3">eq. 4.47</td><td rowspan="6">$= \delta_r e^{-(t-t_0)}$</td></tr>
<tr><td>$= t_0 + t_g$</td><td>$= 0$</td></tr>
<tr><td>$\in (t_0 + t_g, t_0 + t_1')$</td><td rowspan="5">$> 0$</td></tr>
<tr><td>$= t_0 + t_1'$</td><td>$= 1$</td><td>eq. 4.64</td></tr>
<tr><td>$\in (t_0 + t_1', t_0 + t_2)$</td><td rowspan="3">$> 1$</td><td rowspan="3">eq. 4.70</td></tr>
<tr><td>$= t_0 + t_2$</td><td>$= 0$</td></tr>
<tr><td>$\in (t_0 + t_2, t_f]$</td><td>> 0</td><td>—</td></tr>
</table>

Table 1: The chronology of the dynamics of units p, r.

of unit p at time $t \in [t_0, t_0 + t_2]$ is derived from equation 4.35 by assumption 4.45 and determines p's state dynamics 2.6 by differential equation

$$\frac{dy_p}{dt}(t) = -y_p(t) + \varepsilon' + (1+\varepsilon)y_p(t) + \Delta_{pQ} e^{-(t-t_0)} \tag{4.57}$$

when p is unsaturated. The corresponding boundary condition

$$y_p(t_0) = \frac{-\varepsilon' - \Delta_{pQ}}{1+\varepsilon} = \delta_p \tag{4.58}$$

comes from $\xi_p(t_0) = 0$ that is applied to formula 4.56 and determines also the initial defect

$$-\Delta_{pQ} = \varepsilon' + (1+\varepsilon)\delta_p \tag{4.59}$$

which can be bounded as

$$-1 - \varepsilon - \varepsilon' \leq \Delta_{pQ} \leq -\varepsilon' < 0 \tag{4.60}$$

due to $1 \geq \delta_p \geq 0$. The explicit solution 4.47 for differential equation 4.57 follows when initial condition 4.58 is provided.

By plugging solution 4.47 in equation 4.57 an explicit formula for p's state derivative can be calculated:

$$\frac{dy_p}{dt}(t) = \frac{e^{-(t-t_0)} \left(\varepsilon' e^{(1+\varepsilon)(t-t_0)} + \Delta_{pQ}\right)}{1+\varepsilon}\,. \tag{4.61}$$

It follows that before the state $y_p(t)$ of unit p starts to grow it is initially nonincreasing within time $t \in [t_0, t_0 + t_g]$ where

$$t_g = \frac{\ln \frac{-\Delta_{pQ}}{\varepsilon'}}{1+\varepsilon} \tag{4.62}$$

since p's state derivative 4.61 is nonpositive for $t \in [t_0, t_0 + t_g]$, and $t_g \geq 0$ according to inequality 4.60.

Further, by introducing solution 4.47 into equation 4.56 the dynamics of p's excitation can also be expressed explicitly:

$$\xi_p(t) = \frac{\varepsilon' \left(e^{\varepsilon(t-t_0)} - 1\right)}{\varepsilon} > 0\,. \tag{4.63}$$

This ensures that unit p is unsaturated for the duration of the whole time interval $(t_0, t_0 + t_1')$ even though its state $y_p(t)$ is initially decreasing for $t \in (t_0, t_0 + t_g)$. Eventually unit p saturates at 1 exactly at time instant $t_0 + t_1'$ where t_1' is given by formula 4.48 that is derived from equation 4.63 for $\xi_p(t_0+t_1')=1$. The state of unit p at t_0+t_1' can be computed by substituting formula 4.48 into equation 4.47:

$$y_p(t_0+t_1') = \frac{1-\varepsilon' - \Delta_{pQ}\left(1+\frac{\varepsilon}{\varepsilon'}\right)^{-1/\varepsilon}}{1+\varepsilon} \tag{4.64}$$

Also $t_1' > t_g$ according to inequality 4.28 which confirms the actual growth of p's state. Notice that the length t_1' of the period when p is unsaturated is *constant and independent of the initial defects.* This introduces a notion of "discrete time" into the clock operation based on t_1. Recall that the detailed chronology of the dynamics of units p, r during the activity transfer is shown in Table 1.

Similarly, excitation

$$\xi_r(t) = \varepsilon - v(p,r) + v(p,r)y_p(t) + \Delta_{rQ}e^{-(t-t_0)} \tag{4.65}$$

of unit r in time $t \in [t_0, t_0+t_2]$ is derived from equation 4.35 by assumption 4.46. In order to verify that the state dynamics of unit p is indeed controlled by equation 4.57 for the duration of time interval (t_0, t_0+t_1') it must also be checked that unit r remains saturated at 0 in this period, that is $\xi_r(t) \le 0$. Since $v(p,r) \ge 1$, according to equation 4.65 it suffices to show

$$\xi_r(t) \le \varepsilon + y_p(t) - 1 + \Delta_{rQ}e^{-(t-t_0)} \le 0 \tag{4.66}$$

for all $t \in [t_0, t_0+t_1']$ which can be rewritten as

$$\varepsilon\left(\varepsilon' + (1+\varepsilon)(\delta_p + \Delta_{rQ})\right)e^{-(t-t_0)} + \varepsilon'\left(e^{\varepsilon(t-t_0)} - 1\right) - \varepsilon \le \varepsilon\left(\varepsilon' - \varepsilon^2\right) \tag{4.67}$$

by substituting solution 4.47 for $y_p(t)$ in which $-\Delta_{pQ}$ is replaced by formula 4.59. Inequality 4.67 further reduces to

$$\varepsilon\left(\varepsilon' + \varepsilon(1+\varepsilon)\right)e^{-(t-t_0)} + \varepsilon'\left(e^{\varepsilon(t-t_0)} - 1\right) - \varepsilon \le \varepsilon\left(\varepsilon' - \varepsilon^2\right) \tag{4.68}$$

due to assumption 4.44. For $t \in [t_0, t_0+t_\varepsilon]$ where

$$t_\varepsilon = \ln\frac{\varepsilon' + \varepsilon(1+\varepsilon)}{\varepsilon' - \varepsilon^2}\,, \tag{4.69}$$

term $e^{\varepsilon(t-t_0)}$ reaches its maximum at time instant t_0+t_ε whereas $e^{-(t-t_0)} \le 1$, which implies inequality 4.68 by using condition 4.28. For $t \in [t_0+t_\varepsilon, t_0+t_1']$, on the other hand, term $\varepsilon(\varepsilon'+\varepsilon(1+\varepsilon))e^{-(t-t_0)}$ in inequality 4.68 achieves its maximum $\varepsilon(\varepsilon'-\varepsilon^2)$ at time instant t_0+t_ε while $\varepsilon'(e^{\varepsilon(t-t_0)}-1)-\varepsilon \le 0$ reaches 0 at time instant t_0+t_1'. Hence, unit r is saturated at 0 within the period (t_0, t_0+t_1') when unit p is unsaturated.

The state $y_p(t)$ of unit p saturated at 1 follows further the dynamics equation 4.34, that is

$$\begin{aligned} y_p(t) &= 1 - \delta_p(t_0+t_1')e^{-(t-t_0-t_1')} \\ &= 1 - \frac{\left(\varepsilon+\varepsilon'+\Delta_{pQ}\left(1+\frac{\varepsilon}{\varepsilon'}\right)^{-1/\varepsilon}\right)e^{-(t-t_0-t_1')}}{1+\varepsilon} \end{aligned} \tag{4.70}$$

for $t \in [t_0+t_1', t_f]$ where the corresponding defect $\delta_p(t_0+t_1') = 1 - y_p(t_0+t_1')$ is calculated from formula 4.64. By substituting formula 4.70 into equation 4.56 the dynamics of p's excitation is obtained:

$$\xi_p(t) = 1 + (\varepsilon+\varepsilon')\left(1 - e^{-(t-t_0-t_1')}\right) \ge 1 \tag{4.71}$$

for $t \in [t_0 + t'_1, t_0 + t_2]$ which confirms that unit p remains saturated at 1 at least until unit r becomes unsaturated from 0. Also excitation $\xi_r(t)$ of unit r saturated at 0 after p saturates at 1 can be expressed by introducing formula 4.70 into equation 4.65 as follows:

$$\xi_r(t) = \varepsilon - v(p,r)\delta_p(t_0 + t'_1)e^{-(t-t_0-t'_1)} + \Delta_{rQ}e^{-(t-t_0)} \tag{4.72}$$

that reaches 0 at time instant $t_0 + t_2$, i.e. $\xi_r(t_0 + t_2) = 0$ which gives formula 4.49 for t_2 by using equation 4.48. Hence, unit r is unsaturated from 0 after time $t_0 + t_2$ and the dynamics of r's state is further described by a differential equation of the form 4.57. In this way, the activity of unit p is transferred to r.

Finally, it must be checked that unit p remains still saturated at 1 when r is unsaturated from 0. For this purpose, excitation 4.56 of unit p can be rewritten as

$$\begin{aligned} \xi_p(t) \;\; = \;\; & \varepsilon' + 1 + \varepsilon + v(p,r)y_r(t) \\ & -(1+\varepsilon)\delta_p(t_0+t_2)e^{-(t-t_0-t_2)} + \left(\Delta_{pQ} - v(p,r)\delta_r\right)e^{-(t-t_0)} \end{aligned} \tag{4.73}$$

for $t \in [t_0 + t_2, t_f]$ according to equation 4.35 in which the subset of saturated units Q is now replaced with $Q_1 = \mathcal{H} \setminus \{r\}$ while the initial defects δ_r, Δ_{pQ} in equation 4.73 are still related to time instant t_0 and $Q = \mathcal{H} \setminus \{p\}$. The defect $\delta_p(t_0 + t_2)$ at time instant $t_0 + t_2$ of unit p saturated at 1 can be calculated from equation 4.70:

$$\delta_p(t_0 + t_2) = 1 - y_p(t_0 + t_2) = \frac{\varepsilon}{v(p,r) - \frac{(1+\varepsilon)\Delta_{rQ}}{(\varepsilon+\varepsilon')\left(1+\frac{\varepsilon}{\varepsilon'}\right)^{1/\varepsilon} + \Delta_{pQ}}} \tag{4.74}$$

where formula 4.49 is used. In order to prove that $\xi_p(t) \geq 1$ for all $t \in [t_0 + t_2, t_f]$ the underlying negative defect terms in equation 4.73 having the least value for $t = t_0 + t_2$ will be lower bounded by $-\varepsilon' - \varepsilon$ whereas $v(p,r)y_r(t) \geq 0$ is neglected assuring that unit p remains saturated at 1 independently of the output from r. Thus, it is sufficient to prove

$$\begin{aligned} & -(1+\varepsilon)\delta_p(t_0+t_2) + \left(\Delta_{pQ} - v(p,r)\delta_r\right)e^{-t_2} \\ & = \frac{-\varepsilon(1+\varepsilon)\left((\varepsilon+\varepsilon')\left(1+\frac{\varepsilon}{\varepsilon'}\right)^{1/\varepsilon} + v(p,r)\delta_r\right)}{v(p,r)\left((\varepsilon+\varepsilon')\left(1+\frac{\varepsilon}{\varepsilon'}\right)^{1/\varepsilon} + \Delta_{pQ}\right) - (1+\varepsilon)\Delta_{rQ}} > -\varepsilon' - \varepsilon \end{aligned} \tag{4.75}$$

where formulas 4.74, 4.49 have been employed, which further reduces to

$$\begin{aligned} & (1+\varepsilon)(\varepsilon+\varepsilon')\left(\varepsilon\left(1+\frac{\varepsilon}{\varepsilon'}\right)^{1/\varepsilon} + \Delta_{rQ}\right) \\ & < v(p,r)\left((\varepsilon+\varepsilon')^2\left(1+\frac{\varepsilon}{\varepsilon'}\right)^{1/\varepsilon} + (\varepsilon+\varepsilon')\Delta_{pQ} - \varepsilon(1+\varepsilon)\delta_r\right). \end{aligned} \tag{4.76}$$

According to assumption 4.44 and equation 4.59 it suffices to show inequality 4.76 with Δ_{rQ} and Δ_{pQ} replaced by ε and $-\varepsilon' - (1+\varepsilon)\varepsilon$, respectively. In addition, $v(p,r) \geq 1$ and $\delta_r \leq 1$, which leads to

$$\begin{aligned} & (1+\varepsilon)(\varepsilon+\varepsilon')\left(\varepsilon\left(1+\frac{\varepsilon}{\varepsilon'}\right)^{1/\varepsilon} + \varepsilon\right) \\ < \;\; & (\varepsilon+\varepsilon')^2\left(1+\frac{\varepsilon}{\varepsilon'}\right)^{1/\varepsilon} - \varepsilon'(\varepsilon+\varepsilon') - \varepsilon(1+\varepsilon)(1+\varepsilon+\varepsilon') \end{aligned} \tag{4.77}$$

that holds due to inequality 4.28. This completes the argument for unit p to be saturated at 1 after r becomes unsaturated from 0.

2. Note that unit o saturates at 1 according to case 1 of this lemma before p is unsaturated from 0 at time instant t_0. Excitation of unit p derived from equation 4.35 can be lower bounded as

$$\xi_p(t) \geq \varepsilon + (1+\varepsilon)y_p(t) + \Delta_{pQ'}\, e^{-(t-t_0)} \tag{4.78}$$

for $t \in \tau$ by assumption 4.54. According to dynamics equation 2.6 this also provides the following lower bound on the state derivative of unsaturated p:

$$\frac{dy_p}{dt}(t) \geq \varepsilon y_p(t) + \varepsilon + \Delta_{pQ'}\, e^{-(t-t_0)}\,. \tag{4.79}$$

Moreover, in the beginning of time interval τ, the state evolution of unit p is determined by formula 4.47 before the first unit $q \in R$ becomes unsaturated, since assumption 4.45 coincides with assumption 4.54 due to $\varepsilon' = \varepsilon$ and $\xi_q(t_0) \geq 1$ for all $q \in R$ with $v(q,p) < 0$ (see table in Figure 6). Also the initial total weighted defect $\Delta_{pQ'}$ for $Q' = \mathcal{H} \setminus (R \cup \{p\})$ can be expressed in terms of $\Delta_{pQ} = -\varepsilon - (1+\varepsilon)\delta_p$ for $Q = \mathcal{H} \setminus \{p\}$ from equation 4.59 as follows:

$$\Delta_{pQ'} = -\varepsilon - (1+\varepsilon)\delta_p - \sum_{q \in R;\, \xi_q(t_0) \leq 0} v(q,p)\delta_q + \sum_{q \in R;\, \xi_q(t_0) \geq 1} v(q,p)\delta_q \tag{4.80}$$

according to definition 4.36. Hence,

$$\Delta_{pQ'} \geq -\varepsilon \left(1 + 2^{-1/\varepsilon}\right) \tag{4.81}$$

by assumption 4.53.

By introducing inequality 4.81 into formula 4.79 the state derivative of unsaturated p is further lower bounded as

$$\frac{dy_p}{dt}(t) \geq \varepsilon y_p(t) + \varepsilon - \varepsilon \left(1 + 2^{-1/\varepsilon}\right) e^{-(t-t_0)}\,. \tag{4.82}$$

Since $\varepsilon y_p(t) \geq 0$, it follows that

$$\frac{dy_p}{dt}(t) \geq \varepsilon - \varepsilon^2 > 0 \tag{4.83}$$

for $t \geq t_0 + t_d$ where

$$t_d = \ln \frac{1 + 2^{-1/\varepsilon}}{\varepsilon}\,, \tag{4.84}$$

provided that p is still unsaturated. This implies that $y_p(t)$ for $t \geq t_0 + t_d$ grows at least as fast as the straight line with equation

$$\left(\varepsilon - \varepsilon^2\right)(t - t_0 - t_d) - y = 0 \tag{4.85}$$

until unit p saturates at 1. Hence, p saturates at 1 certainly before $t_0 + t_d + t_s < t_0 + 2t_1$ where

$$t_s = \frac{1}{\varepsilon - \varepsilon^2} \tag{4.86}$$

because $\xi_p(t) > y_p(t)$ from equation 2.6 due to p's state derivative 4.83 is positive for $t \geq t_0 + t_d$.

In addition, it will be proved that the subsequent unit r is saturated at 0 at least until p saturates at 1. Excitation

$$\xi_r(t) = -1 + \frac{\varepsilon}{3} + y_p(t) + \Delta_{rQ'}\, e^{-(t-t_0)} \tag{4.87}$$

of unit r saturated at 0 can be derived from equation 4.35 by assumption 4.55. Let $t_y > 0$ be the least local time instant at which

$$y_p(t_0 + t_y) = 1 - \frac{\varepsilon}{3} - \Delta_{rQ'}\, e^{-t_y} \tag{4.88}$$

when r is still saturated at 0 since $\xi_r(t_0 + t_y) = 0$ follows from equation 4.87. Thus, it suffices to prove that the excitation of unit p can be lower bounded at time instant $t_0 + t_y$ as

$$\xi_p(t_0 + t_y) \geq \varepsilon + (1+\varepsilon)\left(1 - \frac{\varepsilon}{3} - \Delta_{rQ'}\, e^{-t_y}\right) + \Delta_{pQ'}\, e^{-t_y} \geq 1 \tag{4.89}$$

according to inequality 4.78. By substituting the error bounds 4.52, 4.81 this reduces to

$$5 - \varepsilon - 3\left(1 + (2+\varepsilon)2^{-1/\varepsilon}\right) e^{-t_y} \geq 0 \tag{4.90}$$

which holds for condition 4.28 due to $e^{-t_y} < 1$, ensuring that unit p is already saturated at 1 at time instant $t_0 + t_y$.

Finally, it must be checked that unit p remains saturated at 1 after $t_0 + t_y$ when unit r may become unsaturated from 0. Inequality 4.78 reads now as

$$\begin{aligned}\xi_p(t) \quad \geq \quad & \varepsilon + (1+\varepsilon)\left(1 - \left(\frac{\varepsilon}{3} + \Delta_{rQ'}\, e^{-t_y}\right) e^{-(t-t_0-t_y)}\right) + y_r(t) \\ & + (\Delta_{pQ'} - \delta_r)\, e^{-(t-t_0)}\end{aligned} \tag{4.91}$$

since the state dynamics of unit p saturated at 1 is controlled by equation 4.34. In order to prove that $\xi_p(t) \geq 1$ for all $t \in (t_0 + t_y, t_f]$ it suffices to show

$$6 - (1+\varepsilon)e^{-(t-t_0-t_y)} - 3\left(1 + (3+\varepsilon)2^{-1/\varepsilon}\right) e^{-(t-t_0)} \geq 0 \tag{4.92}$$

according to inequality 4.91 in which $y_r(t) \geq 0$ and the error bounds 4.51, 4.52, and 4.81 have been applied. Inequality 4.92 follows from $e^{-(t-t_0)} \leq e^{-(t-t_0-t_y)} \leq 1$ and condition 4.28. This completes the argument for unit p to be saturated at 1 after r becomes unsaturated from 0. □

Lemma 4 *All the units in $A \cup B \subseteq \mathcal{S}$ are saturated at 0 when the state of the clock interface unit $x_1 \in \mathcal{C}$ equals $2/3$.*

1. *The state of unit x_1 unsaturated from 0 at time $t_0 \geq 0$ (i.e. $\xi_{x_1}(t_0) = 0$) after reaching value $y_{x_1}(t_0 + t'_\ell) = 2/3$ from below at time $t_0 + t'_\ell$, remains further increasing at least until x_1 is again unsaturated from 1 by the next x'_k unit ($2 \leq k \leq n+1$).*

2. *Let unit x'_k ($2 \leq k \leq n+1$) cease its saturation at 0 at time $t_0 \geq 0$ when $\xi_{x'_k}(t_0) = 0$. Following this, let unit x_1 unsaturate from 1 immediately after time instant $t_0 + t_u > t_0$, where $\xi_{x_1}(t_0 + t_u) = 1$, and reach its state value $y_{x_1}(t_0 + t_\ell) = 2/3$ from above at time $t_0 + t_\ell$ ($t_u < t_\ell$). Let*

$$\Delta_{x_1 Q_1} > -\varepsilon \tag{4.93}$$

for $Q_1 = \mathcal{H} \setminus \{x'_k\}$ and

$$\Delta_{x_1 Q'_1} > -\varepsilon 2^{-1/\varepsilon} \tag{4.94}$$

for $Q'_1 = \mathcal{H} \setminus (A_{t_0} \cup B \cup \{x_1, x'_k\})$ where

$$A_{t_0} = \{q \in A;\, \xi_q(t_0) \geq 1\}\ . \tag{4.95}$$

Further assume that

$$\begin{aligned}\delta_{x_1} &< \varepsilon 2^{-1/\varepsilon} &(4.96)\\ \sum_{q\in A_{t_0}} v(q,x_1)\delta_q &< \varepsilon 2^{-1/\varepsilon} &(4.97)\\ \sum_{q\in B} v(q,x_1)\delta_q &> -\varepsilon 2^{-1/\varepsilon}\,. &(4.98)\end{aligned}$$

Then the state of unit x_1 keeps decreasing after time t_0+t_ℓ at least until x_1 is again unsaturated from 0 *by a_1.*

Proof: It can first be easily checked from Figure 4 that the excitation of all $\alpha_j,\beta_j,\phi_j,\chi_j\in A\cup B$ for $j=1,\ldots,n$ is negative when $y_{x_1}=2/3$:

$$\begin{aligned}\xi_{\alpha_j} &\le v(0,\alpha_j)+v(\psi_j,\alpha_j)+v(\zeta_j,\alpha_j)+v(x_1,\alpha_j)\cdot\frac{2}{3}=-2 &(4.99)\\ \xi_{\beta_j} &\le v(0,\beta_j)+v(\eta_j,\beta_j)+v(x_1,\beta_j)\cdot\frac{2}{3}=-2 &(4.100)\\ \xi_{\phi_j} &\le v(0,\phi_j)+u+v(\psi_j,\phi_j)+v(\omega_j,\phi_j)+v(x_1,\phi_j)\cdot\frac{2}{3}\le -2 &(4.101)\\ \xi_{\chi_j} &\le v(0,\chi_j)+u+v(x_1,\chi_j)\cdot\frac{2}{3}\le -42\,. &(4.102)\end{aligned}$$

1. Obviously, $y_{x_1}(t)$ is further increasing for $t\ge t_0+t_\ell$ since $(dy_{x_1}/dt)(t)>0$ for $y_{x_1}(t_0+t_\ell)=2/3$ according to inequality 4.82, and hence, unit x_1 then saturates at 1.

2. Excitation

$$\begin{aligned}\xi_{x_1}(t_0+t_u) &= W+(12u+39)n+3+2\varepsilon+v(x'_k,x_1)y_{x'_k}(t_0+t_u)\\ &\quad +\Delta_{x_1Q_1}\,e^{-t_u}=1 &(4.103)\end{aligned}$$

of unit x_1 at t_0+t_u is derived from equation 4.35 for Q_1 where

$$\begin{aligned}v(0,x_1)+\sum_{q\in Q_1;\,\xi_q(t)\ge 1} v(q,x_1) &= v(0,x_1)+v(x_1,x_1)+v(c_0,x_1)+v(a_1,x_1)\\ &\quad +v(b_1,x_1)+v(c_k,x_1)+\sum_{q\in A_{t_0}} v(q,x_1)\\ &= W+(12u+39)n+3+2\varepsilon &(4.104)\end{aligned}$$

follows from Figures 1, 2, definition 3.9, and from the fact that $\alpha_j\in A_{t_0}$ iff $\beta_j\notin A_{t_0}$, for all $j=1,\ldots,n$ (see paragraph 3.2). Equation 4.103 gives

$$y_{x'_k}(t_0+t_u)\ge\frac{W+(12u+39)n+2+\varepsilon}{-v(x'_k,x_1)} \tag{4.105}$$

by assumption 4.93 and $v(x'_k,x_1)<0$. We already know from inequality 4.82 where $e^{-(t-t_0)}<1$ that

$$\frac{dy_{x'_k}}{dt}(t)\ge\varepsilon y_{x'_k}(t)-\varepsilon 2^{-1/\varepsilon}\,. \tag{4.106}$$

We prove that the state $y_{x'_k}(t)$ of unit x'_k keeps increasing after x_1 is unsaturated from 1 for $t>t_0+t_u$. Since $y_{x'_k}$ is continuous, it suffices to show that its derivative satisfies

$$\frac{dy_{x'_k}}{dt}(t_0+t_u)>0 \tag{4.107}$$

$t =$	$\xi_{x'_k}(t)$	$\frac{dy_{x'_k}}{dt}(t)$	$\xi_{x_1}(t)$	$y_{x_1}(t)$	$\frac{dy_{x_1}}{dt}(t)$
t_0	$= \mathbf{0}$	< 0	> 1	$= 1 - \delta_{x_1}$	> 0
$t_0 + t_u$			$= \mathbf{1}$	$= 1 - \delta_{x_1} e^{-t_u}$	
$t_0 + t'_d$	—	> 0	—	—	$= \mathbf{0}$
$t_0 + t_\ell$				$= \mathbf{\frac{2}{3}}$	< 0

Table 2: The review of the boundary time instants for units x'_k, x_1.

at time instant $t_0 + t_u$ according to inequality 4.106, which further reduces to

$$\frac{dy_{x'_k}}{dt}(t_0 + t_u) \geq \frac{\varepsilon\,(W + (12u + 39)n + 2 + \varepsilon)}{-v(x'_k, x_1)} - \varepsilon 2^{-1/\varepsilon} > 0 \tag{4.108}$$

by using condition 4.105. Inequality 4.108 follows from $-v(x'_k, x_1) < \varepsilon 2^{1/\varepsilon}$ according to lemma 1.

Similarly, excitation

$$\begin{aligned} \xi_{x_1}(t) &= v(0, x_1) + \sum_{q \in Q'_1;\, \xi_q(t) \geq 1} v(q, x_1) + (1 + \varepsilon) y_{x_1}(t) + v(x'_k, x_1) y_{x'_k}(t) \\ &\quad + \sum_{q \in A_{t_0}} v(q, x_1) y_q(t) + \sum_{q \in B} v(q, x_1) y_q(t) + \Delta_{x_1 Q'_1}\, e^{-(t - t_0)} \end{aligned} \tag{4.109}$$

of unit x_1 for time $t \geq t_0$ is obtained from equation 4.35 for Q'_1. Hence, the derivative of unsaturated x_1 for $t \geq t_0 + t_u$ can be expressed as

$$\begin{aligned} \frac{dy_{x_1}}{dt}(t) &= v(0, x_1) + \sum_{q \in Q'_1;\, \xi_q(t) \geq 1} v(q, x_1) + \varepsilon y_{x_1}(t) + v(x'_k, x_1) y_{x'_k}(t) \\ &\quad + \sum_{q \in A_{t_0}} v(q, x_1) y_q(t) + \sum_{q \in B} v(q, x_1) y_q(t) + \Delta_{x_1 Q'_1}\, e^{-(t - t_0)} \end{aligned} \tag{4.110}$$

according to equation 2.6. Denote by $0 < t'_d < t_\ell$ the least local time instant such that

$$\frac{dy_{x_1}}{dt}(t_0 + t'_d) = 0\,. \tag{4.111}$$

Note that $t'_d > t_u$ since unit x_1 is still saturated at 1 at time $t_0 + t_u$ when its state y_{x_1} is increasing according to equation 4.34. The values defining the boundary time instants (printed in bold) for the variables associated with units x'_k, x_1 and their signs or dynamics are summarized in Table 2.

Now, the change in values of particular terms in the state derivative 4.110 of unit x_1 will be estimated between time instants $t_0 + t_\ell$ and $t_0 + t'_d$. Thus,

$$y_{x_1}(t_0 + t'_d) > y_{x_1}(t_0) = 1 - \delta_{x_1} > 1 - \varepsilon 2^{-1/\varepsilon} > \frac{2}{3} = y_{x_1}(t_0 + t_\ell) \tag{4.112}$$

by assumption 4.96, which implies

$$\varepsilon y_{x_1}(t_0 + t_\ell) - \varepsilon y_{x_1}(t_0 + t'_d) < -\frac{\varepsilon}{3} + \varepsilon^2 2^{-1/\varepsilon}\,. \tag{4.113}$$

Since $v(x'_k, x_1) < 0$ and $y_{x'_k}(t)$ is increasing for $t \geq t_0 + t_u$ it follows

$$v(x'_k, x_1) y_{x'_k}(t) - v(x'_k, x_1) y_{x'_k}(t_0 + t'_d) < 0 \tag{4.114}$$

for $t \geq t_0 + t'_d$. In addition, $v(q, x_1) > 0$ for $q \in A_{t_0} \subseteq A$ and all the units in A_{t_0} are still saturated at 1 at $t_0 + t'_d$ because $y_{x_1}(t)$ is nondecreasing for $t_0 \leq t \leq t_0 + t'_d$, which gives

$$\sum_{q \in A_{t_0}} v(q, x_1) y_q(t) - \sum_{q \in A_{t_0}} v(q, x_1) y_q(t_0 + t'_d) < \varepsilon 2^{-1/\varepsilon} \tag{4.115}$$

for $t \geq t_0 + t'_d$ according to assumption 4.97. Similarly, $v(q, x_1) < 0$ for $q \in B$ and all the units in B are still saturated at 0 at $t_0 + t'_d$ which provides

$$\sum_{q \in B} v(q, x_1) y_q(t) - \sum_{q \in B} v(q, x_1) y_q(t_0 + t'_d) < \varepsilon 2^{-1/\varepsilon} \tag{4.116}$$

for $t \geq t_0 + t'_d$ by assumption 4.98. Also the defect term in equation 4.110 satisfies

$$\Delta_{x_1 Q'_1} e^{-(t-t_0)} - \Delta_{x_1 Q'_1} e^{-t'_d} < \varepsilon 2^{-1/\varepsilon} \tag{4.117}$$

for $t \geq t_0 + t'_d$ according to assumption 4.94. By summing inequalities 4.113–4.117 for differences in particular terms of equation 4.110 we obtain

$$\begin{aligned} \frac{dy_{x_1}}{dt}(t_0 + t_\ell) &= \frac{dy_{x_1}}{dt}(t_0 + t_\ell) - \frac{dy_{x_1}}{dt}(t_0 + t'_d) \\ &< -\varepsilon/3 + \varepsilon(3 + \varepsilon) 2^{-1/\varepsilon} < 0 \end{aligned} \tag{4.118}$$

from inequality 4.28. The state derivative of x_1 is further negative as inequality 4.118 is valid even for $t \geq t_0 + t_\ell$ until x_1 saturates at 0, since bounds 4.114–4.117 still apply while condition 4.113 holds for continuous $y_{x_1}(t)$ that is further decreasing according to inequality 4.118. □

The correct *timing* of the simulation still needs to be verified to ensure a sufficiently fast decrease in the defects of the continuous-time correlates of binary states, because the analysis in lemmas 3, 4 is valid only if the defect bounds 4.44, 4.51–4.53, 4.93, 4.94, and 4.96–4.98 are satisfied. According to inequality 4.43, the absolute value of the total weighted defect of saturated units affecting any unit in $\mathcal{H}$ is bounded by ε after transient time t_1, decreasing further to $\varepsilon 2^{-1/\varepsilon}$ by time $2t_1$. On the other hand, t_1 lower bounds the time necessary for activating a typical clock unit $p \in \mathcal{C}$ (see table in Figure 5) by lemma 3.1.

We will first concentrate on the *clock subnetwork* $\mathcal{C}$ excluding its connection to simulating subnetwork $\mathcal{S}$ via interface unit x_1, which will be resolved later on together with the synchronization analysis for $\mathcal{S}$. In order to validate assumption 4.44, consider e.g. a unit $o' \in \mathcal{C}$ that has also been activated according to lemma 3.1 last before unit $p \in \mathcal{C}$ from formula 4.44 starts its activation. Clearly, unit o' coincides with unit $o \in \mathcal{C}$ in table of Figure 5 except for $o = x'_k$ $(k = 2, \ldots, n+1)$ and $o = x_k$ $(k = 1, \ldots, n+1)$ whose activation is analyzed in lemma 3.2 instead, and therefore

$$o' = \begin{cases} a_k & \text{for } p = b'_k\,, \ 2 \leq k \leq n+1 \\ a_1 & \text{for } p = b_1 \\ b'_k & \text{for } p = b_k\,, \ 2 \leq k \leq n+1 \\ o & \text{otherwise}\,. \end{cases} \tag{4.119}$$

It follows from Figures 1, 2 that $v(o', r) \geq 0$. In fact only $v(z_1, c_k) = 1$ and $v(z_k, c_1) > 0$ for $k = 2, \ldots, n+1$ are positive while the remaining pairs o', r are actually not connected corresponding to $v(o', r) = 0$. In addition, $v(p, r) \geq 1$ and hence the defect in formula 4.44 can be upper bounded as follows:

$$\delta_p + \Delta_{rQ} \leq v(p, r)\delta_p + \Delta_{rQ} + v(o', r)\delta_{o'} = \Delta_{rQ_2} \tag{4.120}$$

where $Q_2 = \mathcal{H} \setminus \{o'\}$, according to definition 4.36.

Assume first that unit $r \neq x_1$ is not connected to any unit in $\mathcal{S}$, which gives

$$\Delta_{rQ_2} = \Delta_{rQ_2'} \qquad (4.121)$$

where $Q_2' = Q_2 \cap \mathcal{C}$. For $o' \neq b_k'$ $(2 \leq k \leq n+1)$ and $o' \neq b_k$ $(1 \leq k \leq n+1)$ all the units in Q_2' are saturated when o' is being activated which takes time t_1. On the other hand, the activation of $o' = b_k', b_k$ takes time $2t_1$ due to its bias $v(0, o') = -1 + \varepsilon/3$, while simultaneously units in $R \cap \mathcal{C} \subseteq Q_2'$ (for particular cases of R see table in Figure 6 in which now o' corresponds to r) saturate within a time period of length t_1 according to lemma 2.2b. Note that the corresponding unit, x_k' or x_k, preceding $o' = b_k'$ or $o' = b_k$, respectively, is already saturated at 1 by lemma 3.2 before o' unsaturates from 0. Furthermore, also all the units in Q_2' are saturated for the duration of the next t_1-period certainly before unit p from formula 4.44 unsaturates. It follows from inequality 4.43 that $\Delta_{rQ_2'} < \varepsilon$ which implies condition 4.44 according to inequality 4.120 and equation 4.121. Similarly, for $r = x_1$ (i.e. $p = a_1$) it will be shown below that all the units in Q_2 that are connected to r, especially units in $A \cup B_{t_0} \subseteq R$ from simulating subnetwork $\mathcal{S}$ are saturated before $o' = c_1$ unsaturates, which clearly suffices for proving condition 4.44.

Analogously, the stronger defect bounds 4.51–4.53 in lemma 3.2 are met for $p = x_k'$ $(2 \leq k \leq n+1)$ and $p = x_k$ $(1 \leq k \leq n+1)$ since according to lemma 3.1 the transient time $2t_1$ that is sufficient to decrease the underlying defects due to inequality 4.43, is guaranteed by the successive separate (all the units in $\mathcal{C}$ are saturated but one) activations of the preceding two units c_k, a_k with $v(c_k, r) = v(a_k, r) = 0$ before unit p unsaturates from 0. It only remains to check inequality 4.53 for $p = x_1$ when $A \cup B_{t_0} \subseteq R$, which again follows from the fact that units in $A \cup B_{t_0} \subseteq \mathcal{S}$ are saturated before $o' = c_1$ unsaturates, whose proof is given below.

Now, the synchronization of the *simulating subnetwork* $\mathcal{S}$ will be analyzed. The *first phase* of any simulated discrete step is delimited by the period during which the clock interface unit $x_1 \in \mathcal{C}$ is passive. Recall that unit x_1 is reset by x_k' for some $2 \leq k \leq n+1$ which saturates at 1 by lemma 3.2 before the subsequent unit b_k' is unsaturated from 0. The activation of b_k' then takes time $2t_1$ by lemma 3.1, from which the first period of length t_1 suffices for x_1 to saturate at 0 according to lemma 2.2b. Thus, the second t_1-period of activating the b_k' belongs already to the first simulation phase. This phase then also includes the successive separate activations of units $b_k, d_k, z_k, c_0, c_1, a_1$, each of length t_1 except for the activation of b_k that requires time $2t_1$ according to lemma 3.1. Altogether the length of the first phase is certainly at least $8t_1$. Note that in this consideration the length of activating the x_k is omitted for which only the upper bound is given in lemma 3.2. Actually, the first two t_1-periods of the first simulation phase are sufficient to stabilize the simulating subnetwork $\mathcal{S}$, i.e. to saturate all its units, e.g. even before d_k unsaturates from 0. In particular, according to lemma 4, units in A are and remain saturated at 0 after $y_{x_1} = 2/3$ while units in B_{t_0} only then saturate at 1 by lemma 2.2b within the first t_1-period, i.e. long before $o' = c_1$ unsaturates as required above. The following second t_1-period ensures that the units in the second and fifth layers of $\mathcal{S}$ saturate (see Figure 3).

Similarly, the *second phase* of any simulated discrete step is defined as the period during which the clock interface unit $x_1 \in \mathcal{C}$ is active. Recall that unit x_1 is activated by a_1 and saturates at 1 by lemma 3.2 before the subsequent unit b_1 becomes unsaturated from 0. Thus, the second simulation phase includes the successive separate activations of units $b_1, d_1, z_1, c_0, c_k, a_k$ for some $2 \leq k \leq n+1$, each of length t_1 except for the activation of b_1 that requires time $2t_1$ according to lemma 3.1. Altogether the length of the second phase is certainly at least $7t_1$. In fact, already

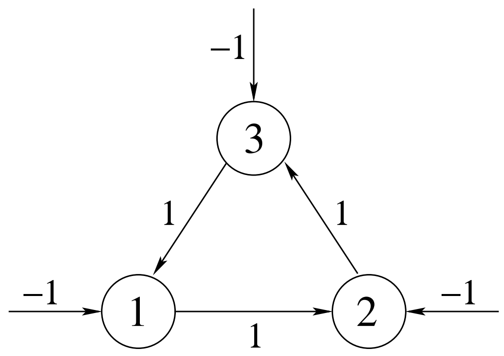


Figure 7: A 3-neuron cycle DRNN.

the first three t_1-periods of the second simulation phase are sufficient to stabilize the simulating subnetwork $\mathcal{S}$, i.e. to saturate all its units even before z_1 unsaturates from 0. In particular, according to lemma 4, units in B are and remain saturated at 0 after $y_{x_1} = 2/3$, while units in A_{t_0} only then saturate at 1 by lemma 2.2b within the first t_1-period. The following two t_1-periods ensure that the units in the second layer of $\mathcal{S}$, followed by third-layer units saturate according to lemma 2.2b (see Figure 3). Obviously, the length of the second simulation phase that is lower bounded by $7t_1$ also guarantees the defect bounds 4.93, 4.94, and 4.96–4.98 in lemma 4.2.

Finally, the lower bound $\Omega(t^*/\varepsilon)$ on the total simulation time follows immediately from the previous time analysis and equation 4.37. In addition, every unsaturated unit in $\mathcal{H}$ saturates within time at most $O(t_1)$ according to lemmas 2.2b, 3. Also during the simulation all the units in $\mathcal{H}$ can simultaneously be saturated for a period of at most t_1 before the respective defects decrease below ε according to inequality 4.43, and the next unit unsaturates. This implies the corresponding upper bound $O(t^*/\varepsilon)$ and completes the proof of the theorem. □

5 A Simulation Example

A computer program HNGEN has been created to automate the construction from theorem 1. The input for HNGEN is a text file containing the asymmetric weights and biases of a given DRNN $\mathcal{N}$, as well as its initial state. The program generates the system of differential equations 2.6 describing the dynamics of continuous-time symmetric Hopfield net $\mathcal{H}$ together with its boundary conditions in the form of a FORTRAN subroutine corresponding to $\mathcal{N}$ to be simulated. This FORTRAN procedure is then presented to a solver from the NAG library (Numerical Algorithms Group, 2002) that provides a numerical solution for this system.

By using the program HNGEN, the underlying construction has been successfully tested on several examples. Consider e.g. the simple 3-neuron cycle DRNN $\mathcal{N}$ depicted in Figure 7, initiated in a state where neuron 1 has output 1 and neurons 2 and 3 output 0. Then the computation of the network consists simply of propagating the unit signal around the cycle. Implementing this system on the HNGEN generator results in a system of 61 differential equations describing the dynamics of a continuous-time symmetric Hopfield net $\mathcal{H}$ with 61 units. Figure 8 shows the numerical evolution of the states corresponding to the clock interface unit x_1 and the three units π_1, π_2, π_3 that represent binary states of the original discrete neurons from $\mathcal{N}$, for a period of eight (2^3) simulated discrete steps confirming the correctness of the simulation. A parameter value of $\varepsilon = 0.3$ was used in this numerical

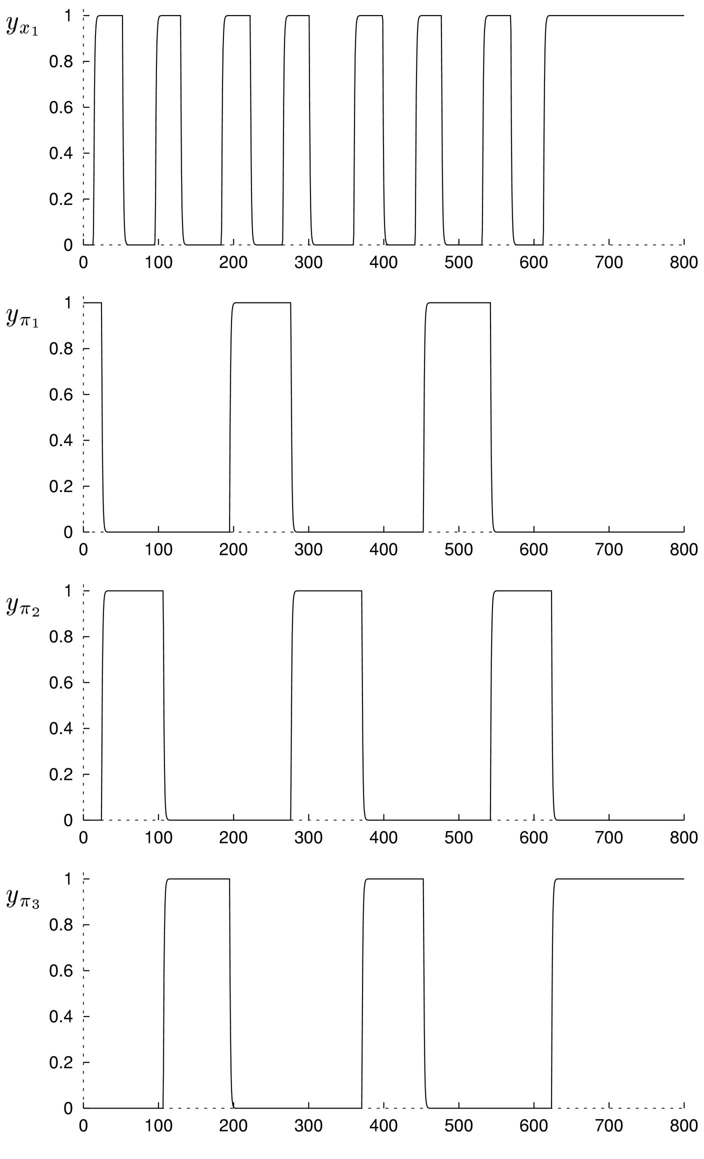


Figure 8: Continuous-time symmetric simulation of 3-neuron cycle DRNN for $\varepsilon = 0.3$.

simulation, showing that the theoretical bound on ε in theorem 1, which would require about $\varepsilon < 0.024$ in the present case, is actually quite conservative.

6 Conclusions and Open Problems

We have proved that an arbitrary convergent discrete-time recurrent neural network can be simulated by a symmetric continuous-time Hopfield net with only a linear increase in the network size. The existence of Liapunov functions for Hopfield nets precludes the use of unbounded oscillations in such a simulation; nevertheless we are able to base the construction on the bounded, but exponentially long sequence of pulses generated by the continuous-time clock subnetwork.

From the point of view of understanding analog computation in general this technique is somewhat unsatisfying, since we are still basically discretizing the continuous-time computation. It would be most interesting to develop some theoretical tools (e.g. complexity measures, reductions, universal computation) for "naturally" continuous-time computations that exclude the use of discretizing oscillations. First steps along this direction have recently been established (Ben-Hur, Siegelmann, & Fishman, 2002; Gori & Meer, 2001).

Another challenge for further research is to prove *upper bounds* on the power of continuous-time systems. Note that in the case of discrete-time analog-state neural networks a single fixed-size network with rational-number parameters can be computationally universal, i.e. able to simulate a universal Turing machine on arbitrary inputs (Siegelmann & Sontag, 1995). Can e.g. this strong universality result be generalized for continuous-time systems? Also, we have established an exponential lower bound on the convergence time of symmetric continuous-time Hopfield nets (Šíma & Orponen, 2001a): can a matching upper bound be proved, or the lower bound be increased?

References

Asarin, E., & Maler, O. (1994). On some relations between dynamical systems and transition systems. In *Proceedings of the 21st ICALP'94 International Colloquium on Automata, Languages, and Programming, LNCS 820* (pp. 59–72). Berlin: Springer-Verlag.

Balcázar, J. L., Díaz, J., & Gabarró, J. (1995). *Structural Complexity I* (2nd ed.). Berlin: Springer-Verlag.

Balcázar, J. L., Gavaldà, R., & Siegelmann, H. T. (1997). Computational power of neural networks: A characterization in terms of Kolmogorov complexity. *IEEE Transactions on Information Theory, 43*(4), 1175–1183.

Ben-Hur, A., Siegelmann, H. T., & Fishman, S. (2002). A theory of complexity for continuous time systems. *Journal of Complexity, 18*(1), 51–86.

Branicky, M. (1994). Analog computation with continuous ODEs. In *Proceedings of the PhysComp'94 Workshop on Physics and Computation* (pp. 265–274). Los Alamitos, CA: IEEE Computer Society Press.

Casey, M. (1996). The dynamics of discrete-time computation, with application to recurrent neural networks and finite state machine extraction. *Neural Computation, 8*(6), 1135–1178.

Cichocki, A., & Unbehauen, R. (1993). *Neural Networks for Optimization and Signal Processing*. Chichester: John Wiley & Sons.

Cohen, M. A., & Grossberg, S. (1983). Absolute stability of global pattern formation and parallel memory storage by competitive neural networks. *IEEE Transactions on Systems, Man, and Cybernetics, 13*(5), 815–826.

Golden, R. M. (1996). *Mathematical Methods for Neural Network Analysis and Design*. Cambridge, MA: The MIT Press.

Goles, E., & Martínez, S. (1989). Exponential transient classes of symmetric neural networks for synchronous and sequential updating. *Complex Systems, 3*(6), 589–597.

Gori, M., & Meer, K. (2001). A step towards a complexity theory for dynamical systems. Technical Report ALCOMFT-TR-01-191, Department of Computer Science, University of Aarhus, Denmark. To appear in *Mathematical Logic Quarterly*.

Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation* (2nd ed.). Upper Saddle River, NJ: Prentice-Hall.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences, 79*, 2554–2558.

Hopfield, J. J. (1984). Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences, 81*, 3088–3092.

Hopfield, J. J., & Tank, D. W. (1985). "Neural" computation of decisions in optimization problems. *Biological Cybernetics, 52*(3), 141–152.

Horne, B. G., & Hush, D. R. (1996). Bounds on the complexity of recurrent neural network implementations of finite state machines. *Neural Networks, 9*(2), 243–252.

Indyk, P. (1995). Optimal simulation of automata by neural nets. In *Proceedings of the 12th STACS'95 Annual Symposium on Theoretical Aspects of Computer Science, LNCS 900* (pp. 337–348). Berlin: Springer-Verlag.

Koiran, P., Cosnard, M., & Garzon, M. (1994). Computability with low-dimensional dynamical systems. *Theoretical Computer Science, 132*, 113–128.

Lepley, M., & Miller, G. (1983). Computational power for networks of threshold devices in an asynchronous environment. Unpublished manuscript, Department of Mathematics, Massachusetts, Institute of Technology.

Maass, W., & Orponen, P. (1998). On the effect of analog noise in discrete-time analog computations. *Neural Computation, 10*(5), 1071–1095.

Maass, W., & Sontag, E. D. (1999). Analog neural nets with Gaussian or other common noise distribution cannot recognize arbitrary regular languages. *Neural Computation, 11*(3), 771–782.

Medsker, L. R., & Jain L. C. (Eds.) (2000). *Recurrent Neural Networks: Design and Applications*. Boca Raton, FL: The CRC Press.

Moore, C. (1990). Unpredictability and undecidability in physical systems. *Physical Review Letters, 64*(20), 2354–2357.

Moore, C. (1998). Finite-dimensional analog computers: flows, maps, and recurrent neural networks. In *Proceedings of the 1st International Conference on Unconventional Models of Computation* (pp. 59–71). Berlin: Springer-Verlag.

Numerical Algorithms Group (2002). Fortran library routine D02PCF. Manual available on-line: http://www.nag.co.uk/numeric/fl/manual/pdf/D02/d02pcf.pdf.

Omohundro, S. (1984). Modelling cellular automata with partial differential equations. *Physica, 10D*, 128–134.

Orponen, P. (1996). The computational power of discrete Hopfield nets with hidden units. *Neural Computation 8*(2), 403–415.

Orponen, P. (1997a). A survey of continuous-time computation theory. In D.-Z. Du and K.-I. Ko (Eds.), *Advances in Algorithms, Languages, and Complexity* (pp. 209–224). Dordrecht: Kluwer Academic Publishers.

Orponen, P. (1997b). The computational power of continuous time neural networks. In *Proceedings of the 24th SOFSEM'97 Seminar on Current Trends in Theory and Practice of Informatics, LNCS 1338* (pp. 86–103). Berlin: Springer-Verlag.

Parberry, I. (1994). *Circuit Complexity and Neural Networks.* Cambridge, MA: The MIT Press.

Siegelmann, H. T. (1999). *Neural Networks and Analog Computation: Beyond the Turing Limit.* Boston, MA: Birkhäuser.

Siegelmann, H. T., Roitershtein, A., & Ben-Hur, A. (2000). Noisy neural networks and generalizations. In S. A. Solla, T. K. Leen, & K.-R. Müller (Eds.), *Advances in Neural Information Processing Systems (NIPS'99), 12* (pp. 335–341). The MIT Press.

Siegelmann, H. T., & Sontag, E. D. (1994). Analog computation via neural networks. *Theoretical Computer Science, 131*(2), 331–360.

Siegelmann, H. T., & Sontag, E. D. (1995). Computational power of neural networks. *Journal of Computer System Science, 50*(1), 132–150.

Šíma, J., & Orponen, P. (2000). A continuous-time Hopfield net simulation of discrete neural networks. In *Proceedings of the 2nd NC'2000 International ICSC Symposium on Neural Computing* (pp. 36–42). Wetaskiwin, Canada: ICSC Academic Press.

Šíma, J., & Orponen, P. (2001a). Exponential transients in continuous-time symmetric Hopfield nets. In *Proceedings of the 11th ICANN'2001 Conference on Artificial Neural Networks, LNCS 2130* (pp. 806-813). Berlin: Springer-Verlag.

Šíma, J., & Orponen, P. (2001b). Computing with continuous-time Liapunov systems. In *Proceedings of the 33rd STOC'2001 Annual ACM Symposium on Theory of Computing* (pp. 722–731). New York: ACM Press.

Šíma, J., Orponen, P., & Antti-Poika, T. (2000). On the computational complexity of binary and analog symmetric Hopfield nets. *Neural Computation, 12*(12), 2965–2989.

Šíma, J., & Wiedermann, J. (1998). Theory of neuromata. *Journal of the ACM, 45*(1):155–178.

Stoll, H. M., & Lee, L.-S. (1988). A continuous-time optical neural network. In *Proceedings of the IEEE International Conference on Neural Networks, volume II* (pp. 373–384).