Construction of Oscillating Chemical Register Machines on Binary Numbers using Mass-Action Kinetics

Raffael Faßler Thomas Hinze Thorsten Lenser Peter Dittrich

Friedrich-Schiller-Universität Jena, Bio Systems Analysis Group
Ernst-Abbe-Platz 1–4, D-07743 Jena, Germany
{raf,hinze,thlenser,dittrich}@minet.uni-jena.de

May 5, 2008

Abstract

Synthetic biology can particularly benefit from methods for construction of biologically inspired computing systems subsumed within the membrane computing framework. In this context, artificial chemistries form special classes of P systems. Adopting discrete concepts of computation for controlling continuous species concentrations enhances the applicability of both modelling techniques. Motivated by the idea to build chemical computers based on minimal requirements in chemistry, we introduce pure chemical register machines operating on binary numbers whose architecture is composed of reaction network motifs acting as fast switching logic gates, oscillators, and reproducible bit storage units. The dynamical machine behavior consistently employs mass-action kinetics. As a case study for simulation, a complete reaction system derived from a register machine calculating the maximum of three natural numbers illustrates the practicability of the design.

1 Introduction

For more than 20 years, decentralized data processing systems in which molecules form the storage medium and (bio)chemical reactions modify these molecules in terms of executing a computation have been studied [12]. Theoretical investigations as well as experimental implementations *in vitro*, *in vivo*, and *in silico* established separate fields of research according to supplementary assumptions in chemistry: While DNA-, RNA- or protein-based computing approaches utilize the convertibility of submolecular structures [10], cell-based computers are focused on the presence of interconnected compartments able to separate molecules and reactions as part of the computational process [11]. Additionally, genetic circuits employ inhibiting and activating effects of gene expressions [3, 6, 7].

All these fields of research have in common the usage of specific assumptions beyond pure chemistries in order to achieve Turing-completeness. Although those assumptions are essential for successful wetlab implementations, the question about minimal requirements in chemistry arises for construction of universal chemical computers from a theoretical perspective. Following an obvious intuition, the number of molecular copies within a species can represent any natural number and therefore encode unbounded memory contents. In [8], a bare-bone language based on this approach including inhibiting reactions is proposed. From

an engineering point of view, small differences between huge discrete amounts of molecules are hard and error-prone to detect. An idea to overcome this insufficiency by emulating electronic circuits for construction of chemical computational units was introduced in [9] restricted to single logic gates. Bridging the gap between single logic gates and register machines can be seen as a challenging task in terms of chemical computing since the inherent space bound has to be overcome in this case. Furthermore, lack of synchronization can affect the composition of logic gates in complex circuits. Nondiscrete data encoding in chemical reaction networks implies stochasticity which can lead to error-prone computations [13]. We introduce a modular design principle for pure chemical register machines based on multiple interconnected reaction network motifs for registers using logic gates together with extended oscillating reaction networks providing clock signals. Consequently, a binary encoding of natural numbers representing register contents is used. In order to overcome the space bound of this encoding scheme, we propose a partially self-reproducing system that extends the number of species and reactions if required. This property of structural dynamics emphasizes consideration within the P systems framework. Instructions are represented by specific molecular species whose interactions emulate the register machine program control synchronized by two offset clocks. Dynamical behavior of the register machine is modeled using mass-action kinetics assuming molecular concentrations instead of discrete particle numbers. Further related work includes chemical abstract machines as concurrent systems without kinetics [2].

The paper is organized as follows: Section 2 defines register machines operating on binary numbers, and mass-action kinetics is briefly introduced. Subsequently, a representation of boolean values as molecular species concentrations is given followed by elementary reaction motifs for flipping and sustaining boolean values in Section 3. Section 4 describes an extension of the Belousov-Zhabotinsky reaction [1, 14] in order to obtain clock signals. Furthermore, a chemical implementation of clock-controlled master-slave flip-flops and their extension to binary registers of variable length is discussed in Section 5 while Section 6 explains the program control. Finally, an example determining the maximum of three natural numbers is presented together with simulation results of the dynamical register machine behavior.

2 Prerequisites

Register Machines on Binary Numbers

A register machine on binary numbers is a tuple $M = (R, L, P, \#_0)$ consisting of the finite set of registers $R = \{\mathrm{R}_1, \ldots, \mathrm{R}_m\}$ each with binary representation of a natural number $\mathrm{R}_h \in \{0,1\}^*$, a finite set of jump labels (addresses) $L = \{\#_0, \ldots, \#_n\}$, a finite set P of instructions, and the jump label of the initial instruction $\#_0 \in L$. Available instructions are: $\#_i :$ INC R_h $\#_j$ (increment register R_h and jump to $\#_j$), $\#_i :$ DEC R_h $\#_j$ (nonnegatively decrement register R_h and jump to $\#_j$), $\#_i :$ IFZ R_h $\#_j$ $\#_p$ (if $\mathrm{R}_h = 0$ then jump to $\#_j$ else jump to $\#_p$), and $\#_i$ HALT (terminate program and output register contents). We assume a pre-initialization of input and auxiliary registers at start with input data or zero. Furthermore, a deterministic principle of operation (expressed by unique usage of instruction labels: $\forall p, q \in P \,.\, (p = \#_i : v) \wedge (q = \#_j : w) \wedge ((i \neq j) \vee (v = w))$) is supposed.

Mass-Action Kinetics for Chemical Reactions

The dynamical behavior of chemical reaction networks describes the species concentrations over the time course. According to biologically predefined motifs, a variety of models exists to formulate the reaction kinetics. Since most of them imply specific assumptions, we restrict ourselves to general mass-action kinetics [4]. Here, a continuous approach to express the dynamical behavior considers production and consumption rates v_p and v_c of each species S in order to change its concentration by $\frac{\mathrm{d}\,[S]}{\mathrm{d}\,t} = v_p - v_c$. These rates result from the reactant concentrations, their stoichiometric factors $a_{i,j} \in \mathbb{N}$ (reactants), $b_{i,j} \in \mathbb{N}$ (products) and kinetic constants $k_j \in \mathbb{R}_+$ assigned to each reaction quantifying its speed. For a reaction system with a total number of n species and r reactions

$$\begin{aligned}
a_{1,1}\mathrm{S}_1 + a_{2,1}\mathrm{S}_2 + \ldots + a_{n,1}\mathrm{S}_n &\xrightarrow{k_1} b_{1,1}\mathrm{S}_1 + b_{2,1}\mathrm{S}_2 + \ldots + b_{n,1}\mathrm{S}_n \\
a_{1,2}\mathrm{S}_1 + a_{2,2}\mathrm{S}_2 + \ldots + a_{n,2}\mathrm{S}_n &\xrightarrow{k_2} b_{1,2}\mathrm{S}_1 + b_{2,2}\mathrm{S}_2 + \ldots + b_{n,2}\mathrm{S}_n \\
&\vdots \\
a_{1,r}\mathrm{S}_1 + a_{2,r}\mathrm{S}_2 + \ldots + a_{n,r}\mathrm{S}_n &\xrightarrow{k_r} b_{1,r}\mathrm{S}_1 + b_{2,r}\mathrm{S}_2 + \ldots + b_{n,r}\mathrm{S}_n,
\end{aligned}$$

the corresponding ordinary differential equations read [5]:

$$\frac{\mathrm{d}\,[S_i]}{\mathrm{d}\,t} = \sum_{h=1}^{r} \left(k_h \cdot (b_{i,h} - a_{i,h}) \cdot \prod_{l=1}^{n} [S_l]^{a_{l,h}} \right) \quad \text{with} \quad i = 1, \ldots, n$$

All initial concentrations $[S_i](0) \in \mathbb{R}_+$, $i = 1, \ldots, n$ are allowed to be set according to the needs of the reaction system.

Chemical Encoding of Binary Values

Each boolean variable $x \in \{0, 1\}$ is represented by two correlated species X^T and X^F with complement concentrations $[X^T] \in \mathbb{R}_+$ and $[X^F] \in \mathbb{R}_+$ such that $[X^T] + [X^F] = c$ holds with $c = \textit{const.}$ The boolean value of the variable x is determined whenever one of the following conditions is fulfilled: The inequality $[X^T] \ll [X^F]$ indicates "false" ($x = 0$) and $[X^F] \ll [X^T]$ "true" ($x = 1$). In case of none of these strong inequalities holds (e.g. $[X^T] = 0.6c$ and $[X^F] = 0.4c$), the system would consider the variable x to be in both states. A method for sustaining the consistency of the system is presented in the subsequent section.

3 Storing and Flipping Single Bits

In logic circuits, many recurring modules like flip-flops can be found. Respectively, in chemical reaction networks some reactions and species forming patterns will occur as multiple motifs in different contexts. This section introduces an elementary motif for storing and flipping a single bit chemically using pairwise *switching* and *maintaining reactions*.

Let A^T and A^F be two species representing a boolean variable a. For computation purposes, its boolean value has to be set and toggled. While initial setting is done by specification of the concentrations $[A^T](0)$ and $[A^F](0)$, for toggling from "true" into "false" a switching

reaction is required that consumes A^T and produces A^F, and a second one for the opposite direction. Each of both reactions involves at least one designated reactant species acting as a unique trigger T_{A^b} with $b \in \{T, F\}$. Distinct triggers for both switching directions are mandatory, otherwise both reactions would be activated at the same time neutralizing their effects. Each trigger T_{A^b} can either effect the switching reaction as a catalyst or is converted into a product T'_{A^b}. The kinetic constant k_s for all switching reactions is uniformly set. Each switching reaction is flanked by a corresponding maintaining reaction responsible for finalizing switching processes and for compensating fluctuations to ensure that no significant rest amounts of vanishing species can remain or arise. Since maintaining reactions do not use any trigger, their kinetics and stoichiometry is adjusted to support the underlying switching process. Their kinetic constants k_m are constrained by $k_\mathrm{m} < k_\mathrm{s}$ with the only exception of maintaining reactions within an oscillator. In this case, a higher value is needed in order to transfer the concentrations of toggling species within one clock cycle. Here, we equate the kinetic constant k_mo with the switching reactions ($k_\mathrm{mo} = k_\mathrm{s}$). So, the pair of coupled switching and maintaining reactions for flipping a boolean variable a is defined through the following elementary motif with arbitrary but finite number of triggers (τ, θ) and their byproducts (ϕ, ψ):

$$\begin{aligned}
A^T + T_{A^T,1} + \ldots + T_{A^T,\tau} &\xrightarrow{k_\mathrm{s}} A^F + T'_{A^T,1} + \ldots + T'_{A^T,\theta} \\
A^F + T_{A^F,1} + \ldots + T_{A^F,\phi} &\xrightarrow{k_\mathrm{s}} A^T + T'_{A^F,1} + \ldots + T'_{A^F,\psi} \\
A^T + 2\,A^F &\xrightarrow{k_\mathrm{m}} 3\,A^F \\
A^F + 2\,A^T &\xrightarrow{k_\mathrm{m}} 3\,A^T
\end{aligned}$$

4 A Chemical Clock by Extending an Oscillating Reaction Network

A chemical counterpart of a clock is necessary in order to sequentialize the register machine instruction processing. Positive edges of clock signals can trigger micro-operations like register increment or jump to the next machine instruction. In our chemical machine model, an extended oscillating reaction network provides all clock signals. As preferred network template for permanent oscillation, we adopt the well-studied Belousov-Zhabotinsky reaction [1, 14] depicted in the upper-left part of Figure 1 whose dynamical behavior results in periodic peak-shaped signals. By using a cascade of downstream switching and maintaining reactions, we extend the primary oscillator. In this way, a normalization with respect to signal shape and concentration course can be reached. Our idea employs both converse output signals O_i^T and O_i^F (stage number indicated by index $i \in \{0, \ldots, 3\}$, see Figure 1 left) of the previous cascade stage as triggers for a subsequent chemical toggle switch. Thus, high and low concentration levels are more and more precisely separated over the time course, and the switching delay in between becomes shortened, see lower-left parts of Figure 1. After three cascade stages, the quality of the chemical clock signal turns out to be suitable for our purposes.

For technical reasons (two-phase register machine instruction processing, see Section 6), two offset clocks with designated output species C_1 and C_2 are employed. Owning the same network structure, they only differ in the time point when coming into operation caused by individual initializations (species producing clock signals C_1: $[O^F_{0,C_1}](0) = 2, [O^T_{0,C_1}](0) = 1$;

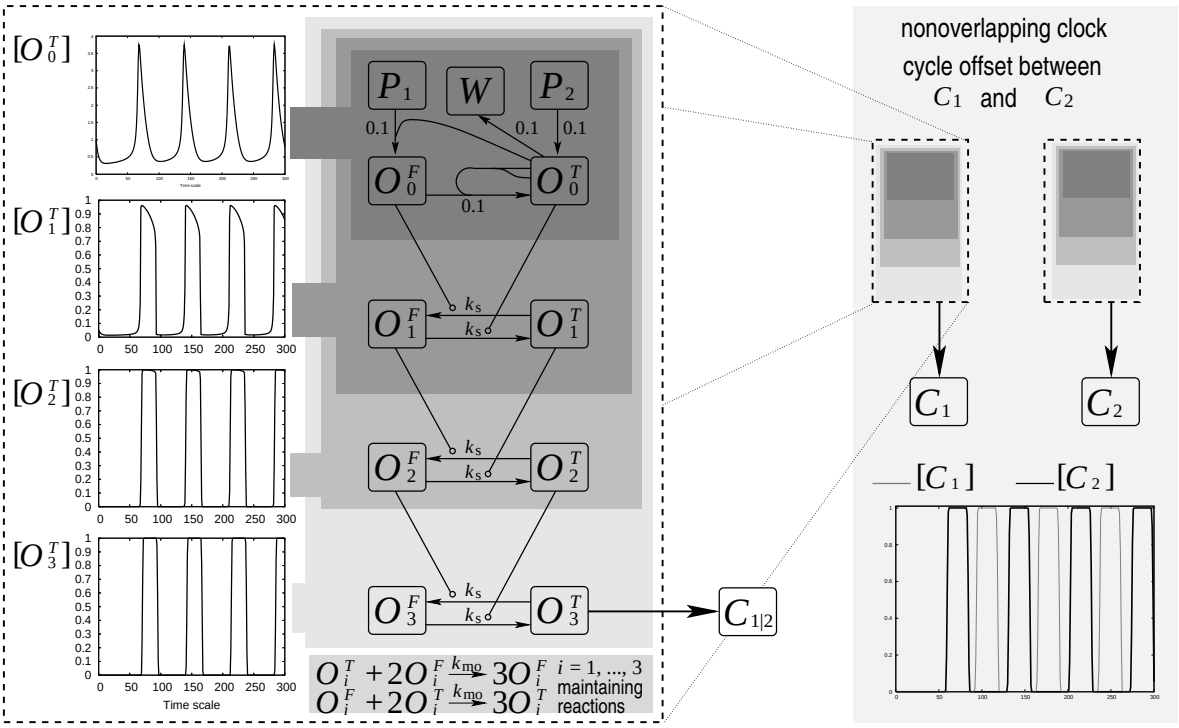

Figure 1: Generation of chemical clock signals by cascadization of toggle switches

corresponding species for clock signals C_2: $[O^F_{0,C_2}](0) = 0, [O^T_{0,C_2}](0) = 0$; species with identical initial concentrations: $[P_{1,C}](0) = 3, [P_{2,C}](0) = 1, [W_C](0) = 0, [O^F_{i,C}](0) = 1, [O^T_{i,C}](0) = 0), i \in \{1,2,3\}, C \in \{C_1, C_2\}$). C_1 and C_2 provide nonoverlapping clock signals whose offset constitutes approximately one half of the clock cycle, see Figure 1 right.

5 Chemical Master-Slave Flip-Flops and Binary Registers

Master-slave flip-flops (MSFFs) are clock-controlled bit storage units of high reliability in binary function. We introduce a reaction network that mimics a MSFF based on the aforementioned chemical clocks and bit manipulating reaction motifs. Moreover, a chain of MSFFs forms a register R_h ($h \in \{1, \ldots, |R|\}$) with bitwise extendable initial length of one bit. In operation, it processes binary numbers $\ldots b_{l_h} b_{l_{h-1}} \ldots b_2 b_1$ with $b_\alpha \in \{0, 1\}$. Furthermore, each register is equipped with predefined triggers in order to carry out micro-operations "increment", "nonnegative decrement", and "comparison to zero", each processed within one clock cycle.

Within a MSFF, bit setting is coupled to specific edges of the clock signal in order to prevent premature switches. In our MSFF implementation, bit setting is organized in two phases (master and slave part). Within the master part, a bit can be preset using specific master species M^T and M^F co-triggered by positive edges of the clock signal C_1, while the subsequent slave part finalizes the setting by forwarding the preset bit from the master species to the correlated slave species S^T and S^F triggered by positive edges of the offset clock signal C_2. A subnetwork consisting of eight switching reactions (see graymost highlighted boxes

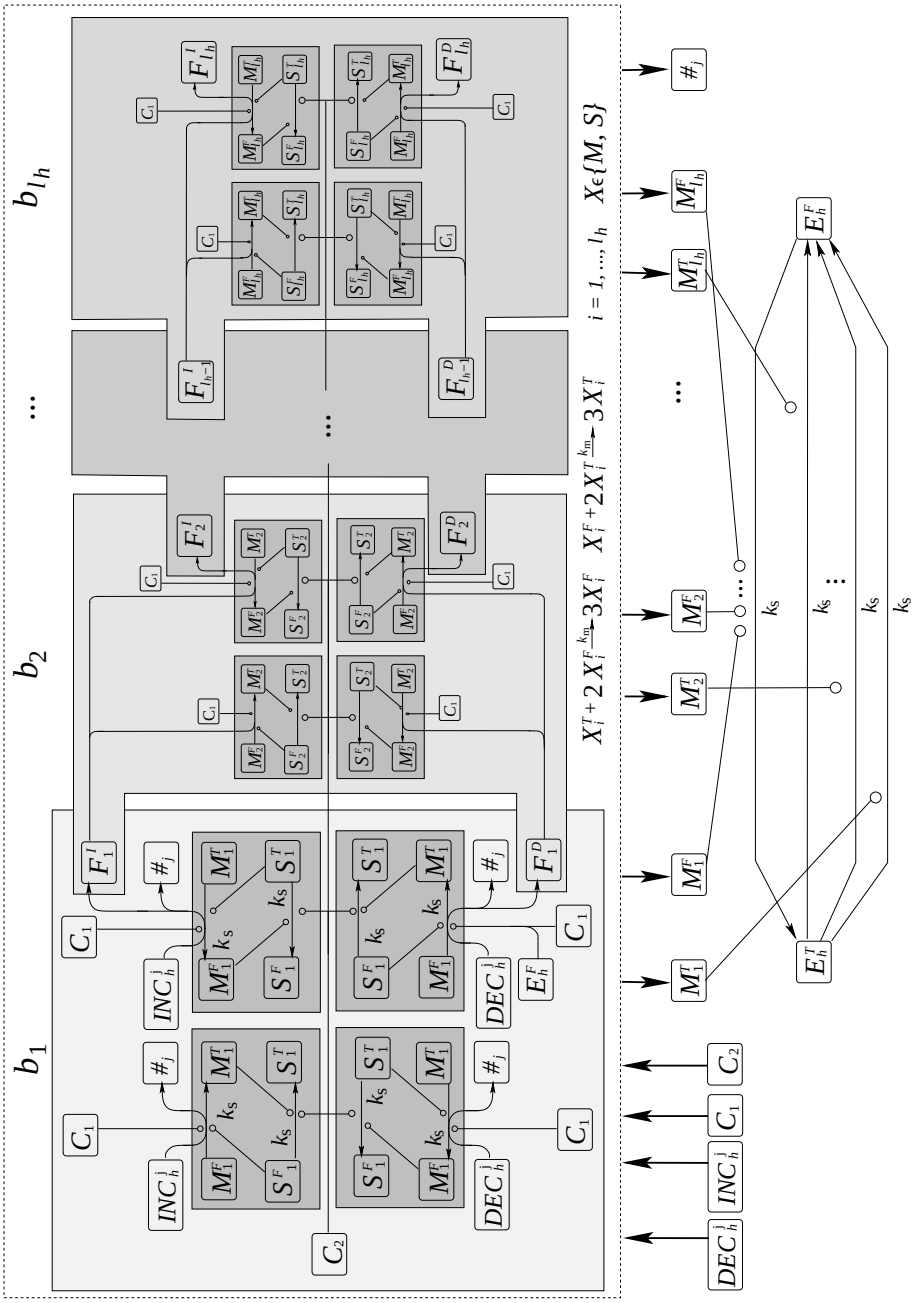

Figure 2: Chemical reaction network of a register capable of processing a bitwise extendable binary number $\dots b_{l_h} b_{l_{h-1}} \dots b_2 b_1$ with $b_\alpha \in \{0,1\}$ including interfaces for micro-operations increment, nonnegative decrement, and comparison to zero. Species for binary data storage: M^F_γ, M^T_γ: contents of flip-flop master parts; S^F_γ, S^T_γ: contents of flip-flop slave parts, $\gamma = 1, \dots, l_h$; E^F_h, E^T_h: indicators for register equality to zero

within each MSFF in Figure 2) covers this task.

With regard to the functionality of a register machine, a sequence of interconnected MSFFs represents a register. Interconnections between neighboured MSFFs reflect the capability of incrementing and decrementing register contents. In case of incrementation, designated trigger molecules INC_h^j effect a successive bit flipping: Starting from the least significant bit b_1, "1" is consecutively converted into "0" until "0" appears first time which is finally converted into "1". Intermediate carry species F_α^I act as forwarding triggers between consecutive bits, see Figure 2. If the most significant bit b_{l_h} is reached increasing the concentration of carry species $F_{l_h}^I$, six new species $M_{l_h+1}^T$, $M_{l_h+1}^F$, $S_{l_h+1}^T$, $S_{l_h+1}^F$, $F_{l_h+1}^D$, and $F_{l_h+1}^I$ are appended to the reaction system together with the corresponding set of reactions forming the subnetwork for managing bit b_{l_h+1} including update of $M_{l_h+1}^F$ and $M_{l_h+1}^T$ within reactions performing comparison to zero, see Figure 2.

Decrementation is organized in a similar way using initial triggers DEC_h^j and intermediate molecules of carry species F_β^D. In order to achieve nonnegative processing, a species E_h^F indicating equality to zero, set by a satellite network, prevents decrementation of binary strings $0 \ldots 0$. Figure 2 shows the reaction network structure of a register whose species F_α^I, F_β^D, M_γ^F, M_γ^T, S_γ^F, and S_γ^T are specific with respect to both register identifier h and bit position l_h within the register. Any comparison to zero is done by a satellite network which uses presence of any species M_κ^T with $\kappa = 1, \ldots, l_h$ as triggers in order to flip an equality indicator bit e (species E_h^T and E_h^F) onto "0", while all species M_κ^F with $\kappa = 1, \ldots, l_h$ are needed for flipping onto "1", respectively. The indicator e can be used for program control, see next section. As a further byproduct of each micro-operation on a register, molecules of the form $\#_j \in L$ encoding the jump label of the subsequent machine instruction are released.

6 Implementing a Chemical Program Control

A sequence of reactions directly derived from the given program P of the underlying register machine $M = (R, L, P, \#_0)$ carries out the program control as follows: For each jump label $\#_j \in L$ we introduce a dedicated *label species* $\#_j$ with initial concentrations $[\#_0](0) = 1$ and $[\#_\kappa](0) = 0$ for $\kappa \in \{1, \ldots, |L| - 1\}$. Accordingly, a set of *instruction species* $I_\nu \in \{INC_h^j, DEC_h^j \mid \forall h \in \{1, \ldots, |R|\} \wedge \forall j \in \{0, \ldots, |L|-1\}\} \cup \{IFZ_h^{j,q} \mid \forall h \in \{1, \ldots, |R|\} \wedge \forall j, q \in \{0, \ldots, |L|-1\}\} \cup \{HALT\}$ is created with initial concentration $[I_\nu](0) = 0$. Furthermore, for each instruction in P a network motif consisting of a *program-control reaction* with kinetic constant $k_\mathrm{p} < k_\mathrm{s}$ and a consecutive *bypass reaction* with $k_\mathrm{b} \leq k_\mathrm{s}$ is defined. Following the two-phase structure of a register machine instruction, these reactions first consume its incipient label species, then produce the corresponding instruction species as an intermediate product and finally convert it into the label species of the subsequent instruction if available. In order to strictly sequentialize the execution of instructions according to the program P, clock species C_1 and C_2 with offset concentration course provided by both oscillators trigger program-control and bypass reactions alternating as catalysts. The set of reactions for each

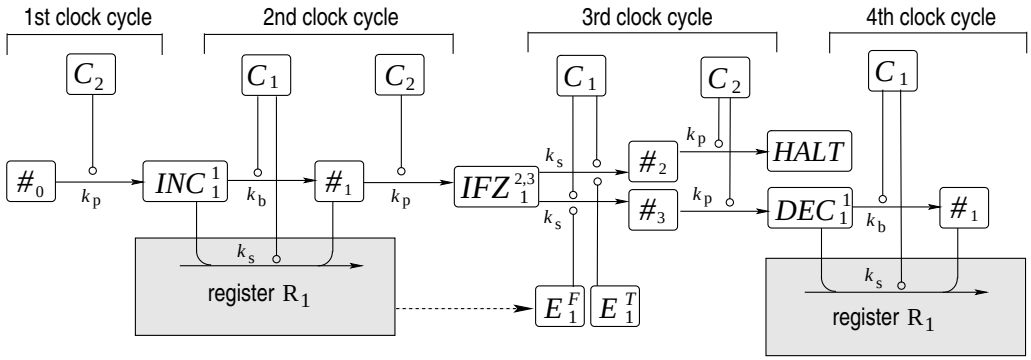

Figure 3: Example of the chemical program control for register machine $M = (\{\mathrm{R}_1\}, \{\#_0, \ldots, \#_3\}, P, \#_0)$ with $P = \{\#_0 : \text{INC R}_1\ \#_1,\ \ \#_1 : \text{IFZ R}_1\ \#_2\ \#_3,\ \ \#_2 : \text{HALT},\ \ \#_3 : \text{DEC R}_1\ \#_1\}$

type of register machine instructions is defined as follows:

instruction	reactions
$\#_i : \text{INC R}_h\ \#_j$	$\#_i + C_2 \xrightarrow{k_\mathrm{p}} INC_h^j + C_2$
	$INC_h^j + C_1 \xrightarrow{k_\mathrm{b}} \#_j + C_1$
$\#_i : \text{DEC R}_h\ \#_j$	$\#_i + C_2 \xrightarrow{k_\mathrm{p}} DEC_h^j + C_2$
	$DEC_h^j + C_1 \xrightarrow{k_\mathrm{b}} \#_j + C_1$
$\#_i : \text{IFZ R}_h\ \#_j\ \#_q$	$\#_i + C_2 \xrightarrow{k_\mathrm{p}} IFZ_h^{j,q} + C_2$
	$IFZ_h^{j,q} + E_h^T + C_1 \xrightarrow{k_\mathrm{s}} \#_j + E_h^T + C_1$
	$IFZ_h^{j,q} + E_h^F + C_1 \xrightarrow{k_\mathrm{s}} \#_q + E_h^F + C_1$
$\#_i : \text{HALT}$	$\#_i + C_2 \xrightarrow{k_\mathrm{p}} HALT + C_2$

Instruction species of the form INC_h^j act as triggers for incrementing the contents of register R_h done within its reaction network part, see Figure 2. Here, INC_h^j is converted into the byproduct $\#_j$ that provides the label species of the subsequent instruction. Accordingly, species DEC_h^j initiate a concerted set of reactions decrementing register R_h nonnegatively. Instruction species of the form $IFZ_h^{j,q}$ utilize a reaction network module attached to register R_h that releases two species E_h^T and E_h^F whose concentrations indicate whether or not $\mathrm{R}_h = 0$. Instruction species of the form INC_h^j, DEC_h^j, and $IFZ_h^{j,q}$ react into the corresponding label species $\#_j$ and $\#_q$. Since there is no reaction with instruction species $HALT$ as reactant, the program stops in this case. Figure 3 illustrates an example of a chemical program control which also gives an overview about the interplay of all predefined modules.

Although instruction species are consumed within register modules, this process could be too slow in a way that a significant concentration of an instruction species outlasts the clock cycle. This unwanted effect is eliminated by bypass reactions running in parallel to the designated register operation.

7 Example: Maximum of Three Natural Numbers

$\mathrm{R}_5 := \max(a, b, c)$ is determined by $M = (\{\mathrm{R}_1, \ldots, \mathrm{R}_5\}\{\#_0, \ldots, \#_{24}\}, P, \#_0)$, initialized with $\mathrm{R}_1 := a$, $\mathrm{R}_2 := b$, and $\mathrm{R}_1 := c$ using following instructions in P:

$\#_0$:	IFZ	R_1	$\#_5$ $\#_1$	
$\#_1$:	IFZ	R_2	$\#_5$ $\#_2$	
$\#_2$:	DEC	R_1	$\#_3$	
$\#_3$:	INC	R_4	$\#_4$	
$\#_4$:	DEC	R_2	$\#_0$	$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\mathrm{R}_4 := \lvert a - b\rvert$
$\#_5$:	IFZ	R_1	$\#_6$ $\#_9$	
$\#_6$:	IFZ	R_2	$\#_{12}$ $\#_7$	$\ldots b \geq a \rightsquigarrow$ move R_2 to R_4, obtain $\mathrm{R}_4 := b = \max(a, b)$
$\#_7$:	DEC	R_2	$\#_8$	
$\#_8$:	INC	R_4	$\#_6$	
$\#_9$:	IFZ	R_1	$\#_{12}$ $\#_{10}$	$\ldots a > b \rightsquigarrow$ move R_1 to R_4, obtain $\mathrm{R}_4 := a = \max(a, b)$
$\#_{10}$:	INC	R_4	$\#_{11}$	
$\#_{11}$:	DEC	R_1	$\#_9$	
$\#_{12}$:	IFZ	R_4	$\#_{17}$ $\#_{13}$	
$\#_{13}$:	IFZ	R_3	$\#_{17}$ $\#_{14}$	
$\#_{14}$:	INC	R_5	$\#_{15}$	
$\#_{15}$:	DEC	R_4	$\#_{16}$	
$\#_{16}$:	DEC	R_3	$\#_{12}$	$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\mathrm{R}_5 := \lvert \mathrm{R}_4 - c\rvert$
$\#_{17}$:	IFZ	R_4	$\#_{18}$ $\#_{21}$	
$\#_{18}$:	IFZ	R_3	$\#_{24}$ $\#_{19}$	$. c \geq \mathrm{R}_4 \rightsquigarrow$ mv. R_3 to R_5, obtain $\mathrm{R}_5 := c = \max(\mathrm{R}_4, c)$
$\#_{19}$:	INC	R_5	$\#_{20}$	
$\#_{20}$:	DEC	R_3	$\#_{18}$	
$\#_{21}$:	IFZ	R_4	$\#_{24}$ $\#_{22}$	$.\mathrm{R}_4 > c \rightsquigarrow$ mv. R_4 to R_5, obtain $\mathrm{R}_5 := \mathrm{R}_4 = \max(\mathrm{R}_4, c)$
$\#_{22}$:	DEC	R_4	$\#_{23}$	
$\#_{23}$:	INC	R_5	$\#_{21}$	
$\#_{24}$:	HALT			

For solving the problem instance $\max(2, 1, 3)$, M results in a chemical reaction network on two bit registers ($l_h = 2, h = 1, \ldots, 5$) composed of 142 species and 223 reactions whose dynamical behavior using kinetic constants $k_\mathrm{s} = 4$, $k_\mathrm{m} = 1$, $k_\mathrm{mo} = 3$, $k_\mathrm{b} = 0.5$, $k_\mathrm{p} = 1$ and initial quantities $[M^T_{1,1}](0) = 0$, $[M^T_{1,2}](0) = 1$, $[M^T_{2,1}](0) = 1$, $[M^T_{2,2}](0) = 0$, $[M^T_{3,1}](0) = 1$, $[M^T_{3,2}](0) = 1$, $[M^T_{4,1}](0) = 0$, $[M^T_{4,2}](0) = 0$, $[M^T_{5,1}](0) = 0$, $[M^T_{5,2}](0) = 0$ is shown in Figure 4.

8 Discussion

Biologically inspired molecular computing approaches mostly benefit from a massive data parallel principle of operation. One might argue that nondeterministic models for computation could be the best choice to formalize computational aspects of molecular interactions. Indeed, those descriptions are known. Molecular substructures or spatial data, both preferrably represented as strings, provide the contextual information in these cases. Here, we restrict ourselves to symbol objects encoding molecular concentrations. Contexts can be found in the structure of the reaction network that leads to a desired synchronized dynamical behavior. Consequently, we sequentialized reactions by introduction of a clock and modularized design principles. This way, we obtain a reliable chemical counterpart of a register machine on binary numbers. Although the emulation of this deterministic machine is quite slow, technical details of performing a sequence of computational steps can be studied precisely according to mass-action kinetics. So, concentration gradients reach maxima or minima over the time

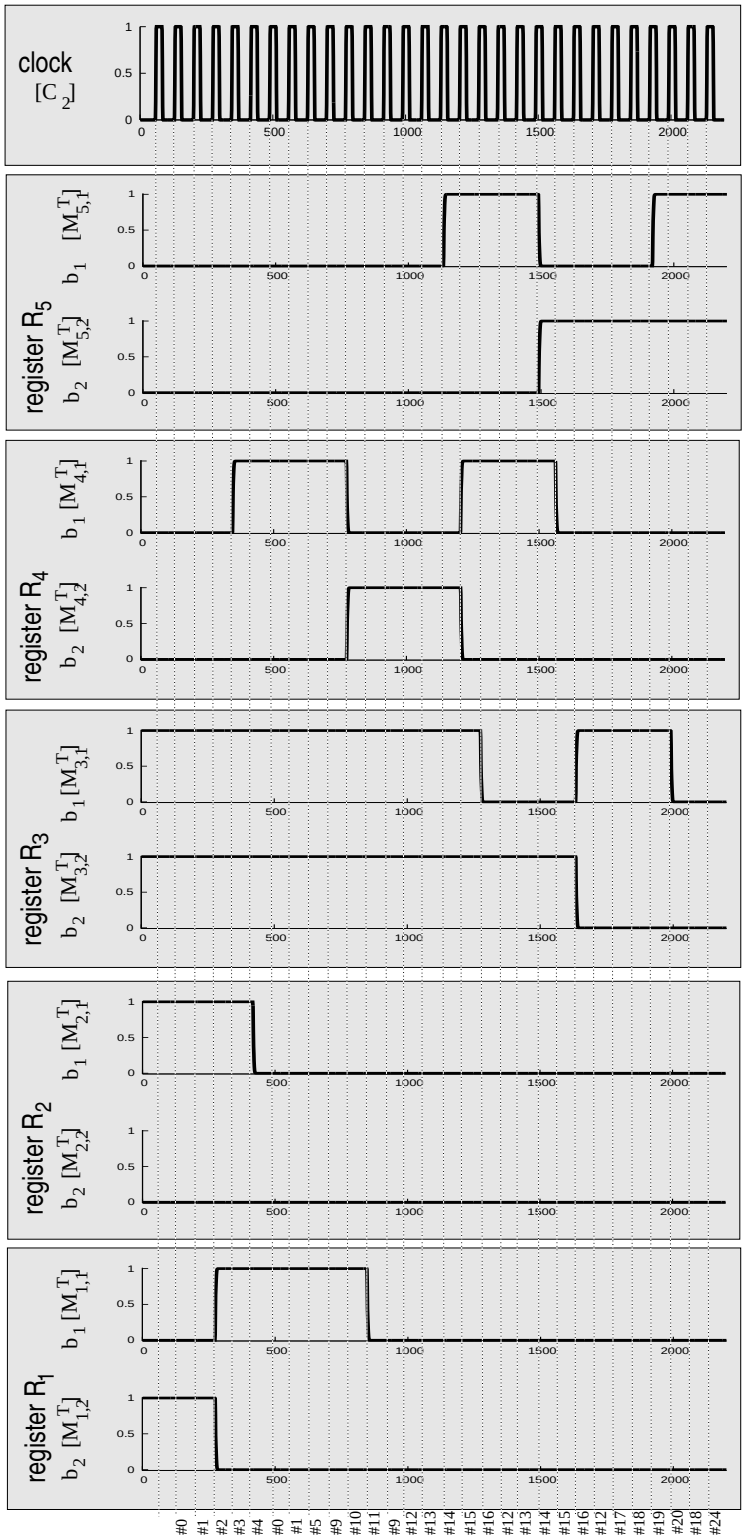

Figure 4: Dynamical behavior of a chemical register machine determining $\max(2, 1, 3)$

course asymptotically. For that reason, there are correlations between kinetic parameters to guarantee the finalization of a micro-operation within a clock cycle. The ratio of kinetic parameters (kinetic constants as well as stoichiometry) employed in all switching, maintaining, bypass, and program-control reactions of the register machine is balanced to achieve reliability in operation.

All simulations of the dynamical register machine behavior were carried out using CellDesigner version 3.5.2, an open source software package for academic use (`www.celldesigner.org`). The register machine implementation in SBML (Systems Biology Markup Language) is available from the authors upon request.

The execution time of chemical register machine programs can be reduced by parallelization using a shared memory model whose principle of operation follows the CREW strategy (concurrent read exclusive write). To this end, the sequential program P is decomposed into a number of nonoverlapping threads $T_\tau \subseteq P$ with $\bigcup T_\tau = P$ whereas each thread subsumes instructions allowed to run concurrently within one clock cycle. On the one hand, serially direct consecutive writing instructions (INC, DEC) on pairwise different registers might form a common thread for writing. On the other hand, any set of serially direct consecutive reading instructions (IFZ) represents a thread candidate for reading. Eventually, each halting instruction (HALT) becomes a separate thread. The chemical program control then acts thread-based: While reactions performing all instructions within one thread run simultaneously, a series of threads is controlled by handing over dedicated return label species which initiate the subsequent thread. This way, no further explicit synchronization mechanisms are required.

9 Conclusions

This paper presents a design principle for pure chemical register machines operating on binary numbers. In contrast to discrete term rewriting techniques to evolve chemical reactions, we consistently employ continuous molecular concentrations due to mass-action reaction kinetics. Dynamically extendable sets of species and reaction rules enable Turing completeness inspired by assumption of artificial self-reproduction. A strict modularization of reaction subnetwork motifs for storing single bits in terms of master-slave flip-flops can be seen as prerequisite to encompass reproducible parts of the system. Although the sequentialization of data processing causes a slow-down of computations in comparison to nondeterministic models, the effort for synchronization within the reaction network remains low and micro-operations within computational steps can be studied in detail more easily. Further work will deal with parallelization of instructions independently operating on distinct registers.

Acknowledgments

This work is part of the ESIGNET project (Evolving Cell Signaling Networks *in silico*), which has received research funding from the European Community's Sixth Framework Programme (project no. 12789). Further funding from the German Research Foundation (DFG, grant DI852/4-2) is gratefully acknowledged.

References

[1] B.P. Belousov. A Periodic Reaction and Its Mechanism. *Compilation of Abstracts in Radiation Medicine* **147**:145, 1959

[2] G. Berry, G. Boudol. The Chemical Abstract Machine. *Theoretical Computer Science* **96**:217-248, 1992

[3] N. Busi et al. Computing with Genetic Gates, Proteins, and Membranes. In H.J. Hoogeboom et al. (Eds.), *Membrane Computing. LNCS* **4361**:233-249, 2006

[4] K.A. Connors. *Chemical Kinetics.* VCH Publishers, 1990

[5] P. Dittrich, J. Ziegler, W. Banzhaf. Artificial Chemistries: A Review. *Artificial Life* **7(3)**:225-275, 2001

[6] T.S. Gardner, C.R. Cantor, J.J. Collins. Construction of a Genetic Toggle Switch in *Escherichia coli. Nature* **403**:339-342, 2000

[7] T. Hinze, S. Hayat, T. Lenser, N. Matsumaru, P. Dittrich. Hill Kinetics Meets P Systems. In G. Eleftherakis et al. (Eds.) *Membrane Computing. LNCS* **4860**:320-335, 2007

[8] A. Liekens, C. Fernando. Turing Complete Catalytic Particle Computers. In F.A. Costa et al. (Eds.), *Advances in Artificial Life. LNCS* **4648**:1202-1211, 2007

[9] M.O. Magnasco. Chemical Kinetics is Turing Universal. *Physical Review Letters* **78(6)**:1190-1193, 1997

[10] G. Păun, G. Rozenberg, A. Salomaa. *DNA Computing: New Computing Paradigms.* Springer Verlag, 1998

[11] G. Păun. *Membrane Computing: An Introduction.* Springer Verlag, 2002

[12] T. Sienko, A. Adamatzky, N. Rambidi, M. Conrad (Eds.), *Molecular Computing.* MIT Press, 2003

[13] D. Soloveichik, M. Cook, E. Winfree, J. Bruck. Computation with Finite Stochastic Chemical Reaction Networks. *Nat. Comp.*, DOI 10.1007/s11047-008-9067-y, 2008

[14] A.M. Zhabotinsky. Periodic Processes of Malonic Acid Oxidation in a Liquid Phase. *Biofizika* **9**:306-311, 1964