nature biotechnology

In silico simulation of biological network dynamics

Lukasz Salwinski & David Eisenberg

Realistic simulation of biological networks requires stochastic simulation approaches because of the small numbers of molecules per cell. The high computational cost of stochastic simulation on conventional microprocessor-based computers arises from the intrinsic disparity between the sequential steps executed by a microprocessor program and the highly parallel nature of information flow within biochemical networks. This disparity is reduced with the Field Programmable Gate Array (FPGA)-based approach presented here. The parallel architecture of FPGAs, which can simulate the basic reaction steps of biological networks, attains simulation rates at least an order of magnitude greater than currently available microprocessors.

Biological systems consist of thousands of simple elements linked together to form complex networks capable of adapting to a diversity of stimuli[1]. The elements of those networks are molecules and molecular complexes within a cell. They perform simple tasks, such as amplification or integration of a signal[2,3], and the complexity of the cellular response emerges from the structure and dynamics of the entire network[2–4]. To understand and predict the behavior of such networks under a realistic variety of external stimuli, efficient algorithms for simulation are required because the kinetics of all but trivial biochemical networks cannot be described analytically. Moreover, because of the low abundance of some molecules within cells, Monte Carlo simulation methods must be used to capture the stochastic behavior of the system.

Exact Monte Carlo methods, pioneered nearly three decades ago by Gillespie[5] and subsequently improved[6–9], have been applied successfully to simulations of small biological networks[10,11]. But simulation of larger networks approaching the size of those describing the behavior of entire cells has not yet been possible because of both limited experimental data and the high computational demands of the conventional stochastic algorithms[12], which scale at best as $O(n \log n)$ with the network size n. With the rise of various 'omics' approaches, the limitation of experimental data is being lifted, but the computational demands remain staggering for simulating networks of thousands of reactions involving thousands of reactants. The problem stems from the intrinsic disparity between the sequential nature of microprocessor architecture and the highly parallel nature of biological systems, with the result that simulation times become prohibitively long.

Here we demonstrate that a stochastic simulation algorithm can be efficiently implemented by using reprogrammable FPGA devices to build a microelectronic circuit that simulates the kinetics of biochemical networks. Such devices, built as an array of simple configurable logic blocks embedded in a programmable interconnection matrix, are ideally suited to implement highly parallel architectures comparable in complexity to biochemical networks. Circuits based on FPGAs scale efficiently so that simulations of realistic biological systems should be possible.

The basic building blocks of the design are shown in **Figure 1a**. The circuit simulates the elementary bimolecular reaction

$$A + B \rightarrow S$$

The circuit is composed of N_A, N_B and N_S counters that store the number of molecules of each reacting species as well as of a set of linear feedback shift register–based, pseudo-random number generators (RND_A, RND_B, RND_k) and comparators. Every clock cycle, the generators are used to draw a sample from a random variable distribution that reflects the probability of the reaction progressing by one molecular step in a specified, discrete time interval. Networks of coupled reactions can be simulated by combining those building blocks into larger circuits with up to ~500 reactions fitting into a single, commercially available, integrated circuit. Larger systems can be simulated by partitioning the design between several FPGAs incorporated into a single device.

When two or more reactions involving a particular reactive species are to progress concurrently within the same time step, additional control circuitry is used to resolve conflicts that arise. As currently implemented, all requested reactions may progress only when the net change of the number of molecules of interest is not greater than one; otherwise the progression of all the reactions changing the abundance of this molecule is cancelled.

The size of the discrete time step is a key parameter that determines the speed of the simulation. It is limited by both the maximum expected probability of the cancellations mentioned above and by the rate of the fastest reaction within the simulated system. The expected probability of cancellations, P_{cancel}, is equal to a product:

$$P_{cancel} = P_{R1} \times P_{R2}$$

where P_{R1} and P_{R2} are the probabilities that two competing reactions $R1$ and $R2$ advance within the same discrete time interval. These probabilities can be controlled by adjusting the size of the time step.

The rate of the most rapid reaction within the simulated system requires the time step to be adjusted so that the probability of more than a single pair of molecules undergoing this reaction within a given time interval is small when compared to the probability P_R that exactly one

UCLA-DOE Institute for Genomics and Proteomics, Howard Hughes Medical Institute, Molecular Biology Institute, Box 951570, Los Angeles, California 90095-1570, USA. Correspondence should be addressed to D.E. (david@mbi.ucla.edu).

Published online 4 July 2004; doi:10.1038/nbt991

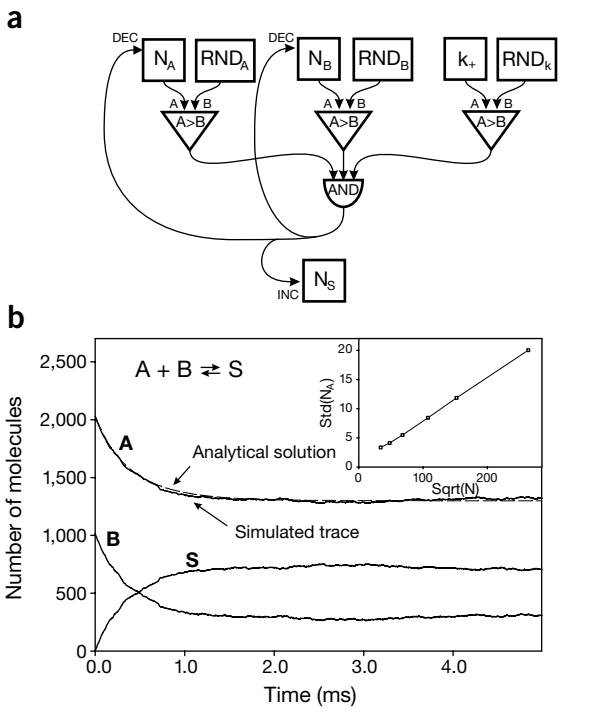

Figure 1 FPGA simulation of chemical reactions. (**a**) FPGA simulation of an elementary bimolecular reaction. The number of molecules of each reactive species is stored as a number within 16-bit bidirectional counters (N_a, N_b, N_c) whereas constants, such as those defining the stochastic reaction rates are stored within 16-bit registers (N_k). After loading these with the initial values, the state of all the counters is updated, in two phases, every cycle of the system clock.

Every rising edge of the clock triggers generation of a set of 16-bit random numbers by linear feedback shift register–based pseudo-random number generators[14]. The generated values are then compared with the current state of the relevant counters and registers. Because the random numbers generated by distinct generators are statistically independent, the probability that each of them is smaller than the corresponding counter (register) value is:

$$P_t = P(RND_A < N_A,\ RND_B < N_B,\ RND_k < N_k) = a \times N_A \times N_B$$

where:

$$a = \frac{N_k}{(2^{16}-1)^3}$$

With a proper choice of N_k, P_t has the same distribution as the probability of the reaction:

$$A + B \rightarrow C$$

progressing by one step within a predefined short time interval.

A random variable T with P_t distribution is readily calculated by performing a logical AND operation on the comparators' outputs. It can be used to update the contents of the counters when triggered by the falling edge of the system clock. In the case of the reaction modeled here, N_A and N_B counters would be incremented and N_C decremented by one every time T = true, to reflect the occurrence of a single reactive event. (**b**) Stochastic simulation of a simple chemical equilibrium. The figure shows traces generated by a circuit designed to simulate a simple equilibrium described by an equation $A + B = S$. The simulated traces agree well with the analytical solution of the deterministic differential equations (dashed line) whereas the magnitude of the observed fluctuations scales with the square root of the system size, as expected (inset).

pair reacts. The upper limit on the probability that the reaction progresses by more than one step within a given time interval can be estimated as:

$$P_{multi} \approx \frac{P_R}{(1-P_R)^2} - P_R \approx P_R^2$$

and again can be controlled by the appropriate choice of the time step size. In practice, setting the time interval so that the $\max(P_R) < 0.01$ limits both P_{cancel} and P_{multi} to <1% of P_R for the most rapid reaction and to correspondingly less for all other reactions. Notice that both errors arise as an intrinsic consequence of approximating a Poisson distribution with a linear one. Further reduction of the errors represented by these quantities is thus possible by a more accurate implementation of the Poisson distribution, taking into account its nonlinearity.

Once the FPGA is programmed and the counters loaded with the initial values, the state of the entire system is updated every cycle of the system clock, running at frequencies on the order of 100 MHz. Although slower than the clocks running currently available microprocessors, the massively parallel architecture of the FPGA-based system can result in simulation rates at least an order of magnitude higher than conventional microprocessor-based systems implementing the Gillespie algorithm.

Figure 1b shows the time course of a simple equilibrium reaction described by an equation

$$A + B = S$$

that was simulated using building blocks shown in **Figure 1a**. The simulated traces (solid lines) agree well with the analytical solution of the deterministic differential equation (dashed lines). The solid traces reveal the magnitude of the stochastic fluctuations. These scale, as expected, as the square root of the system size (inset).

Whereas time evolution of the above reaction can be analyzed analytically, the behavior of systems even slightly more complex, such as enzymatic reactions described by Michaelis-Menten equations, have to be simulated numerically even when the size of the system justifies deterministic description of the system. **Figure 2** shows typical time courses generated by a FPGA circuit simulating simple Michaelis-Menten kinetics for a system of 200 substrate molecules and another of 20,000 substrate molecules. Notice that the simulated traces generated for the 20,000 molecule system closely follow the deterministic solution of the corresponding set of coupled kinetic equations whereas the 200 molecule system reveals pronounced stochastic fluctuations around the deterministic values (dashed curves).

Finally, to test the scalability of the FPGA approach we built a circuit that simulates prokaryotic gene expression according to the model of Kierzek[13]. It is composed of 11 coupled reactions involving 12 distinct reactive species. Notice that despite a significant increase of the system complexity, its entire state has been updated every clock cycle, as in the preceding examples. The traces generated during FPGA-implemented simulation (**Fig. 3**) show good qualitative agreement with the results of a Gillespie-based simulation. Most notably, at low induction strength, pronounced stochastic fluctuations of the protein synthesis rate are easily observed.

The above examples constitute a proof-of-principle demonstration that reprogrammable FPGA technology can be applied to efficiently simulate stochastic behavior of biological systems. Such simulations will ultimately be needed to understand and predict the dynamic behavior of systems as complex as entire cells. Additionally, because of the broad spectrum of problems such as the kinetics of biochemical reactions that can be described using the same mathematical formalism—systems of ordinary differential equations—the approach presented here can find application in other areas of research.

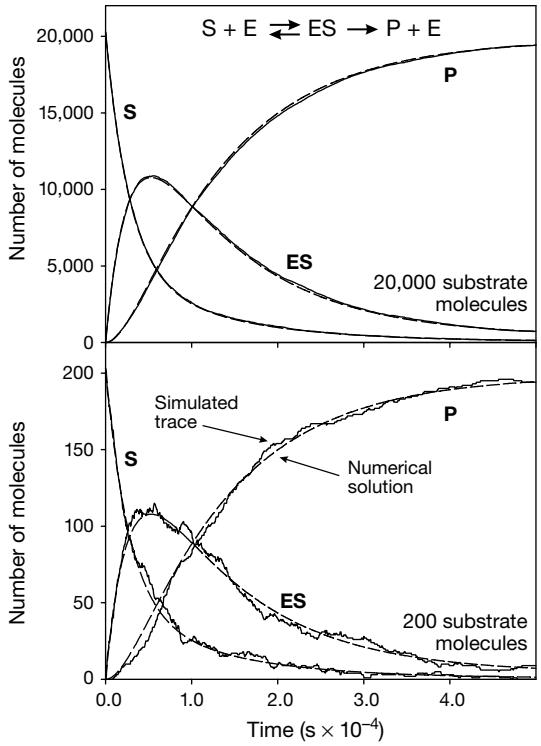

Figure 2 Stochastic simulation of the Michaelis-Menten kinetics. The figure shows traces generated by a circuit designed to simulate the well-known Michaelis-Menten kinetics of enzymatic reaction (S, substrate; P, product; E, free enzyme; ES, enzyme-substrate complex). For systems of both large and small size, the simulated traces (solid lines) agree well with the numerical solution (dashed lines) of the deterministic differential equations describing Michealis-Menten kinetics, whereas the magnitude of the observed fluctuations scales with the square root of the system size, as expected (not shown).

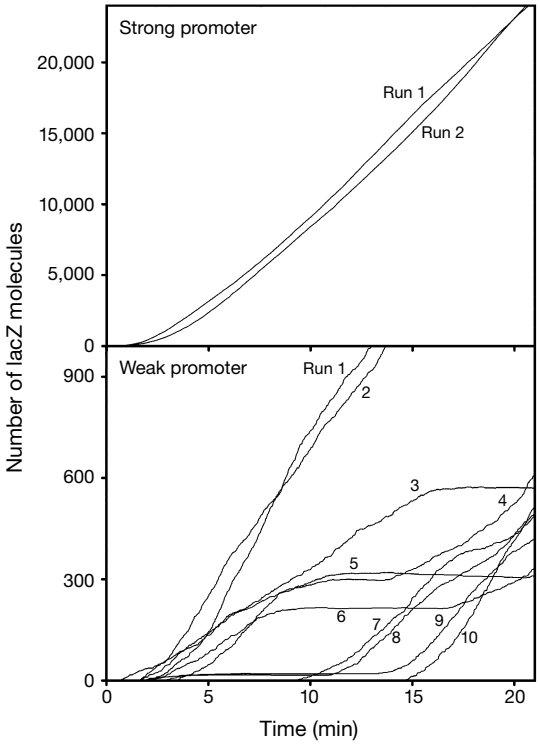

Figure 3 Stochastic simulation of lacZ expression. A simplified model of lacZ expression (after refs. 13,15) was implemented within our FPGA device. Dynamics of the model were simulated at two levels of the promoter strength. The figure shows how the amount of the synthesized protein changed during independent runs (two traces for the strong and ten for the weak promoter). Notice the drastic increase of the noise level for the weak promoter case, qualitatively in agreement with the results of the simulations with the Gillespie algorithm[13].

METHODS

All designs were implemented on D2/DIO2 boards (Digilent) that use SpartanII-200 FPGA (Xilinx). A USB interface based on USB MOD2C module (Elexol Pty) was designed to speed up communication between the FPGA and a Linux workstation used to collect and process data generated during simulation runs.

The designs were described in VHDL hardware description language and processed within the ISE 6.1i (Xilinx) development environment to generate binary files ready to be downloaded into FPGA. Both the VHDL description of the designs, as well as auxiliary programs used to initialize and control the simulations, are available upon request.

ACKNOWLEDGMENTS

We thank Robert Grothe for helpful discussion, Stefan Czarnecki for engineering advice and an anonymous reviewer for valuable comments. This work was supported by the Howard Hughes Medical Institute, Department of Energy and National Institutes of Health.

COMPETING INTERESTS STATEMENT

The authors declare that they have no competing financial interests.

Received 16 March; accepted 7 May 2004
Published online at http://www.nature.com/naturebiotechnology/

1. Hartwell, L.H., Hopfield, J.J., Leibler, S. & Murray, A.W. From molecular to modular cell biology. *Nature* **402**, C47–C52 (1999).
2. Bray, D. Protein molecules as computational elements in living cells. *Nature* **376**, 307–312 (1995).
3. Tyson, J.J., Chen, K.C. & Novak, B. Sniffers, buzzers, toggles and blinkers: dynamics of regulatory and signaling pathways in the cell. *Curr. Opin. Cell Biol.* **15**, 221–231 (2003).
4. Levchenko, A. Dynamical and integrative cell signaling: Challenges for the new biology. *Biotechnol. Bioeng.* **84**, 773–782 (2003).
5. Gillespie, D.T. Exact stochastic simulation of coupled chemical-reactions. *J. Phys. Chem. Us.* **81**, 2340–2361 (1977).
6. Gibson, M.A. & Bruck, J. Efficient exact stochastic simulation of chemical systems with many species and many channels. *J. Phys. Chem. A* **104**, 1876–1889 (2000).
7. Haseltine, E.L. & Rawlings, J.B. Approximate simulation of coupled fast and slow reactions for stochastic chemical kinetics. *J. Chem. Phys.* **117**, 6959–6969 (2002).
8. Gillespie, D.T. & Petzold, L.R. Improved leap-size selection for accelerated stochastic simulation. *J. Chem. Phys.* **119**, 8229–8234 (2003).
9. Rao, C.V. & Arkin, A.P. Stochastic chemical kinetics and the quasi-steady-state assumption: application to the Gillespie algorithm. *J. Chem. Phys.* **118**, 4999–5010 (2003).
10. Arkin, A., Ross, J. & McAdams, H.H. Stochastic kinetic analysis of developmental pathway bifurcation in phage lambda-infected *Escherichia coli* cells. *Genetics* **149**, 1633–1648 (1998).
11. Kastner, J., Solomon, J. & Fraser, S. Modeling a Hox gene network *in silico* using a stochastic simulation algorithm. *Dev. Biol.* **246**, 122–131 (2002).
12. Endy, D. & Brent, R. Modelling cellular behaviour. *Nature* **409**, 391–395 (2001).
13. Kierzek, A.M. STOCKS: STOChastic kinetic Simulations of biochemical systems with Gillespie algorithm. *Bioinformatics* **18**, 470–481 (2002).
14. Pearson, R.B. An algorithm for pseudo random number generation suitable for large-scale integration. *J. Comput. Phys.* **49**, 478–489 (1983).
15. Kierzek, A.M., Zaim, J. & Zielenkiewicz, P. The effect of transcription and translation initiation frequencies on the stochastic fluctuations in prokaryotic gene expression. *J. Biol. Chem.* **276**, 8165–8172 (2001).