B.3 Evaluation of Research and Other Scholarly Contributions

Contents

B.3.1 Research Statement

I have been pursuing research in different disciplines ranging from digital circuit design, to algorithms, to mathematics, to synthetic biology. My work tends to be *inductive* (as opposed to *deductive*) and *conceptual* (as opposed to *applied*). A broad theme is the application of computational expertise from an established discipline – electronic circuit design – to design problems with new technologies – nanoscale circuitry and biomolecular systems. Here I discuss three themes: "Computing with Molecules," "Computing with Random Bit Streams," and "Computing with Nanoscale Lattices." Please see my website for further details: http://tinyurl.com/marc-riedel-group.

1. Computing with Molecules

The theory of reaction kinetics underpins our understanding of molecular systems [11]. It is a simple and elegant formalism: molecular reactions define *rules* according to which reactants form products; each rule fires at a *rate* that is proportional to the quantities of the corresponding reactants that are present. This is illustrated in Figure 1. On the computational front, there has been a wealth of research into efficient methods for simulating molecular systems, ranging from ordinary differential equations (ODEs) [7] to stochastic simulation [10]. On the mathematical front, entirely new branches of theory have been developed to characterize the dynamics of chemical reaction networks [35].

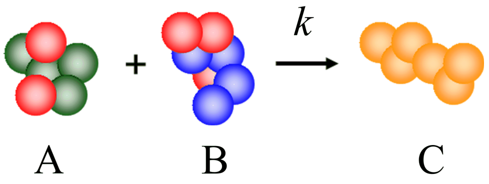


Figure 1: Molecular reactions define rules according to which reactants form products. Here molecules of type A combine with molecules of type B to form molecules of type C, at a rate proportional to the molecular concentrations of A and B as well as a rate constant k.

Most of this work is from the vantage point of *analysis*: a set of molecular reactions exists, designed by nature and perhaps modified by human engineers; the objective is to understand and characterize its behavior. Comparatively little work has been done at a conceptual level in tackling the inverse problem of *synthesis*: how can one design a set of molecular reactions that implements specific behavior?

Just as electronic systems implement computation in terms of voltage (*energy per unit charge*), molecular systems *compute* in terms of molecular concentrations (*molecules per unit volume*). As part of the community of molecular computing, my students and I have been studying techniques for implementing a variety of computational constructs with molecular reactions such as **logic**, **memory**, **arithmetic**, and **signal processing**.

Our method targets DNA-strand displacement as the experimental chassis [38]. Our contribution can be positioned as the *front-end* of the design flow [14–16,30,33]; the DNA assembler and the experimental procedure developed by Erik Winfree's group at Caltech can can be described as the *back-end* [34]. The Winfree group has shown that the kinetics of arbitrary molecular reactions can be *emulated* with DNA strand displacements. Reaction rates are controlled by designing sequences with different binding strengths. The binding strengths are controlled by the length and sequence composition of "toehold" sequences of DNA.

The impetus for this research is not computation per se. Molecular computation will never compete with conventional computers made of silicon integrated circuits for tasks such as number crunching. Chemical systems are inherently slow and messy, taking minutes or even hours to finish, and producing fragmented results. Rather, the goal is to create "**embedded controllers**" – viruses and bacteria that are engineered to perform useful molecular computation in situ, where it is needed, for instance, for drug delivery and biochemical sensing. Exciting work in this vein includes [3,28,36,37].

Still in its early stages, the field of synthetic biology has been driven by experimental expertise; much of its success has been attributable to the skill of the researchers in specific domains of biology. There has been a concerted effort to assemble repositories of standardized components. However, creating and integrating synthetic components remains an ad hoc process. The field has now reached a stage where it calls for computer-aided design tools.

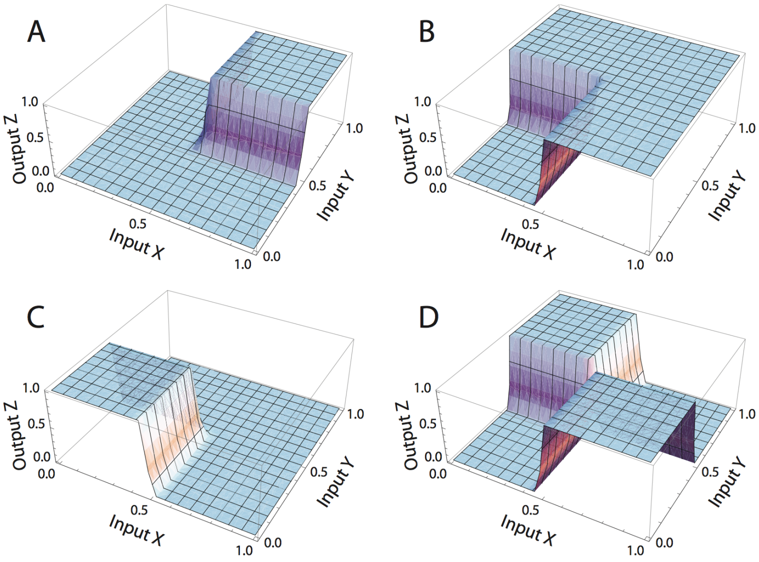


Figure 2: Simulations of our DNA implementation of logic gates. The input signals are molecular concentrations X and Y; the output signal is a molecular concentration Z. (A) AND gate, (B) OR gate, (C) NOR gate, and (D) XOR gate.

Richard Newton had a visionary view, articulated in a talk that he gave shortly before his death in 2007, titled "The Future Is Bio-Design Automation" – a view reprised by Jan Rabaey in his keynote speech at the Design Automation Conference in 2007 commemorating Newton. The view is that synthetic biology represents but a new technological substrate for design automation. For electronics, we have a design methodology that involves clear abstractions, standardized interfaces, a constrained design space, and availability of intellectual property. The same requirements exist in biology; designers need to build models, compress them for analysis, and synthesize them into substrates such as *E. coli* or yeast. As Rabaey commented, "the potential synergy with electronic design automation (EDA) is huge."

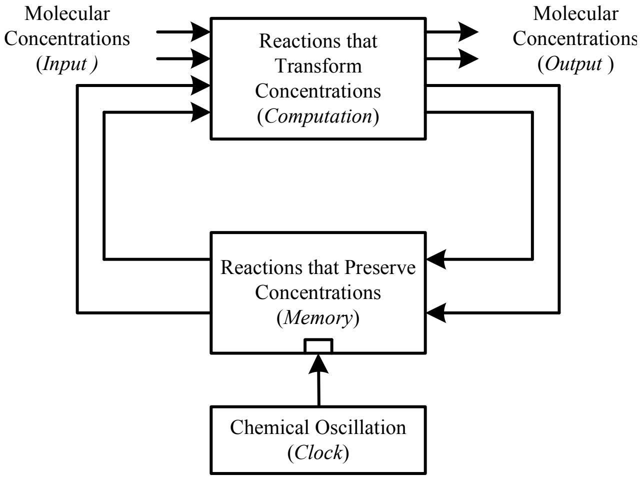


Figure 3: Block diagram of a synchronous sequential system for digital signal processing with molecular reactions.

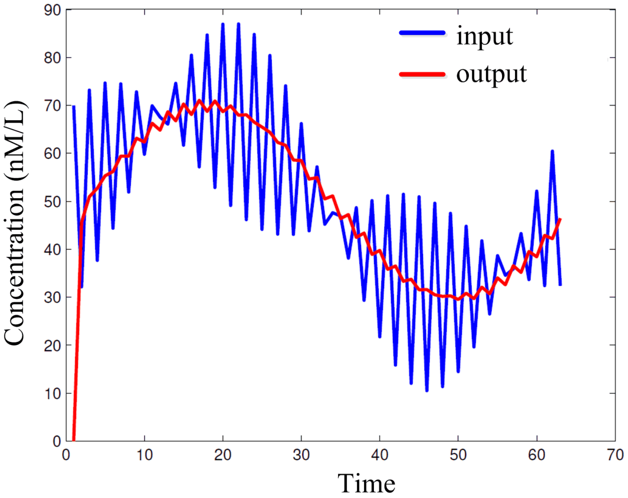


Figure 4: Simulations of our DNA implementation of a moving-average FIR filter. This filter removes the high-frequency component from an input signal, producing an output signal consisting of only the low-frequency component. Here the "signals" are molecular concentrations.

1.(a) Digital Logic

My students and I have developed a novel methodology for implementing digital logic with molecular reactions. In our scheme, a low concentration of a molecular type corresponds to a logical value of zero; a high concentration corresponds to a logical value of one. Based on a bistable mechanism for representing bits, we implement a constituent set of logical components, including combinational components such as AND, OR, and XOR gates, as well as sequential components such as D latches and D flip-flops. Simulation results for our DNA logic gates are shown in Figure 2. Using these components, we build full-fledged digital circuits such as a binary counters and linear feedback shift registers [15].

1.(b) Arithmetic & Programming Constructs

We have developed a strategy for implementing arithmetic with molecular reactions – operations such as increments and decrements, multiplication, logarithms, and exponentiation [33]. Building on these results, we have developed a compiler that translates arbitrary constructs from a C-like language into a robust implementation with molecular reactions [31].

Unlike previous schemes for biomolecular computation, ours produces designs that are dependent only on coarse rate categories for the reactions ("fast" and "slow"). Given such categories, the computation is exact and independent of the specific reaction rates. In particular, it does not matter how fast any "fast" reaction is relative to another, or how slow any "slow" reaction is relative to another – only that “fast” reactions are fast relative to “slow” reactions.

1.(c) Digital Signal Processing

We have developed a methodology for implementing digital signal processing (DSP) operations such as filtering with molecular reactions [14]. From a DSP specification, we demonstrate how to synthesize reactions that produce time-varying output quantities of molecules as a function of time-varying input quantities.

The general structure of our design is illustrated in Figure 3. As in an electronic system, our molecular system consists of separate sets of reactions for *computation* and for *memory.* A clock signal synchronizes transfers between the two. To generate the clock signal, we design reactions that produce sustained oscillations in the molecular concentrations.

We have demonstrated robust designs for both Finite-Impulse Response (FIR) and Infinite-Impulse Reponses (IIR) filters. The simulation results from our DNA implementation of a moving-average FIR filter are shown in Figure 4.

2. Computing with Random Bit Streams

Humans are accustomed to counting in a positional number system – decimal radix. Nearly all computer systems operate on another positional number system – binary radix. From the standpoint of *representation*, such positional systems are compact: given a radix b, one can represent b^n distinct numbers with n digits. Each choice of the digits $d_i \in \{0, \ldots, b-1\}$, $i = 0, \ldots, n-1$, results in a different number N in $[0, \ldots, b^n - 1]$:

$$N = \sum_{i=0}^{n-1} b^i d_i.$$

However, from the standpoint of *computation*, positional systems impose a burden: for each operation such as addition or multiplication, the signal must be “decoded,” with each digit weighted according to its position. The result must be “re-encoded” back into positional form.

We have developed a novel paradigm for digital computation that is based on a *stochastic* representation of data: a real-valued number x ($0 \leq x \leq 1$) is represented by a sequence of random bits, each of which has probability x of being one and probability $1 - x$ of being zero. These bits can either be serial streaming on a single wire or in parallel on a bundle of wires. When serially streaming, the signals are probabilistic in *time*, as illustrated in Figure 5(a); when in parallel, they are probabilistic in *space*, as illustrated in Figure 5(b). Here we frame the discussion in terms of serial bit streams. However, our approach is equally applicable to parallel wire bundles. Indeed, we have advocated this sort of stochastic representation for technologies such as nanowire crossbar arrays [22].

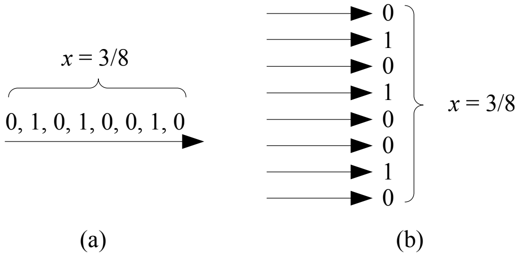


Figure 5: Stochastic representation: (a) a stochastic bit stream; (b) a stochastic wire bundle. A real value x in the unit interval $[0, 1]$ is represented as a bit stream or a bundle. For each bit in the bit stream or the bundle, the probability that it is one is x.

Consider the operation of multiplication. Any student who has designed a binary multiplier in a course on logic design can appreciate all the complexity that goes into wiring up such an operation. A conventional design for a 3-bit carry-save multiplier operating on binary radix-encoded numbers consists of perhaps 9 AND gates, 3 half adders, and 3 full adders, for a total of 30 gates. Figure 6

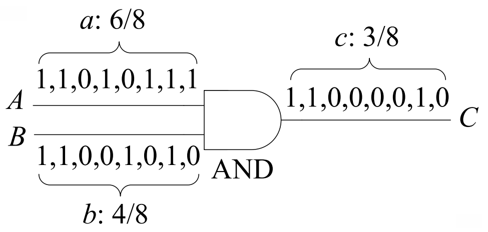


Figure 6: **Multiplication** on stochastic bit streams with an AND gate. The inputs are stochastic bit streams A and B and the output is a stochastic bit stream C. Here, the probability of A is 6/8 and that of B is 4/8. The probability of $C = 6/8 \times 4/8 = 3/8$, as expected.

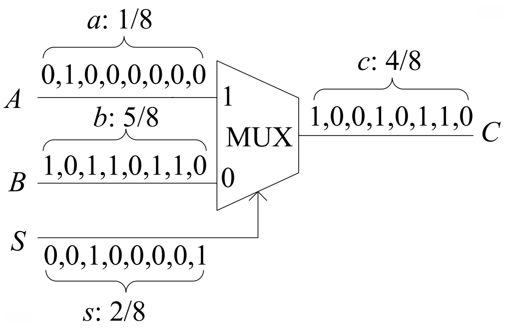


Figure 7: **Scaled addition** on stochastic bit streams, with a multiplexer (MUX). S is the selecting input. When $S = 1$, the output $C = A$. Otherwise, when $S = 0$, the output $C = B$. Here the inputs are $1/8, 5/8$, and $2/8$. The output is $2/8 \times 1/8 + (1 - 2/8) \times 5/8 = 4/8$, as expected.

shows a stochastic multiplier: it consists of but a *single* AND gate. The inputs are two independent input stochastic bit streams A and B. The number represented by the output stochastic bit stream C is

$$\begin{aligned} c &= P(C = 1) = P(A = 1 \text{ and } B = 1) \\ &= P(A = 1)P(B = 1) \\ &= a \cdot b. \end{aligned} \tag{1}$$

The probability of getting a one at the output, $P(C = 1)$, is equal to the probability of simultaneously getting ones at the inputs, namely, $P(A = 1)$ times $P(B = 1)$. So the AND gate multiplies the two values represented by the stochastic bit streams. Multiplication is simple and efficient in the stochastic representation precisely because the representation is uniform; no decoding and no re-encoding are required to operate on the values.

Consider the operation of addition implemented stochastically. It is not feasible to add two probability values directly; this could result in a value greater than one, which cannot be represented as a probability value. However, we can perform *scaled* addition. Figure 7 shows a scaled adder operating on real numbers in the stochastic representation. It consists of a multiplexer (MUX), a digital construct that selects one of its two input values to be the output value, based on a third "selecting" input value. For the multiplexer shown in Figure 7, S is the selecting input. When $S = 1$, the output $C = A$. Otherwise, when $S = 0$, the output $C = B$. With the assumption that the three input stochastic bit streams A, B, and S are independent, the number represented by the output stochastic bit stream C is

$$\begin{aligned} c &= P(C = 1) \\ &= P(S = 1 \text{ and } A = 1) + P(S = 0 \text{ and } B = 1) \\ &= P(S = 1)P(A = 1) + P(S = 0)P(B = 1) \\ &= s \cdot a + (1 - s) \cdot b. \end{aligned} \tag{2}$$

Thus, with this stochastic representation, the computation performed by a multiplexer is the scaled addition of the two input values a and b, with a scaling factor of s for a and $1 - s$ for b.

The task of *analyzing* combinational circuitry operating on stochastic bit streams is well understood [21]. For instance, it can be shown that, given an input x, an inverter (i.e., a NOT gate) implements the operation $1 - x$. Given inputs x and y, an OR gate implements the operation

$x + y - xy$. Aspects such as signal correlations of reconvergent paths must be taken into account. Algorithmic details for such analysis were first fleshed out by the testing community [29]. They have also found mainstream application for tasks such as timing and power analysis [18, 20].

We have tackled the more challenging task of *synthesizing* logical computation on stochastic bit streams that implements the functionality that we want. Naturally, since we are mapping probabilities to probabilities, we can only implement functions that map the unit interval $[0, 1]$ onto the unit interval $[0, 1]$. We focus on combinational circuits, that is to say, memoryless digital circuits built with logic gates such as AND, OR, and NOT. For such circuits, suppose that we supply stochastic bit streams as the inputs; we will observe stochastic bit streams at the outputs. Accordingly, combinational circuits can be viewed as constructs that accept real-valued probabilities as inputs and compute real-valued probabilities as outputs.

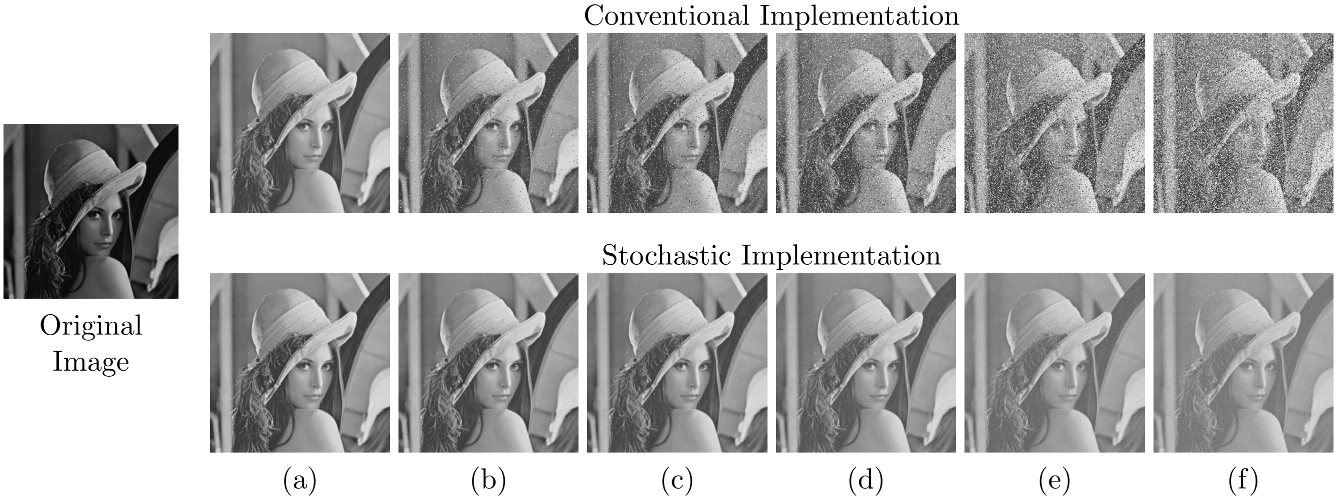


Figure 8: Fault tolerance for the gamma correction function. The images in the top row are generated by a conventional implementation. The images in the bottom row are generated by our stochastic implementation. Soft errors are injected at a rate of (a) 0%; (b) 1%; (c) 2%; (d) 5%; (e) 10%; (f) 15%.

Based on constructs for multiplication and scaled addition, we can readily implement polynomial functions of a specific form, namely polynomials with non-negative coefficients that sum up to a value of no more than one. What if the target function is a polynomial that is not decomposable this way? Suppose that it maps the unit interval onto the unit interval but it has some coefficients less than zero or some greater than one.

We have proposed a general method for synthesizing arbitrary polynomial functions on stochastic bit streams [23, 24]. We have shown that the necessary condition is also sufficient: we provided a constructive method for implementing any polynomial that satisfies the condition that the target maps the unit interval onto the unit interval. Our method is based on some novel mathematics for manipulating polynomials in a special form called *Bernstein polynomials*. In [26] we showed how to convert a general power-form polynomial into a Bernstein polynomial with coefficients in the unit interval.

Our strategy for computation has a pseudo *analog* character, reminiscent of computations performed by physical systems such as electronics on continuously variable signals such as voltage. In our case, the variable signal is the probability of obtaining a one in a stochastic yet *digital* bit stream. Indeed, our system could be built from ordinary, cheap digital electronics such as CMOS.

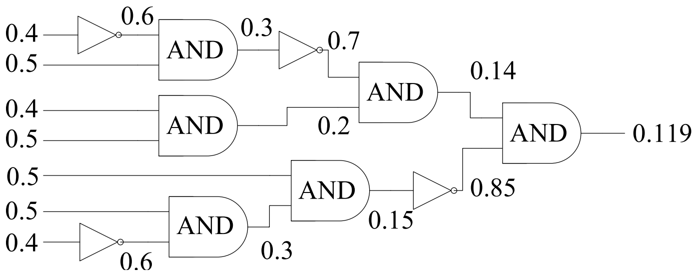


Figure 9: A circuit synthesized by our algorithm to realize the decimal output probability 0.119 from the input probabilities 0.4 and 0.5.

This is certainly counterintuitive: why impose an analog view on digital values? As we have demonstrated, it is often advantageous to do so, both from the standpoint of the hardware resources required as well as the error tolerance of the computation. Since stochastic bit streams are uniform, with no bit privileged above any other, the computation is highly error tolerant. As higher and higher rates of bit flips occur, the accuracy degrades gracefully. The images in Figure 8 illustrate the fault tolerance of stochastic computation. When soft errors are injected at a rate of 15%, the image generated by the conventional method is full of noisy pixels, while the image generated by the stochastic method is still recognizable.

Schemes for probabilistic computation can exploit physical sources to generate random values in the form of bit streams. Generally, each source has a fixed bias and so provides bits that have a specific probability of being one versus zero. If many different probability values are required, it can be difficult or expensive to generate all of these directly from physical sources. We have demonstrated novel techniques for synthesizing combinational logic that transforms a set of source probabilities into different target probabilities [25, 27]. This is illustrated in Figure 9.

3. Computing with Nanoscale Lattices

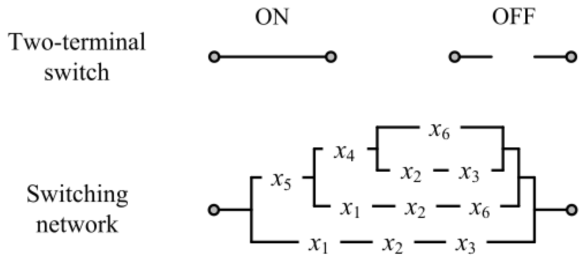


Figure 10: Two-terminal switching network implementing the Boolean function $x_1x_2x_3 + x_1x_2x_5x_6 + x_2x_3x_4x_5 + x_4x_5x_6$.

In his seminal Master's Thesis, Claude Shannon made the connection between Boolean algebra and switching circuits [32]. He considered **two-terminal switches** corresponding to electromagnetic relays. An example of a two-terminal switch is shown in the top part of Figure 10. The switch is either ON (closed) or OFF (open).

A Boolean function can be implemented in terms of connectivity across a network of switches, often arranged in a series/parallel configuration. An example is shown in the bottom part of Figure 10. Each switch is controlled by a Boolean literal. If the literal is 1 (0) then the corresponding switch is ON (OFF). The Boolean function for the network evaluates to 1 if there is a closed path between the left and right nodes. It can be computed by taking the sum (OR) of the product (AND) of literals along each path. In the figure, these products are $x_1x_2x_3$, $x_1x_2x_5x_6$, $x_2x_3x_4x_5$, and $x_4x_5x_6$.

We have been studying methods for synthesizing Boolean functions with networks of **four-terminal switches** [1, 2]. Although conceptually general, our model is applicable to a variety of nanoscale technologies, such as nanowire crossbar arrays [4] and magnetic switch-based structures [17]. General features of nanowire technology are illustrated in Figure 11. The connections between horizontal and vertical wires are FET-like junctions. When high or low voltages are applied to input nanowires, the FET-like junctions that cross these develop a high or low impedance, respectively.

Nanowire switches are, of course, assembled in large arrays. Indeed, the impetus for nanowire-based technology is the potential density, scalability and manufacturability [5, 12, 19]. Many other novel and emerging technologies fit the general model of four-terminal switches. For instance, researchers are investigating *spin waves* [8]. Unlike conventional circuitry such as CMOS that transmits signals electrically, spin-wave technology transmits signals as propagating disturbances in the ordering of magnetic materials. Potentially, spin-wave based logic circuits could compute with significantly less power than conventional CMOS circuitry.

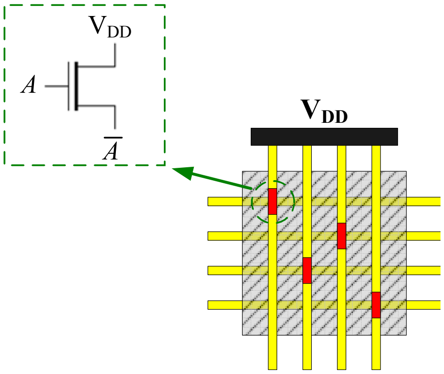


Figure 11: A nanowire crossbar architecture.

Our model is illustrated in the top part of Figure 12. The four terminals of the switch are all either mutually connected (ON) or disconnected (OFF). We consider networks of four-terminal switches arranged in rectangular *lattices*. An example is shown in the bottom part of Figure 12. Again, each switch is controlled by a Boolean literal. If the literal takes the value 1 (0) then corresponding switch is ON (OFF). The Boolean function for the lattice evaluates to 1 iff there is a closed path between the top and bottom edges of the lattice. Again, the function is computed by taking the sum of the products of the literals along each path. These products are $x_1x_2x_3$, $x_1x_2x_5x_6$, $x_2x_3x_4x_5$, and $x_4x_5x_6$ – the same as those in Figure 10. We conclude that this lattice of four-terminal switches implements the same Boolean function as the network of two-terminal switches in Figure 10.

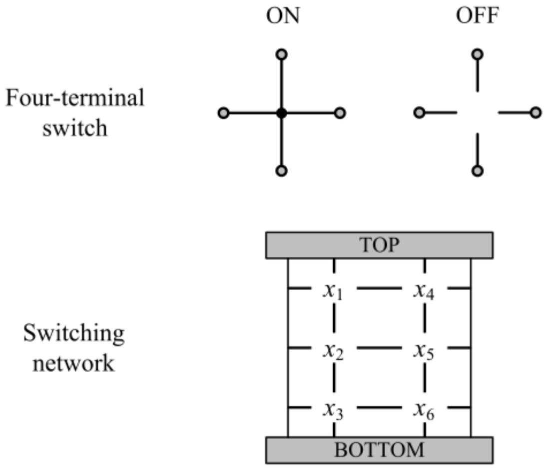
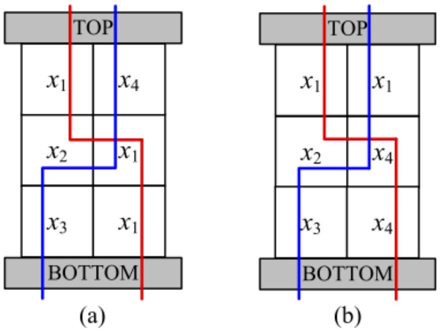


Figure 12: Four-terminal switching network implementing the Boolean same function $x_1x_2x_3 + x_1x_2x_5x_6 + x_2x_3x_4x_5 + x_4x_5x_6$.



(a) (b)

Figure 13: Two 3×2 lattices implementing different Boolean functions.

We have addressed the following synthesis problem: how should we assign literals to switches in a lattice in order to implement a given target Boolean function [1]? Suppose that we are asked to implement the function $f(x_1, x_2, x_3, x_4) = x_1x_2x_3 + x_1x_4$. We might consider the lattice in Figure 13(a). The product of the literals in the first column is $x_1x_2x_3$; the product of the literals in the second column is x_1x_4. We might also consider the lattice in Figure 13(b). The products for its columns are the same as those for (a). In fact, the two lattices implement two different functions, only one of which is the intended target function. To see why this is so, note that we must consider all possible paths, including those shown by the red and blue lines. In (a) the product x_1x_2 corresponding to the path shown by the red line covers the product $x_1x_2x_3$ so the function is $f_a = x_1x_2 + x_1x_4$. In (b) the products $x_1x_2x_4$ and $x_1x_2x_3x_4$ corresponding to the paths shown by the red and blue lines are redundant, covered by column paths, so the function is $f_b = x_1x_2x_3 + x_1x_4$.

In this example, the target function is implemented by a 3×2 lattice with four paths. If we were given a target function with more products, a larger lattice would likely be needed to implement it; accordingly, we would need to enumerate more paths. Here the problem is that the number of paths grows exponentially with the lattice size. Any synthesis method that enumerates paths quickly becomes intractable.

We have presented an efficient algorithm for this task – one that does not exhaustively enumerate paths but rather exploits the concept of Boolean function *duality* [9, 13]. Our algorithm produces lattices with a size that grows linearly with the number of products of the target Boolean function. It runs in time that grows polynomially.

A significant tangent for this work is its mathematical contribution: lattice-based implementations present a novel view of the properties of Boolean functions. We are studying the applicability of these properties to the famous problem of testing whether two monotone Boolean functions in irredundant sum-of-product form are dual. This is one of the few problems in circuit complexity whose precise tractability status is unknown [6].

B.3.2 Awards and Recognition

- I received the **Faculty Early Career Development (CAREER) Award** from the National Science Foundation. This is the NSF's most prestigious award in support of junior faculty.
- My paper "The Synthesis of Combinational Logic to Generate Probabilities" was nominated for the **IEEE/ACM William J. McCalla Best Paper Award** at the International Conference on Computer-Aided Design (ICCAD).
- My paper "The Synthesis of Robust Polynomial Arithmetic with Stochastic Logic" was nominated as a **Research Highlight** by Communications of the ACM.
- My paper "The Synthesis of Cyclic Combinational Circuits" received the **Best Paper Award** at the Design Automation Conference (DAC).
- I received the Charles H. Wilts Prize for the **Best Doctoral Research** in Electrical Engineering at Caltech.

B.3.3 Research Funding

I have been awarded four major grants from the National Science Foundation (NSF) and from the Semiconductor Research Corporation (SRC), totaling $1.5 million. These grants include support for research in molecular computing (from the NSF BIO Computing Program and the NSF Core Computing and Communication Foundations Program); for interdisciplinary applications of design automation (from the NSF Design Automation for Micro and Nano Systems Program); and for computing with random circuits and percolation (from the SRC Focus Center Research Program on Functional Engineered Nano-Architectonics).

I have also received significant funding from internal sources, including seed funding from the University of Minnesota Digital Technology Center and student support from the Biomedical Informatics and Computational Biology (BICB) program, a joint University of Minnesota, Mayo Clinic, and IBM Rochester initiative.

For each source, my percentage of the total amount is listed in brackets.

External Sponsored Funding

- Agency: National Science Foundation
  Program: BIO Computing
  Title: "Digital Signal Processing with Biomolecular Reactions"
  Investigators: Keshab Parhi (PI) and Marc Riedel (co-PI)
  Amount: $400,000 [50%]
  Duration: 2011–2015

- Agency: National Science Foundation
  Program: **NSF CAREER Award**
  Title: "Computing with Things Small, Wet, and Random – Design Automation for Digital Computation with Nanoscale Technologies and Biological Processes"
  Investigator: Marc Riedel (PI)
  Amount: $500,000 [100%]
  Duration: 2009–2014

- Agency: National Science Foundation
  Program: Design Automation for Micro and Nano Systems, EAGER Grant
  Title: "Synthesizing Signal Processing Functions with Biochemical Reactions"
  Investigators: Keshab Parhi (PI) and Marc Riedel (co-PI)
  Amount: $200,000 [50%]
  Duration: 2009–2011

- Agency: SRC Focus Center Research Program (FCRP)
  Program: Functional Engineered Nano-Architectonics (FENA)
  Title: "The Concurrent Logical and Physical Design of Nanoscale Digital Circuits"
  Investigator: Marc Riedel (PI)
  Amount: $325,000 [100%]
  Duration: 2007–2010

University Sources

- Agency: University of Minnesota, Digital Technology Center
  Program: Digital Technology Initiatives (DTI) Seed Grant
  Title: "Computational Method for Forward Biological Engineering"
  Investigators: Y. Kaznessi (PI), C. Schmidt-Dannert (co-PI), and M. Riedel (co-PI)
  Amount: $97,800 [25%]
  Duration: 2011–2012

- Agency: University of Minnesota
  Program: Biomedical Informatics and Computational Biology (BICB)
  Funding: Student Traineeships for Brian Fett and Adrianna Fitzgerald
  Investigator: Marc Riedel (PI)
  Amount: $78,000 [100%]
  Duration: 2007–2009

Excerpts from Reviews

Some excerpts from the reviews of my NSF proposal titled "*Computing with Things Small, Wet, and Random – Design Automation for Digital Computation with Nanoscale Technologies and Biological Processes*" are as follows:

- "The panel concurred that this proposal was potentially highly transformative. The panel felt that if this proposal succeeds, it could really transform both computer system design based on nanotechnology, and the way synthetic biology is carried out."
- "Potentially one of the most transformative proposals I have seen, very ambitious, and at some moments even brilliant."
- "The background of the PI is outstanding. His work is both original, deep, and very clearly explained."
- "This is an excellent proposal. The PI has done superb work to show the feasibility of the project, and his thesis is insightful and constructive."
- "The panel appreciated the excellence of the PI's published work, and the bold vision expressed in the write up."

Some excerpts from the reviews of my NSF Proposal titled "*Digital Signal Processing with Biomolecular Reactions*" are as follows:

- "Intellectually this is a very interesting and highly innovative project. This is one of the most interesting computational proposals I have seen in many years."
- "This is an excellent proposal to design biochemical reactions to implement filters for digital signal processing. The investigators have an excellent track record. The outreach and education plan is very good."
- "The investigators are outstanding and have a great track record. I believe they will be able to accomplish what they propose."

B.3.4 Publications & Patents

I have published on a wide variety of topics spanning a range of disciplines from computer-aided design, to computational biology, to mathematics, to synthetic biology, to communications, to theoretical computer science, to control theory. I have published a total of 45 peer-reviewed papers (with 7 more currently under review):

- Four papers in IEEE Transactions (IEEE Transactions on Computers, IEEE Transactions on CAD of Integrated Circuits & Systems, and IEEE Transactions on Parallel and Distributed Computing).
- Two papers in mathematics journals (Journal of Discrete Applied Mathematics and the European Journal of Combinatorics). Both are top journals.
- Three papers in journals with a biology readership (PLoS ONE, the Journal of Systems and Synthetic Biology, and the International Journal of Nanotechnology and Molecular Computation).

- **12 papers** at the **Design Automation Conference** and the **International Conference on Computer-Aided Design** These conferences are the most prestigious and selective publication venues for research in computer-aided design of circuits, with acceptance rates in the range of 20% to 25%. They only accept work with complete results, and so the papers have an archival quality. My DAC and ICCAD papers span a range of topics from logic synthesis, to molecular computing, to nanoscale lattices. Indeed, I have brought a significant interdisciplinary dimension to these conferences.
- A paper at the Asia and South Pacific Design Automation Conference. This conference is nearly as competitive as DAC and ICCAD.
- A paper at the Allerton Conference on Communication, Control, and Computing, a well-regarded conference in the area of controls.
- A paper at the Asilomar Conference on Signals, Systems, and Computers, a well-regarded conference in the area of communications.
- Four papers at the Pacific Symposium on Biocomputing, an international, multidisciplinary conference with archival proceedings.
- Two papers at IEEE Symposia (Great Lakes Symposium on VLSI and International Symposium on Quality Electronic Design).
- 11 papers at the International Workshop on Logic and Synthesis. The workshop is competitive, with a rigorous peer-review of full-length papers and an acceptance rate of around 35%.
- Two papers at other workshops (IEEE Workshop on Signal Processing Systems and the International Workshop on Synthesis and System Integration of Mixed Information Technologies).
- Two book chapters.

For what follows, my advisees are denoted with (†). If I was a primary author on the paper, this is indicated with (1); if I was a secondary author, this is indicated with (2).

Peer-Reviewed Journal Articles and Book Chapters

1. "Logic Synthesis for Switching Lattices"
   Mustafa Altun† and Marc Riedel(1)
   IEEE Transactions on Computers, 13 pages, to appear, 2011
2. "Characterizing the Memory of the GAL Regulatory Network in *Saccharomyces cerevisiae*"
   Vishwesh Kulkarni,† Venkatesh Kareenhalli, Ganesh Viswananthan, and Marc Riedel(1)
   Systems and Synthetic Biology, 13 pages, to appear, 2011
3. "Cyclic Boolean Circuits"
   Marc Riedel(1) and Jehoshua Bruck
   Journal of Discrete Applied Mathematics, 42 pages, to appear (pending minor revision), 2011
4. "Transforming Probabilities with Combinational Logic"
   Weikang Qian,† Marc Riedel,(1) Hongchao Zhou, and Jehoshua Bruck
   IEEE Transactions on CAD of Integrated Circuits & Systems, 14 pages, to appear 2011

5. "Rate-Independent Constructs for Chemical Computation"
   Philip Senum† and Marc Riedel(1)
   PLoS ONE, Vol. 6, Issue 6, pp. 1–12, 2011

6. "Uniform Approximation and Bernstein Polynomials with Coefficients in the Unit Interval"
   Weikang Qian,† Marc Riedel,(1) and Ivo Rosenberg
   European Journal of Combinatorics, Vol. 32, No. 3, pp. 448–463, 2011

7. "An Architecture for Fault-Tolerant Computation with Stochastic Logic"
   Weikang Qian,† Xin Li, Marc Riedel,(1) Kia Bazargan, and David Lilja
   IEEE Transactions on Computers, Vol. 60, No. 1, pp. 93–105, 2011

8. "Synthesizing Combinational Logic to Generate Probabilities: Theories and Algorithms"
   Weikang Qian,† Marc Riedel,(1) Kia Bazargan, and David Lilja
   Advanced Techniques in Logic Synthesis, Optimizations and Applications
   Sunil Khatri and Kanupriya Gulati, Editors, Springer Publishing, pp. 1–28, 2010

9. "The Synthesis of Stochastic Logic for Nanoscale Digital Circuits"
   Weikang Qian,† John Backes,† and Marc Riedel(1)
   International Journal of Molecular and Nanoscale Computation
   Vol. 1, Issue 4, pp. 39–57, 2010

10. "Computing in the RAIN: A Reliable Array of Independent Nodes"
    Vasken Bohossian, Charles Fan, P. LeMahieu, Marc Riedel,(1) Lihao Xu, and Jehoshua Bruck
    IEEE Transactions on Parallel and Distributed Computing, Vol. 12, No. 2, pp. 99–114, 2001

11. "Tolerating Faults in Counting Networks"
    Marc Riedel(1) and Jehoshua Bruck
    Dependable Network Computing, Dimiter Avresky, Editor
    Kluwer Academic Publishing, pp. 267–278, 2000

Peer-Reviewed Conference Papers

1. "The Synthesis of Linear Finite State Machine-Based Stochastic Computational Elements"
   Peng Li, Weikang Qian,† Marc Riedel,(2) Kia Bazargan, and David Lilja
   ACM/IEEE Asia and South Pacific Design Automation Conference, 8 pages, 2012

2. "Networks of Passive Oscillators"
   Vishwesh Kulkarni,† Marc Riedel,(1) and Guy-Bart Stan
   Allerton Conference on Communication, Control, and Computing, 7 pages, 2011

3. "Asynchronous Sequential Computation with Molecular Reactions"
   Hua Jiang,† Marc Riedel,(1) and Keshab Parhi
   Asilomar Conference on Signals, Systems, and Computers, 8 pages, 2011

4. "Synchronous Sequential Computation with Molecular Reactions"
   Hua Jiang,† Marc Riedel,(1) and Keshab Parhi
   ACM/IEEE Design Automation Conference, 6 pages, 2011

5. "Rate-Independent Constructs for Chemical Computation"
   Philip Senum† and Marc Riedel(1)
   Pacific Symposium on Biocomputing, 11 pages, 2011

6. "Binary Counting with Chemical Reactions"
Aleksandra Kharam,† Hua Jiang,† Marc Riedel,(1) and Keshab Parhi
Pacific Symposium on Biocomputing, 12 pages, 2011

7. "Reduction of Interpolants for Logic Synthesis"
John Backes† and Marc Riedel(1)
IEEE/ACM International Conference on Computer-Aided Design, 8 pages, 2010

8. "Digital Signal Processing with Biomolecular Reactions"
Hua Jiang,† Aleksandra Kharam,† Marc Riedel,(1) and Keshab Parhi
IEEE/ACM International Conference on Computer-Aided Design, 8 pages, 2010

9. "Lattice-Based Computation of Boolean Functions"
Mustafa Altun† and Marc Riedel(1)
ACM/IEEE Design Automation Conference, 6 pages, 2010

10. "Writing and Compiling Code into Biochemistry"
Adam Shea,† Brian Fett,† Marc Riedel,(1) and Keshab Parhi
Pacific Symposium on Biocomputing, 9 pages, 2010

11. "The Synthesis of Combinational Logic to Generate Probabilities"
Weikang Qian,† Marc Riedel,(1) Kia Bazargan, and David Lilja
IEEE/ACM International Conference on Computer-Aided Design, 8 pages, 2009
(Nominated for **IEEE/ACM William J. McCalla Best Paper Award**)

12. "Synthesizing Sequential Register-Based Computation with Biochemistry"
Adam Shea,† Brian Fett,† Marc Riedel,(1) and Keshab Parhi
IEEE/ACM International Conference on Computer-Aided Design, 8 pages, 2009

13. "Nanoscale Computation Through Percolation"
Mustafa Altun,† Marc Riedel,(1) and Claudia Neuhauser
ACM/IEEE Design Automation Conference, WACI Track, 2 pages, 2009

14. "A Reconfigurable Stochastic Architecture for Reliable Computing"
Xin Li, Weikang Qian,† Marc Riedel,(2) Kia Bazargan, and David Lilja
IEEE Great Lakes Symposium on VLSI Design, 6 pages, 2009

15. "Estimation and Optimization of Reliability of Noisy Digital Circuits"
Satish Sivaswamy, Kia Bazargan, and Marc Riedel(2)
IEEE International Symposium on Quality Electronic Design, 6 pages, 2009

16. "Stochastic Transient Analysis of Biochemical Systems"
Bin Cheng† and Marc Riedel(1)
Pacific Symposium on Biocomputing, 11 pages, 2009

17. "Module Locking in Biochemical Synthesis"
Brian Fett† and Marc Riedel(1)
IEEE/ACM International Conference on Computer-Aided Design, 7 pages, 2008

18. "The Analysis of Cyclic Circuits with Boolean Satisfiability"
John Backes† and Marc Riedel(1)
IEEE/ACM International Conference on Computer-Aided Design, 7 pages, 2008

19. “The Synthesis of Robust Polynomial Arithmetic with Stochastic Logic”
    Weikang Qian† and Marc Riedel$^{(1)}$
    ACM/IEEE Design Automation Conference, 6 pages, 2008
    (Nominated as a **Research Highlight** in Communications of the ACM, 2010)

20. “Synthesizing Stochasticity in Biochemical Systems”
    Brian Fett,† Jehoshua Bruck, and Marc Riedel$^{(1)}$
    ACM/IEEE Design Automation Conference, 6 pages, 2007

21. “The Synthesis of Cyclic Combinational Circuits”
    Marc Riedel$^{(1)}$ and Jehoshua Bruck
    ACM/IEEE Design Automation Conference, 6 pages, 2003
    (Received the **DAC Best Paper Award**)

Peer-Reviewed Workshop Papers

1. “Resolution Proofs as a Data Structure for Logic Synthesis”
   John Backes† and Marc Riedel$^{(1)}$
   IEEE/ACM International Workshop on Logic and Synthesis, 8 pages, 2011

2. “Synthesizing Cubes to Satisfy a Given Intersection Pattern”
   Weikang Qian† and Marc Riedel$^{(1)}$
   IEEE/ACM International Workshop on Logic and Synthesis, 8 pages, 2010

3. “Two-Level Logic Synthesis for Probabilistic Computation”
   Weikang Qian† and Marc Riedel$^{(1)}$
   IEEE/ACM International Workshop on Logic and Synthesis, 8 pages, 2010

4. “Reduction of Interpolants for Logic Synthesis”
   John Backes† and Marc Riedel$^{(1)}$
   IEEE/ACM International Workshop on Logic and Synthesis, 6 pages, 2010

5. “Digital Signal Processing with Biomolecular Reactions”
   Hua Jiang,† Aleksandra Kharam,† Marc Riedel,$^{(1)}$ and Keshab Parhi
   IEEE Workshop on Signal Processing Systems, 6 pages, 2010

6. “The Synthesis of Cyclic Dependencies with Craig Interpolation”
   John Backes† and Marc Riedel$^{(1)}$
   IEEE/ACM International Workshop on Logic and Synthesis, 7 pages, 2009

7. “Synthesizing Sequential Register-Based Computation with Biochemistry”
   Adam Shea,† Brian Fett,† Marc Riedel,$^{(1)}$ and Keshab Parhi
   IEEE/ACM International Workshop on Logic and Synthesis, 8 pages, 2009

8. “The Synthesis of Combinational Logic to Generate Probabilities”
   Weikang Qian,† Marc Riedel,$^{(1)}$ Kia Bazargan, and David Lilja
   IEEE/ACM International Workshop on Logic and Synthesis, 8 pages, 2009

9. “The Synthesis of Stochastic Logic to Perform Multivariate Polynomial Arithmetic”
   Weikang Qian† and Marc Riedel$^{(1)}$
   IEEE/ACM International Workshop on Logic and Synthesis, 8 pages, 2008

10. "The Synthesis of Stochastic Logic for Nanoscale Digital Circuits"
    Weikang Qian,† John Backes,† and Marc Riedel(1)
    IEEE/ACM International Workshop on Logic and Synthesis, 8 pages, 2007

11. "Application of LUT Cascades to Numerical Function Generators"
    Tsutomu Sasao, Jon Butler, and Marc Riedel(1)
    Workshop on Synthesis & System Integration of Mixed Information, 7 pages, 2004

12. "Timing Analysis of Cyclic Combinational Circuits"
    Marc Riedel(1) and Jehoshua Bruck
    IEEE/ACM International Workshop on Logic and Synthesis, 8 pages, 2004

13. "Cyclic Combinational Circuits: Analysis for Synthesis"
    Marc Riedel(1) and Jehoshua Bruck
    IEEE/ACM International Workshop on Logic and Synthesis, 8 pages, 2003

Patents

1. "Method and Means for the Synthesis of Cyclic Combinational Circuits"
   Marc Riedel and Jehoshua Bruck
   U.S. Patent 7,249,341

2. "A Reliable Array of Distributed Computing Nodes"
   Vincent Bohossian, Charles Fan, Paul LeMahieu, Marc Riedel, Lihao Xu, and Jehoshua Bruck
   U.S. Patent 6,128,277

B.3.5 Five Selected Publications

Of the papers listed in Section B.3.4, I have selected the following five papers for inclusion in my dossier. You can find all of my papers on my website `http://tinyurl.com/marc-riedel-papers`.

1. "Logic Synthesis for Switching Lattices"
   Mustafa Altun and Marc Riedel
   IEEE Transactions on Computers, 13 pages, to appear, 2011

2. "Characterizing the Memory of the GAL Regulatory Network in *Saccharomyces cerevisiae*"
   Vishwesh Kulkarni, Venkatesh Kareenhalli, Ganesh Viswananthan, and Marc Riedel
   Systems and Synthetic Biology, 13 pages, to appear, 2011

3. "Transforming Probabilities with Combinational Logic"
   Weikang Qian, Marc Riedel, Hongchao Zhou, and Jehoshua Bruck
   IEEE Transactions on CAD of Integrated Circuits & Systems, 14 pages, to appear 2011

4. "Rate-Independent Constructs for Chemical Computation"
   Philip Senum and Marc Riedel
   PLoS ONE, Vol. 6, Issue 6, pp. 1–12, 2011

5. "An Architecture for Fault-Tolerant Computation with Stochastic Logic"
   Weikang Qian, Xin Li, Marc Riedel, Kia Bazargan, and David Lilja
   IEEE Transactions on Computers, Vol. 60, No. 1, pp. 93–105, 2011

B.3.6 Presentations

Since 2006, my advisees and I have given approximately **80 invited** and **contributed** presentations, colloquia, and tutorials. I have given over 50 myself. This includes the conference and workshop papers listed in Section B.3.4. In also includes technical seminars at numerous universities; invited talks at international conferences and symposia; and public engagement talks. Some have been to very large audiences: I spoke to an audience of 300 hundred people at the Synthetic Biology 3.0 conference in Zürich in 2007 and a similar number at the Symposium on the Foundations of Nanoscience (FNANO) conference in Salt Lake City in 2010. Public engagement talks that I have given include a *Café Scientifique* talk at the Bryant Lake Bowl, hosted by the Bell Museum of Natural History.

For what follows, the presenter is indicated with (∗). In all cases, the presenter is either myself or one of my advisees.

Presentations with Published Abstracts

1. "Synthesizing Logical Computation on Stochastic Bit Streams for Sensing Applications"
   Marc Riedel* (**invited**)
   IEEE CANDE Workshop, San Jose, CA, 2011

2. "Digital Signal Processing with DNA"
   Hua Jiang,* Marc Riedel, and Keshab Parhi
   International Conference on DNA Computing, Pasadena, CA, 2011

3. "Synthesizing Logical Computation on Stochastic Bit Streams"
   Marc Riedel* (**invited**)
   CMOS Emerging Technologies Workshop, Whistler, BC, 2011

4. "Asynchronous Sequential Computation with Molecular Reactions"
   Hua Jiang,* Marc Riedel, and Keshab Parhi
   International Workshop on Bio-Design Automation, San Diego, CA, 2011

5. "Biological Network Reconstruction Using Literature Curated and High Throughput Data"
   Vishwesh Kulkarni,* Kalyanasundaram Subramanian, Reza Arastoo,
   Mayuresh Kothare, and Marc Riedel
   International Workshop on Bio-Design Automation, San Diego, CA, 2011

6. "Rate-Independent Constructs for DNA Computing"
   Philip Senum and Marc Riedel*
   Annual Institute of Biological Engineering Conference, Atlanta, GA, 2011

7. "Lattice-Based Computation with Percolation"
   Mustafa Altun and Marc Riedel* (**invited**)
   IEEE/ACM International Symposium on Nanoscale Architectures, Anaheim, CA, 2010

8. "Signal Processing Functions with Biomolecular Reactions"
   Hua Jiang, Marc Riedel,* and Keshab Parhi
   International Workshop on Bio-Design Automation, Anaheim, CA, 2010

9. Session Summary: "Engineering Biology: Fundamentals and Applications"
   Marc Riedel,* Soha Hassoun, and Ron Weiss (**invited**)
   ACM/IEEE Design Automation Conference, Anaheim, CA, 2010

10. "Digital Signal Processing with Biochemistry"
    Marc Riedel* (**invited**)
    Symposium on the Foundations of Nanoscience, Salt Lake City, UT, 2010

11. "Iterative Computation with Biomolecular Reactions"
    Hua Jiang, Marc Riedel,* and Keshab Parhi
    Annual Institute of Biological Engineering Conference, Boston, MA, 2010

12. "Stochastic Logic and Stochastic Biological Processes"
    Marc Riedel* (**invited**)
    Information Theory and Applications Workshop, UC San Diego, 2010

13. "Computing with Things Small, Wet, and Random"
    Marc Riedel* (**invited**)
    IEEE CANDE Workshop, Monterey, CA, 2009

14. "Stochastic Chemical Reaction Networks"
    Marc Riedel* (**invited**)
    International Workshop on Stochasticity, Banff, Alberta, 2009

15. "Synthesizing Sequential Register-Based Computation with Biochemistry"
    Adam Shea, Brian Fett, Marc Riedel,* and Keshab Parhi
    International Workshop on Bio-Design Automation, San Francisco, CA, 2009

16. "Synthesizing Circuit Constructs with Chemical Reaction Networks"
    Marc Riedel* (**invited**)
    Emergence in Chemical Systems Conference, Anchorage, AK, 2009

17. "Rate-Independent Biochemical Synthesis"
    Adam Shea, Brian Fett, and Marc Riedel*
    Annual Institute of Biological Engineering Conference, Santa Clara, CA, 2009

18. "Modular Stochastic Biochemistry"
    Brian Fett and Marc Riedel*
    Synthetic Biology 4.0, Hong Kong, 2008

19. "Biochemical Pathways from Generic Designs"
    Brian Fett and Marc Riedel*
    Synthesis of Cells Meeting, Kobe, Japan, 2008

20. "The Computer-Aided Synthesis of Stochastic Biochemistry"
    Brian Fett and Marc Riedel*
    Advances in Synthetic Biology Conference, Cambridge, UK, 2008

21. "Synthesizing Stochasticity"
    Brian Fett and Marc Riedel*
    Synthetic Biology 3.0, Zürich, Switzerland, 2007

22. "Using The Probability Gradient to Analyze Bifurcating Biochemical Systems"
    Brian Fett* and Marc Riedel
    International Conference on Systems Biology, Yokohama, Japan, 2006

23. "Exact Stochastic Simulation with Event Leaping"
    Marc Riedel* and Jehoshua Bruck
    International Conference on Systems Biology, Boston, MA, 2005

Invited Talks, Colloquia, and Panels (without published abstracts)

1. "Random and Loopy Circuits: Complexity in Electronic and Biological Circuit Design"
   Dept. of Defense Research and Engineering Complex Systems Study
   Host: Robert Bond
   Squam Lake, NH, July 27, 2010

2. Panelist: "CAD for Nanoelectronic Circuits and Architectures – Are We There Yet?"
   IEEE/ACM International Symposium on Nanoscale Architectures
   Organizer: Prof. Garrett Rose
   Anaheim, CA, June 17, 2010

3. "Robust Stochastic Computation with Biomolecular Reactions"
   NSF Workshop on Shared Organizing Principles in Biology
   Organizer: Prof. Melanie Mitchel
   Arlington, VA, May 25, 2010

4. "Computing with Things Small, Wet, and Random"
   Biological and Medical Physics Seminar Series
   Host: Prof. Vincent Noireaux
   University of Minnesota, March 30, 2010

5. "Computing with Things Small, Wet, and Random"
   Computer Science Seminar
   Host: Prof. Soha Hassoun
   Tufts University, March 1, 2010

6. Tutorial: "Programming Constructs for Chemical Reaction Networks"
   Pacific Symposium on Biocomputing
   Organizer: Dr. Gil Alterovitz
   Kona, Hawaii, Jan. 7, 2010

7. "Computing with Things Small, Wet, and Random"
   Electrical and Computer Engineering Seminar
   Host: Prof. Azadeh Davoodi
   University of Wisconsin, Feb. 27, 2009

8. "Computing with Things Small, Wet, and Random"
   Electrical and Computer Engineering Seminar
   Host: Prof. Lin Zhong
   Rice University, Feb. 17, 2009

9. "Computing with Things Small, Wet, and Random"
   Electrical and Computer Engineering Seminar
   Host: Prof. Anxiao (Andrew) Jiang
   Texas A&M University, Feb. 17, 2009

10. "Synthesizing Nearly Rate Independent Biochemical Computation"
    NSF Expeditions in Computing – Molecular Programming Workshop
    Organizer: Prof. Erik Winfree
    Oxnard, CA, Jan. 10, 2009

11. "Computing with Things Small, Wet, and Random"
    Electrical and Computer Engineering Seminar
    Host: Prof. Rick Kiehl
    UC Davis, Sep. 29, 2008

12. "Synthesizing Stochastic Logic"
    SRC Center on Functional Engineered Nano-Architectonics (FENA) Annual Meeting
    Organizer: Prof. Kang Wang
    La Jolla, CA, June 13, 2008

13. Tutorial: "Synthesizing Stochastic Biochemical Reactions"
    Tech Tune Up
    Organizer: Prof. Ahmed Tewfik
    University of Minnesota, May 26, 2008

14. "Synthesizing Stochasticity in Ciruits and in Biology"
    DARPA MTO LIBRA Workshop
    Organizer: Dr. John Damoulakis
    Arlington, VA, Nov. 29, 2007

15. Public Lecture: "Circuit Engineers Doing Biology –
    A Discourse on the Changing Landscape of Scientific Research"
    Café Scientifique Public Seminar Series, Bell Museum of Natural History
    Organizer: Peggy Korsmo-Kennon
    Bryant-Lake Bowl, Minneapolis, MN, Nov. 20, 2007

16. "High-Performance Computing for the Analysis and Synthesis of Biochemistry"
    IBM Company Seminar
    Host: Tim Mullins
    Rochester, MN, Oct. 8, 2007

17. Guest Lecture: "Molecular Computing"
    IST 4, Information and Logic
    Instructor: Prof. Jehoshua Bruck
    California Institute of Technology, May 25, 2007

18. "Analysis and Synthesis of Biochemical Reactions"
    Cadence Research Labs Seminar
    Host: Dr. Andreas Kuelmann
    Berkeley, CA, May 24, 2007

19. Tutorial: “Analysis and Synthesis of Stochastic Biochemical Reactions”
Tech Tune Up
Organizer: Prof. Kia Bazargan
University of Minnesota, May 23, 2007

20. “Analysis and Synthesis of Stochastic Logic for Nanoscale Computation”
SRC Center on Functional Engineered Nano-Architectonics (FENA) Workshop
Organizer: Prof. Kang Wang
UCLA, April 19, 2007

21. “Synthesizing Stochasticity in Biochemical Reaction Networks”
Mathematical Biology Seminar
Host: Prof. Hans Othmer
University of Minnesota, March 21, 2007

22. “Exact Stochastic Simulation with Event Leaping”
Mathematical Biology Seminar
Host: Prof. Hans Othmer
University of Minnesota, Nov. 2, 2006

23. “Cycles – The Good and the Bad in Logic Synthesis and Computational Biology”
Medtronic Technology Quarterly Seminar
Host: Sara Audet
Fridely, MN, Oct. 5, 2006

24. “Cycles – The Good and the Bad in Logic Synthesis and Computational Biology”
Electrical Engineering Seminar
Host: Prof. Mustafa Kamash
UC Santa Barbara, May 17, 2006

B.3.7 Professional Service

I have been an active participant in the organization of workshops, tutorials, conferences, and publications for organizations such as the IEEE and the ACM. Among these activities:

- I served in different capacities – Program Chair, General Chair, Publications Chair, and Panel Chair – for the **International Workshop on Logic and Synthesis** (IWLS) from 2006 to 2009. Now in its 20th year, IWLS is a well-established and well-regarded workshop with attendance in the range of 50 to 100 people. It solicits full-length papers that are peer-reviewed; acceptance is competitive. My duties as Program Chair in 2009 consisted of assembling the technical program and producing the proceedings. My duties as General Chair in 2008 included all logistical and financial aspects of the meeting.

- I initiated a new workshop, the **DAC International Workshop on Biodesign Automation** (IWBDA) in 2009. I served as Program Chair, General Chair, and Steering Committee Chair in 2009, 2010, and 2011, respectively. IWBDA provides a forum for cross-disciplinary discussion, with the aim of seeding collaboration between the synthetic biology and the electronic design communities. The broad focus is on concepts, methodologies, and software tools for the automated synthesis of novel biological functions. A specific focus is the application of computational expertise from electronic circuit design to these areas.

The workshop has been a remarkable success, with approximately **100 attendees** in 2009, **85 attendees** in 2010 and **120 attendees** in 2011. The workshop websites are

2009: `http://2009.biodesignautomation.org`
2010: `http://2010.biodesignautomation.org`
2011: `http://www.biodesignautomation.org`

I have served on the technical program committees of several conference, including the ACM/IEEE Design Automation Conference (DAC), the IEEE Great Lakes Symposium on VLSI (GLS-VLSI), and the IEEE/ACM International Conference on Computer-Aided Design (ICCAD). Also, I have served on NSF review panels.

I have been an active member of the ACM Special Interest Group on Design Automation (SIGDA), serving as Associate Editor its monthly newsletter, co-chairing its Technical Committee on Logic/RTL Design, and organizing the CAD-athlon programming competition.

Finally, I have served as a referee for numerous journals. These include: **Nature Reviews Microbiology**, the **Proceedings of the National Academy of Sciences**, IEEE Transactions on Computers, IEEE Transactions on Computer-Aided Design of Circuits and Systems, IEEE Transactions on Information Theory, ACM Transactions on Design Automation of Electronic Systems, ACM Journal on Emerging Technologies, Bioinformatics, Journal of Chemical Physics, and SCIAM Journal on Scientific Computing.

References

[1] M. Altun and M. D. Riedel. Lattice-based computation of Boolean functions. In *Design Automation Conference*, pages 609–612, 2010.

[2] M. Altun, M. D. Riedel, and C. Neuhauser. Nanoscale digital computation through percolation. In *Design Automation Conference*, pages 615–616, 2009.

[3] J. C. Anderson, E. J. Clarke, A. P. Arkin, and C. A. Voigt. Environmentally controlled invasion of cancer cells by engineered bacteria. *Journal of Molecular Biology*, 355(4):619–627, 2006.

[4] Y. Cui and C. M. Lieber. Functional nanoscale electronic devices assembled using silicon nanowire building blocks. *Science*, 291(5505):851–853, 2001.

[5] A. DeHon. Nanowire-based programmable architectures. *ACM Journal on Emerging Technologies in Computing Systems*, 1(2):109–162, 2005.

[6] T. Eiter, K. Makino, and G. Gottlob. Computational aspects of monotone dualization: A brief survey. *Discrete Applied Mathematics*, 156(11):1952–2005, 2008.

[7] P. Érdi and J. Tóth. *Mathematical Models of Chemical Reactions: Theory and Applications of Deterministic and Stochastic Models*. Manchester University Press, 1989.

[8] M. M. Eshaghian-Wilner, A. Khitun, S. Navab, and K. Wang. A nano-scale reconfigurable mesh with spin waves. In *International Conference on Computing Frontiers*, pages 5–9, 2006.

[9] M. L. Fredman and L. Khachiyan. On the complexity of dualization of monotone disjunctive normal forms. *Journal of Algorithms*, 21(3):618–628, 1996.

[10] D. Gillespie. Exact stochastic simulation of coupled chemical reactions. *Journal of Physical Chemistry*, 81(25):2340–2361, 1977.

[11] F. Horn and R. Jackson. General mass action kinetics. *Archive for Rational Mechanics and Analysis*, 47:81–116, 1972.

[12] Y. Huang, X. Duan, Y. Cui, L. J. Lauhon, K. Kim, and C. M. Lieber. Logic gates and computation from assembled nanowire building blocks. *Science*, 294(5545):1313–1317, 2001.

[13] T. Ibaraki and T. Kameda. A theory of coteries: Mutual exclusion in distributed systems. *IEEE Transactions on Parallel and Distributed Systems*, 4(7):779–794, 1993.

[14] H. Jiang, A. P. Kharam, M. D. Riedel, and K. K. Parhi. Digital signal processing with biomolecular reactions. In *IEEE International Conference on Computer-Aided Design*, pages 417–424, 2010.

[15] H. Jiang, M. D. Riedel, and K. K. Parhi. Synchronous sequential computation with molecular reactions. In *Design Automation Conference*, 2011.

[16] A. Kharam, H. Jiang, M. D. Riedel, and K. Parhi. Binary counting with chemical reactions. In *Pacific Symposium on Biocomputing*, 2011.

[17] A. Khitun, M. Bao, and K. L. Wang. Spin wave magnetic nanofabric: A new approach to spin-based logic circuitry. *IEEE Transactions on Magnetics*, 44(9):2141–2152, 2008.

[18] J.-J. Liou, K.-T. Cheng, S. Kundu, and A. Krstic. Fast statistical timing analysis by probabilistic event propagation. In *Design Automation Conference*, pages 661–666, 2001.

[19] Y. Luo, C. P. Collier, J. O. Jeppesen, K. A. Nielsen, E. Delonna, G. Ho, J. Perkins, H. Tseng, T. Yamamoto, J. F. Stoddart, and J. R. Heath. Two-dimensional molecular electronics circuits. *ChemPhysChem*, 3(6):519–525, 2002.

[20] R. Marculescu, D. Marculescu, and M. Pedram. Logic level power estimation considering spatiotemporal correlations. In *International Conference on Computer-Aided Design*, pages 294–299, 1994.

[21] K. P. Parker and E. J. McCluskey. Probabilistic treatment of general combinational networks. *IEEE Transactions on Computers*, 24(6):668–670, 1975.

[22] W. Qian, J. Backes, and M. D. Riedel. The synthesis of stochastic circuits for nanoscale computation. *International Journal of Nanotechnology and Molecular Computation*, 1(4):39–57, 2010.

[23] W. Qian, X. Li, M. D. Riedel, K. Bazargan, and D. J. Lilja. An architecture for fault-tolerant computation with stochastic logic. *IEEE Transactions on Computers*, 60(1):93–105, 2011.

[24] W. Qian and M. D. Riedel. The synthesis of robust polynomial arithmetic with stochastic logic. In *Design Automation Conference*, pages 648–653, 2008.

[25] W. Qian, M. D. Riedel, K. Barzagan, and D. Lilja. The synthesis of combinational logic to generate probabilities. In *International Conference on Computer-Aided Design*, pages 367–374, 2009.

[26] W. Qian, M. D. Riedel, and I. Rosenberg. Uniform approximation and Bernstein polynomials with coefficients in the unit interval. *European Journal of Combinatorics*, 32(3):448–463, 2011.

[27] W. Qian, M. D. Riedel, H. Zhou, and J. Bruck. Transforming probabilities with combinational logic. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2011 (to appear).

[28] D. Ro, E. Paradise, M. Ouellet, K. Fisher, K. Newman, J. Ndungu, K. Ho, R. Eachus, T. Ham, M. Chang, S. Withers, Y. Shiba, R. Sarpong, and J. Keasling. Production of the antimalarial drug precursor artemisinic acid in engineered yeast. *Nature*, 440:940–943, 2006.

[29] J. Savir, G. Ditlow, and P. H. Bardell. Random pattern testability. *IEEE Transactions on Computers*, 33(1):79–90, 1984.

[30] P. Senum and M. D. Riedel. Rate-independent biochemical computational modules. In *Proceedings of the Pacific Symposium on Biocomputing*, 2011.

[31] P. Senum and M. D. Riedel. Rate-independent constructs for chemical computation. *PLoS ONE*, 6(6), 2011.

[32] C. E. Shannon. A symbolic analysis of relay and switching circuits. *Transactions of the AIEE*, 57(12):713–723, 1938.

[33] A. Shea, B. Fett, M. D. Riedel, and K. Parhi. Writing and compiling code into biochemistry. In *Proceedings of the Pacific Symposium on Biocomputing*, pages 456–464, 2010.

[34] D. Soloveichik, G. Seelig, and E. Winfree. DNA as a universal substrate for chemical kinetics. *Proceedings of the National Academy of Sciences*, 107(12):5393–5398, 2010.

[35] S. Strogatz. *Nonlinear Dynamics and Chaos with Applications to Physics, Biology, Chemistry, and Engineering.* Perseus Books, 1994.

[36] S. Venkataramana, R. M. Dirks, C. T. Ueda, and N. A. Pierce. Selective cell death mediated by small conditional RNAs. *Proceedings of the National Academy of Sciences*, 2010 (in press).

[37] D. M. Widmaier, D. Tullman-Ercek, E. A. Mirsky, R. Hill, S. Govindarajan, J. Minshull, and C. A. Voigt. Engineering the *Salmonella* type III secretion system to export spider silk monomers. *Molecular Systems Biology*, 5(309):1–9, 2009.

[38] B. Yurke, A. J. Turberfield, A. P. Mills, Jr., F. C. Simmel, and J. Neumann. A DNA-fuelled molecular machine made of DNA. *Nature*, 406:605–608, 2000.