FAST: A Framework for Simulation and Analysis of Large-Scale Protein-Silicon Biosensor Circuits

Ming Gu, *Member, IEEE*, and Shantanu Chakrabartty, *Senior Member, IEEE*

***Abstract*—This paper presents a computer aided design (CAD) framework for verification and reliability analysis of protein-silicon hybrid circuits used in biosensors. It is envisioned that similar to integrated circuit (IC) CAD design tools, the proposed framework will be useful for system level optimization of biosensors and for discovery of new sensing modalities without resorting to laborious fabrication and experimental procedures. The framework referred to as FAST analyzes protein-based circuits by solving inverse problems involving stochastic functional elements that admit non-linear relationships between different circuit variables. In this regard, FAST uses a factor-graph netlist as a user interface and solving the inverse problem entails passing messages/signals between the internal nodes of the netlist. Stochastic analysis techniques like density evolution are used to understand the dynamics of the circuit and estimate the reliability of the solution. As an example, we present a complete design flow using FAST for synthesis, analysis and verification of our previously reported conductometric immunoassay that uses antibody-based circuits to implement forward error-correction (FEC).**

***Index Terms*—Biomolecular circuit, biosensors, computer-aided design, factor-graphs, inverse problems, message passing, simulation.**

I. Introduction

DESIGN of reliable biosensors requires understanding, modeling and characterization of fundamental noise, stochastic interactions between proteins and device artifacts. In a theoretical study reported in [1], [2], it was shown that the signals acquired from biosensors could potentially exhibit large variability due to random interactions between biomolecules or due to the noise at the interface of the transducer. While the effects of variability could potentially be alleviated by improving experimental protocols and device fabrication process, in [3] we had proposed using a forward error-correcting (FEC) approach to improve the reliability of biosensors. The FEC biosensor, whose architecture is shown in Fig. 1, uses protein-based reactive circuits (using antibodies, aptamers, or DNA) in conjunction with a transducer which converts the binding of an analyte with the protein into a measurable optical or electrical signal. A biomolecular encoder synthetically introduces redundancy into the protein-protein interaction before the signal

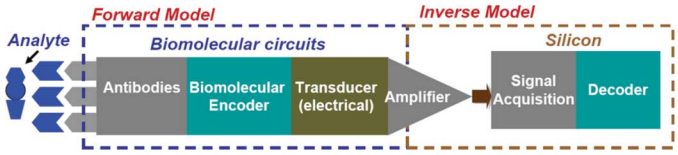


Fig. 1. System level architecture of an FEC biosensor interface which can be analyzed using FAST.

generated by the transducer is read out [3]. Acquisition and decoding of the sensor signal is performed using silicon (CMOS) circuits and the final result is a score which is proportional to the probability of the target analyte being present in the sample. In [3] we reported examples of protein (antibody) based combinatorial circuits and in [4] we experimentally demonstrated the feasibility of a small-scale biomolecular encoder using lateral-flow immunoassays. However, these preliminary experiments indicated that the full potential of an FEC biosensor can be only be realized using large-scale biomolecular encoders which integrates millions of protein-based circuits. This would require a simulation framework that could be used to model, analyze and predict the reliability of large-scale biomolecular encoders without resorting to laborious and expensive experimental procedures. In this paper, we present a computer aided design (CAD) framework called FAST (**F**actor-graph based **A**nalysis of **S**tochastic circui**T**s), which can be used for design, synthesis and verification of large-scale hybrid protein-silicon circuits.

A FAST based design-flow of FEC biosensors is summarized by the chart in Fig. 2. First, simple protein circuits are prototyped and their responses are experimentally measured. The experimental data is then used for generating a library of behavioral models, equivalent circuit models and channel (or noise) models. These models are instantiated and imported when the user specifies a topology comprising of the basic protein-circuits. During analysis, FAST allows additional constraints to be incorporated into the design, which includes limitations on the size of the biosensor substrate, cross-talk due to the substrate, analyte propagation and transducer artifacts. FAST then generates a factor-graph netlist that captures the probabilistic interdependencies between different circuit elements. Inference or estimation of probability distributions of target variables are obtained using Monte-Carlo simulations and the dynamics of the factor-graph circuit is understood using a density-evolution analysis. The outcome of the simulations are detection error-rate (DER) curves which can then be used to determine the reliability of the circuit. As a final step in the design-flow, the encoder circuit is fabricated to validate the simulated reliability metrics or some important detection property as predicted by FAST.

Manuscript received May 17, 2012; revised August 17, 2012; accepted September 28, 2012. This work was supported by a research grant from the National Science Foundation (CCF:1117186). This paper was recommended by Associate Editor K. C. Cheung.

The authors are with the Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI 48824 USA (e-mail: shantanu@egr.msu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TBCAS.2012.2222403

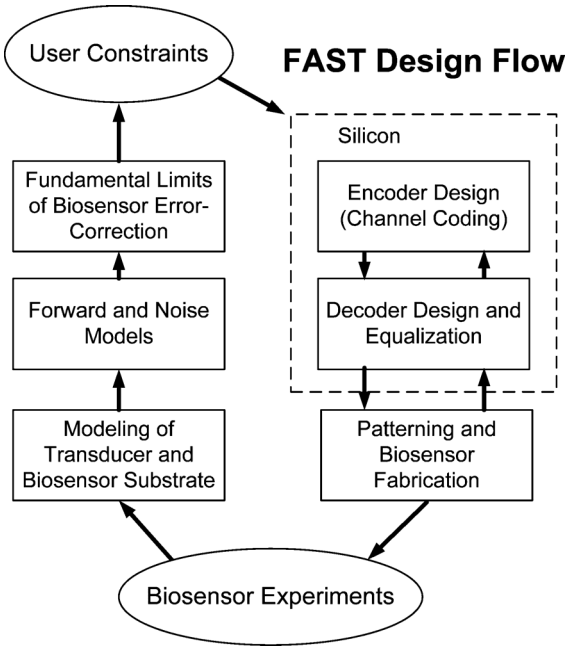


Fig. 2. FAST design flow.

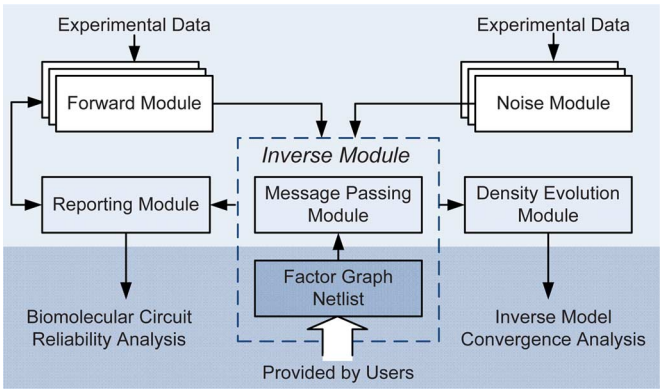


Fig. 3. Software architecture of FAST.

The software architecture of FAST is shown in Fig. 3 and is comprised of six key modules.

- **Forward module:** This module uses experimental data to generate behavioral models of the basic protein circuits. These behavioral models are then used to emulate the response of the protein circuits during simulation.
- **Noise module:** This module uses experimental data to estimate the noise parameters in the behavioral models of protein circuits. In FAST, two types of noise sources are modeled: (a) systematic noise introduced by sensor artifacts, variations in experimental protocols and experimental conditions; and (b) random noise due to stochastic interactions between biomolecules and due to measurement noise. The noise models in conjunction with the behavioral models are used for large-scale Monte-Carlo simulations.
- **Factor-graph module:** In its general form, a factor-graph graphically captures the interdependencies between different independent random variables. In FAST, this module provides an interface to a user to specify the topology of a biomolecular encoder. The module then generates a netlist similar to SPICE and presents it as an input to the FAST message-passing module.
- **Message passing module:** Message-passing is an iterative decoding algorithm that computes the probability distributions of random variables in a factor graph [5]. This module is the core computational unit of FAST and uses a distributed algorithm for generating inferences over the parameters of interest. The algorithm can be parallelized and hence can be scaled to achieve faster convergence.
- **Density evolution module:** Density evolution (DE) provides a way to study the convergence of message passing algorithm by tracking distributions of messages that are propagated between different nodes in a factor graph. The evolution of this distribution and its steady-state could then be used to determine the dynamics and the performance of the circuit.
- **Reporting module:** This module is used by FAST to generate multi-dimensional detection error-rate (DER) data to determine the reliability of detection of the circuit or to investigate novel detection modalities through post-processing.

We now describe in details, the theory and the underlying algorithms used for implementing some of the FAST modules.

II. Forward and Noise Module

The behavioral model of a biosensor depends on: (a) the operational physics of the protein-silicon interface; (b) on the type of biomolecules used for recognizing the analyte of interest; and (c) on the type of transducer used for converting the recognition event into a measurable signal. In literature, numerous modeling studies have been reported that capture the response of different types of biosensors at a system level [19]–[22]. Some of the examples include acoustic-wave biosensors [23], amperometric biosensors [24], silicon-based thermal biosensors [25], and cell-based biosensor [26]. Without the loss of generality, in this paper, we will focus on behavioral models that are applicable to affinity-based biosensors, or biosensors that use antibodies, aptamers or DNA as recognition elements. Typically, in an affinity-based biosensor the recognition elements are immobilized at specific locations on a substrate (usually made of silicon or porous material like nitrocellulose paper [3]). This is illustrated in Fig. 4 where the substrate is referred to as the reaction chamber. When the sample containing the analyte is applied to the biosensor, the analyte diffuses across the substrate and binds with its specific recognition element (specific antibodies in Fig. 4). Note that the analyte could potentially bind with the non-specific recognition element (shown with different colors in Fig. 4). However, the probability of non-specific binding is relatively low for a carefully designed immunoassay. The binding event is then converted into a measurable signal by the transducer (optical or electrical) and is used for detecting the presence of the analyte or for measuring the analyte concentration. At a microscopic level, the analyte diffusion in the biosensor reaction chamber can be modeled as a random-walk [1], [2] stochastic process, whose parameters are determined by the concentration gradient of the analyte, the ambient temperature, the transport properties of the substrate, and the aqueous medium

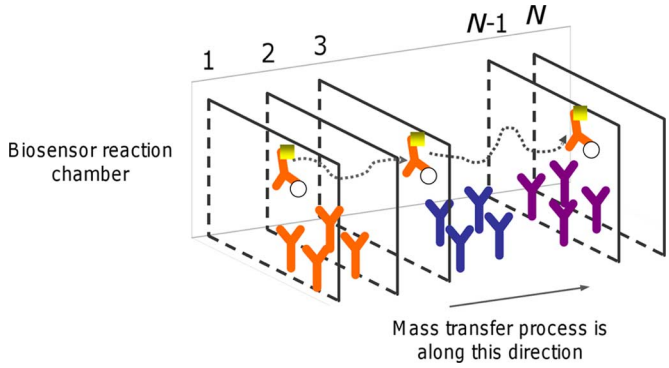


Fig. 4. Analyte motion, capturing, association and dissociation within a biosensor reaction chamber.

in which all the biomolecules reside. Similarly, the binding between the analyte and the recognition element is determined by the thermodynamics of the interaction process between the biomolecules [2]. This interaction could be modeled using a probabilistic model like a Markov chain, in which the states of the Markovian model signify whether the analyte is bounded to its specific recognition element or is unbounded and free to diffuse to another binding site (as shown in Fig. 4).

Both analyte diffusion and analyte binding microscopic models are useful for understanding the dynamical response of the biosensor and for understanding the sources of variability and fundamental noise. However, this dynamics is also strongly dependent on the type of proteins and the type of transducers, which requires a fairly large and complex library of behavioral models. While the long-term goal is to incorporate different types of dynamical models in FAST, for the first version, our focus has been on modeling the steady-state, input-output response of a generic affinity-based biosensor. We use a simple parametric model whose complexity is only determined by the concentration of the analytes being detected. The model is general enough to be applied to different types of affinity-based biosensor, as will be clear from the following analysis.

Let X_A denote the concentration of the analyte A and let G denote the magnitude of the output signal (optical or electrical) generated by the transducer. Then, the change in magnitude of the output signal ΔG would be directly proportional to the change in analyte concentration ΔX_A and is expressed as

$$\Delta G = k\mathcal{N}(X_A)\Delta X_A \tag{1}$$

where k is a proportionality constant independent of the analyte concentration modeling the sensitivity of the biosensor. $\mathcal{N}(\cdot)$ is a function that models the saturation response of the biosensor. Saturation arises due to the lack of availability of the analyte binding sites (or due to quenching of the optical transducers) as more analytes hybridize or react with their specific biomolecules (see Fig. 4). Even though different forms of $\mathcal{N}(\cdot)$ are possible, $\mathcal{N}$ should satisfy the following general properties:

$$\mathcal{N}(X) > 0 \tag{2}$$

$$\frac{d}{dX}\mathcal{N}(X) \leq 0 \tag{3}$$

$$\mathcal{N}(0) = C < \infty \tag{4}$$

$$\mathcal{N}(\infty) = 0. \tag{5}$$

Conditions (2) and (3) ensure that the change in magnitude reduces as more hybridization sites are used up. Conditions (4) and (5) specify the boundary conditions for $\mathcal{N}(\cdot)$. One possible choice of $\mathcal{N}(\cdot)$ could be

$$\mathcal{N}(X) = \frac{1}{X_{A0} + X} \tag{6}$$

where $X_{A0} > 0$ is a concentration parameter that determines the saturation characteristics under control conditions (when no analyte is present in the sample). Equation (1) could then be expressed in its differential form as

$$dG = k\frac{dX}{X_{A0} + X} \tag{7}$$

$$\int_{G_0}^{G} dG = k\int_{0}^{X_A} \frac{dX}{X_{A0} + X} \tag{8}$$

$$G(X_A) = G_0 + k\log\left(\frac{X_{A0} + X_A}{X_{A0}}\right). \tag{9}$$

Thus, the signal generated by an affinity-based biosensor can be expressed as a log-linear function of the analyte concentration.

For a multi-analyte biosensor, the substrate is immobilized with receptors specific to different analytes, as shown in Fig. 4. In this regard, the biosensor signal will be a function of the concentration corresponding to different analytes. For the sake of simplicity, we will consider only two analytes A and B. If X_A and X_B denote the concentration of the analytes A and B, then the gradient of the transducer signal with respect to the respective analyte concentrations can be expressed as

$$\frac{\partial G}{\partial X_A} = k_A\left(\frac{1}{X_{A0} + X_A}\right) + k_{AB}\left(\frac{1}{X_{AB0} + X_A + X_B}\right) \tag{10}$$

$$\frac{\partial G}{\partial X_B} = k_B\left(\frac{1}{X_{B0} + X_B}\right) + k_{AB}\left(\frac{1}{X_{AB0} + X_A + X_B}\right) \tag{11}$$

where k_A, k_B, X_{A0}, X_{B0} are model parameters corresponding to analytes A and B respectively. The parameters k_{AB}, X_{AB0} model the effect of joint hybridization of the analytes A and B on the output signal. The joint hybridization could be signified as non-specific binding between an analyte and receptor molecule non-specific to that analyte. In other cases, the joint hybridization could be synthetically introduced, as is the case in the FEC biosensor architecture. The solution of (10) and (11) is given by

$$\begin{aligned} G(X_A, X_B) = G_0 &+ k_A\log\left(\frac{X_{A0} + X_A}{X_{A0}}\right) \\ &+ k_B\log\left(\frac{X_{B0} + X_B}{X_{B0}}\right) \\ &+ k_{AB}\log\left(\frac{X_{AB0} + X_A + X_B}{X_{AB0}}\right). \end{aligned} \tag{12}$$

Equations (12) could in principle be generalized to more than two analytes and different types of logical circuits (for e.g.,

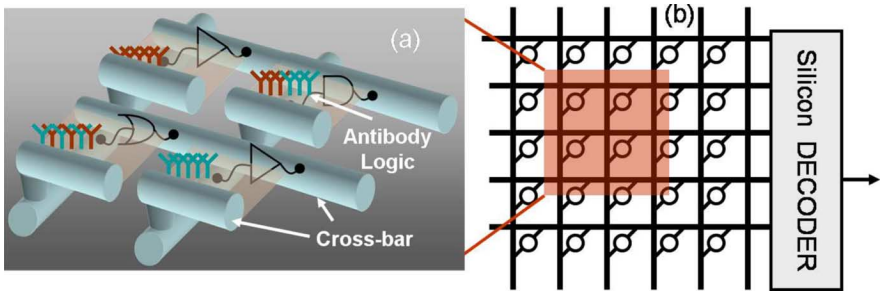


Fig. 5. Illustration of a cross-bar bio-silicon circuit topology that can be specified by the user.

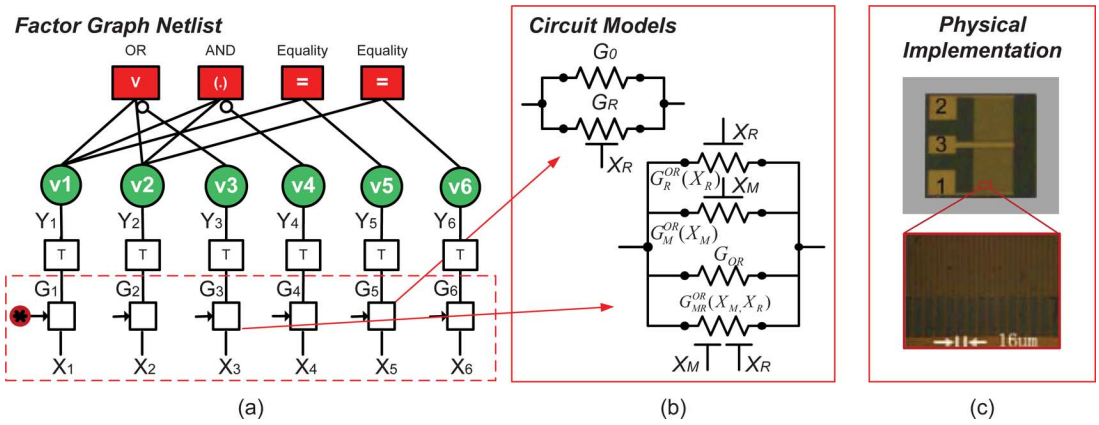


Fig. 6. The factor graph netlist of an example protein array, its equivalent circuit models and a micrograph of its physical implementation.

"AND" or "OR" gates). In the results section, we will demonstrate the use of these behavioral models to capture the functional response of different circuits constructed on an affinity-based biosensor.

The behavioral models can be augmented to include the effect of noise and variability in the biosensor signal. Like the behavioral models, noise models are also specific to the biosensor platforms and is dependent on the physics of: (a) reagent handling; (b) analyte propagation through the substrate; (c) stochastic interactions between the proteins; (d) noise at the interface of the transducer; and (e) noise in read-out instrumentation. While future versions of FAST will incorporate complex platform specific noise models, in the first version of FAST we have used a simple additive white Gaussian noise (AWGN) model which is parameterized by an amplitude parameter. This is a reasonable assumption since the physics of each biosensor processing is unique and hence independent of each other. Therefore, (9) can be modified to obtain an augmented biosensor model specific to analyte A as

$$G(X_A) = G_0 + k_A \log\left(1 + \frac{X_A}{X_{A0}}\right) + G_{An}. \qquad (13)$$

G_{An} is a zero mean Gaussian random variable that models the cumulative noise of the biosensor.

Similarly, the joint model given by the (12) can be augmented as

$$G(X_A, X_B) = G_0 + k_A \log\left(\frac{X_{A0} + X_A}{X_{A0}}\right) + k_B \log\left(\frac{X_{B0} + X_B}{X_{B0}}\right) + k_{AB} \log\left(\frac{X_{AB0} + X_A + X_B}{X_{AB0}}\right) + G_{ABn} \qquad (14)$$

where G_{ABn} denotes the corresponding additive noise parameter. As will be shown in the Section IV, all the parameters in the (13) and (14) including G_n and G_{ABn} are estimated for specific logic circuits using log-linear regression methods.

III. Inverse Module and Density Evolution Module

Once a library of behavioral models (consisting of model parameters) has been created, the user can specify different topologies of protein-silicon circuits. A cross-bar topology is shown in Fig. 5 and consists of antibody based logic circuits which are spatially patterned at cross-bar points (intersection between the horizontal and vertical lines). Each cross-bar point (or the logic circuit) could be individually addressed and the change in conductance could be measured using a peripheral silicon decoder. For simulation purposes FAST models the cross-bar topology as an equivalent factor-graph which captures the logical dependencies between the antibody-based logic circuits, as shown in Fig. 6.

A. Fast Probabilistic Inverse Problem

The FAST simulation engine then uses the factor-graph to solve a probabilistic inverse problem whose objective is to generate inferences over circuit parameters given noisy observations. For a biomolecular circuit, the noisy observations corre-

spond to the signal generated by the protein based circuit (referred to as the forward model as shown in Fig. 1) and is given by $G_i, i = 1,\ldots,N$, where N is the number of observations. The simulation engine then produces a posteriori probability estimates $(P(X_k \,|\, G_1,\ldots,G_N))$ of the parameters of interest (X_k) given the noisy observations $(G_i, i = 1,\ldots,N)$. For example, in a biomolecular circuit $X_k \in \{0,1\}$ could be boolean variables indicating the presence or absence of pathogens in the given sample. Then, a straightforward application of the Bayesian rule leads to

$$\begin{aligned} P(X_k \,|\, G_1,\ldots,G_N) &= \propto \sum_{\sim X_k} P(X_1,\ldots,X_N,G_1,\ldots,G_N) \\ &= \sum_{\sim X_k} \prod_{j=1}^{N} P(G_j \,|\, X_k) P(X_1,\ldots,X_N) \\ &= \sum_{\sim X_k} f(X_1,\ldots,X_N) \prod_{j=1}^{N} P(G_j \,|\, X_k) \end{aligned} \tag{15}$$

where we have assumed that the noisy observations conditioned on the parameters of interest are independent of each other. Equation (15) illustrates the functional architecture of FAST, where $P(G_j \,|\, X_k)$ is dependent on individual biomolecular circuit whose parameters have been pre-computed using in-laboratory experiments. These parameters could then be imported into FAST and stored in the software repository. The function $f(X_1,\ldots,X_N)$ in (15) represents the interconnection between different biomolecular circuits and will be referred to as the factor graph netlist. The factor-graph netlist consists of two kinds of nodes: (a) variable nodes which represent the circuit parameters to be estimated and (b) check nodes which capture the functional relationship between different circuit parameters (connected to the check nodes using "edges"). For example, in Fig. 6 the circuit parameters are functionally related to each other through different logic functions ("AND", "OR" and "Equality"). The factor-graph netlist also consists of transducer nodes (denoted by "T") which integrates the circuit models from the forward and noise modules (see Fig. 6). For example, an OR logic model will be used to generate the posterior probability $P(A + B \,|\, G_i)$ that either one of the two analytes A or B is present given the measurement G_i. All the posterior probabilities are combined together using a factor-graph to solve the probabilistic inverse problem or to compute the probability of presence of target pathogens $P(A \,|\, G_1, G_2,.,.), P(B \,|\, G_1, G_2,\ldots)$ given all the measured conductances $G_1,\ldots,G_N$. The inference according to (15) is determined using a message passing algorithm, where each node in the netlist propagates messages along the edge to each of its immediate neighbors. These messages take the form of probability estimates that the node to which the message is being sent to is in the state 0 or 1. The messages are computed using two basic and distributed operations [7]: (a) marginalization which involves summation over local joint distributions; and (b) multiplication of local marginal probabilities. The messages are propagated back and forth between the nodes for a pre-determined number of iterations before a decision on the variables $X_1,\ldots,X_k$ are made. The distributed and local computations avoids unnecessary summation or multiplication over irrelevant internal nodes, thus facilitating inference over large-scale biomolecular circuits.

B. Density Evolution Module

Density evolution provides a way to study the factor-graph decoding algorithm by analyzing the evolution of message densities on a hypothetical infinite tree-like with similar local topology as the factor-graph of interest [12]. Traditionally, density evolution is used to observe the asymptotic message distributions and hence estimates a minimum threshold parameter (e.g., noise-variance of a Gaussian channel) that ensures reliability (error-rate) performance less than some specific tolerance. The density evolution procedure computes the probability density of messages during an iterative decoding algorithm based on a recursive estimation procedure that can be evaluated analytically or numerically using Monte-Carlo simulations [10], [11]. In this regard, it can be utilized to observe the convergence of code. Here we briefly describe density evolution using the example factor-graph shown in Fig. 6(a). For details the readers are referred to [10]. If y denotes the random variable corresponding to the messages propagated from the check to the variable nodes of the netlist, density evolution tracks the cumulative distribution P_e^{ℓ} with respect to iteration time ℓ according to

$$P_e^{\ell} = \int_{-\infty}^{0} P_{\ell}(x)\, dx \tag{16}$$

where $P_{\ell}(x)$ is the probability density function (pdf) of messages x. Under the assumption that no antigens are present in the analyte, a convergent message passing algorithm over the factor graph netlist will yield P_e^{ℓ} which approaches zero as $\ell \to \infty$. Monitoring P_e^{ℓ} will therefore be indicative of the dynamic behavior of the factor-graph netlist.

IV. Example FAST Design Flow

We now demonstrate an example FAST design flow (shown in Fig. 2) for our previously reported conductometric immunoassay. The immunoassay uses pathogen specific antibodies for bio-recognition and uses silver-enhanced gold-nanoparticles as transducers [6], [13]. The immunoassay can be integrated directly with CMOS circuits and the silver-enhancement based amplification technique can achieve detection limits that are comparable to that of optical methods [13], [14]. For the sake of completeness we briefly describe the principle operation of the immunoassay and is summarized in Fig. 7. A single pathogen biosensor (referred to as EQ or equality gate in Fig. 7) consists of a layer of antibodies (specific to the target pathogen) which is immobilized between two conductive electrodes. When the sample containing the target antigens (An) is applied to the device, the antigens bind with its specific antibodies. Then, when secondary antibodies conjugated with gold(Au) nanoparticles is applied a sandwich (Ab-An-Ab-Au) structure is formed between the electrodes, as shown in Fig. 7. However, the gold-nanoparticles by itself is insufficient to bridge the gap between the electrodes. Therefore, a biomolecular amplification procedure called "silver-enhancement" [15]

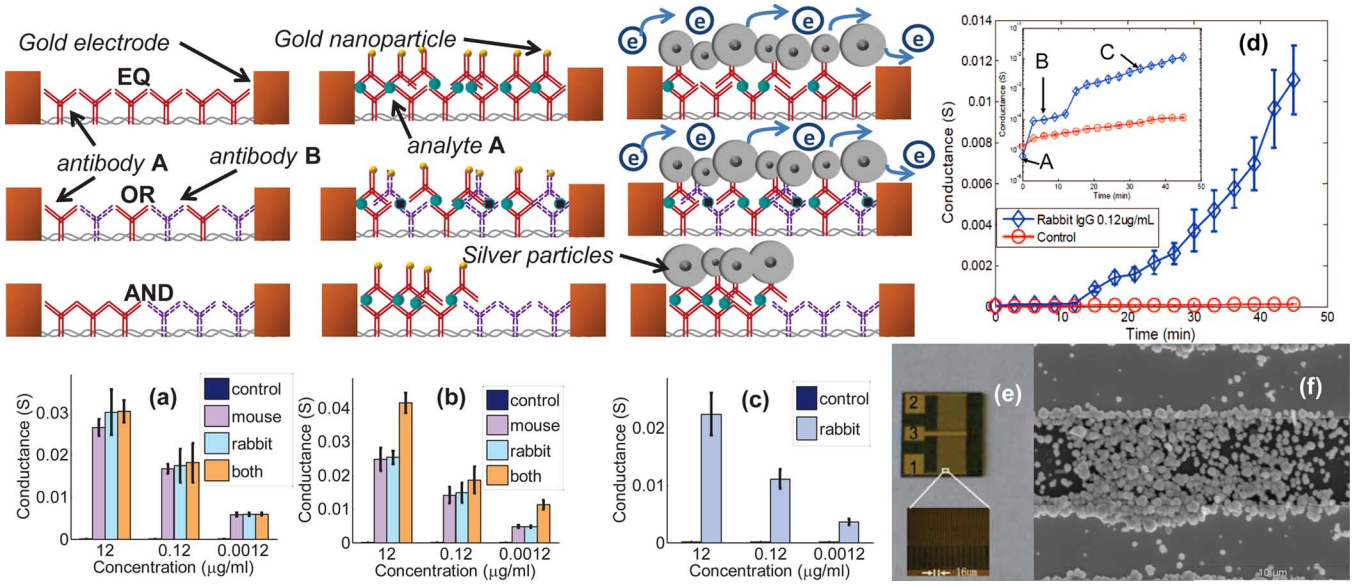


Fig. 7. Principle of operation and experimental results corresponding to our previously reported immunoassay (adapted from [6], [13]). The experimental data has been used to demonstrate the complete FAST design flow.

is used to amplify the conductance signal. Note that different biosensors use different methods of biomolecular amplification and hence the behavioral model has to be appropriately modified.

For the model immunoassay, the amplication procedure is shown in Fig. 7 where the sandwich (Ab-An-Ab-Au) is subjected to a solution of silver and hydroquinone (photographic developing solution). The gold nano-particle surface act as a catalyst and reduce silver ions into metallic silver in the presence of a reducing agent (for e.g., hydroquinone). The reduced silver deposits on the gold surface, enlarging the size of the gold nano-particle as shown in Fig. 7. As the size of the silver islands grow, they offer a low resistive path for the electrons to hop between the electrodes which is registered as an increase in conductance across the electrodes. With the increase in enhancement time, the growth of silver-enhanced particles completely bridges the gap between the electrodes (shown in Fig. 7) and facilitates direct transport of electrons between the electrodes. The silver-enhancement procedure, in principle, can be viewed as a biomolecular integrator which effectively reduces the effect of noise, however, at the expense of slower response time. The change in conductance across the bridge serves as a measure of analyte concentration (magnitude of the input signal).

Fig. 7(e) shows the experimental prototypes which were fabricated [6], [13]. The inter-digitated interface consists of three electrodes each of length 5 mm, and with 500 fingers with a spacing between the figures being 5 μm each. Fig. 7(f) shows the SEM of the biomolecular device post silver-enhancement where the nanoparticles completely bridge the gap between the electrodes. In [6], [13], [4] rabbit IgG was used as one of the target analyte and biosensor was characterized by measuring the conductance across the electrodes. The result shown in Fig. 7(d) revealed three distinct regions of operation (labeled A,B and C) during the enhancement procedure. Region A represents the conductance which is measured without the application of the silver enhancement solution. Region B represents the conductance which is measured when the silver macroparticles have not completed the bridge between the electrodes. Region C represents the conductance when the bridge has been successfully formed between the electrodes.

In [6], [13] we reported a procedure to fabricate basic logic circuits ("AND" and "OR") by pico-liter dispensing of antibody patterns on different spatial locations of the silicon substrate. The basic architecture and operational principle of the logic circuits is also illustrated in Fig. 7. In the case of "OR" gate, when either one of the proteins is present (one of the input is valid), the silver-enhanced gold nanoparticles will ultimately bridge the two electrodes (assuming the antibody density is large enough). While in the case of the "AND" gate, the condition of completing the conductive bridge between the two electrodes will be determined by the presence of both the proteins in the sample. When either one of the pathogen exists, the bridge will be partially formed which results in low measured conductance. Fig. 7(a) and (b) show preliminary results where the the conductances are measured for different concentrations of rabbit and mouse IgG (12 μg/ml -0.12 ng/ml) and under different logic conditions. Control experiments for all the experiments were obtained using bovine IgG. It can be seen from Fig. 7(a) and (b) that under different concentration levels of analytes the response of the logic function remains consistent, however, the magnitude of the measured conductance scales log-linearly with the IgG concentration.

Once the experimental data corresponding to different logic circuits is presented as input to the FAST forward and noise module, multi-variable log-linear regression is performed to estimate parameters of the behavioral and noise models given by (13) and (14). Note that each combination of the analyte will give rise it a specific set of model parameters. These parameters are stored in a model repository (implemented as an indexed database) in FAST. estimated using a multi-variable log-linear regression on the experimental data. The regression error is used to estimate the noise parameter (variance) and captures the cu-

TABLE I
BEHAVIORAL MODEL PARAMETERS

Parameter	Description	Value
G_{M0}	Control conductance (Mouse IgG)	160 μS
G_{R0}	Control conductance (Rabbit IgG)	120 μS
X_{M0}	Detection limit (Mouse IgG)	190 μg/ml
X_{R0}	Detection limit (Rabbit IgG)	195 μg/ml
k_M	Sensitivity (Mouse IgG)	$4.4 \cdot 10^{-3}$ μS
k_R	Sensitivity (Rabbit IgG)	$4.4 \cdot 10^{-3}$ μS
G^{OR}	Control conductance (Mouse + Rabbit IgG OR gate)	120 μS
k_M^{OR}	Sensitivity (Mouse IgG in OR gate)	$-3 \cdot 10^{-4}$ μS
k_R^{OR}	Sensitivity (Rabbit IgG in OR gate)	$1 \cdot 10^{-3}$ μS
k_{MR}^{OR}	Sensitivity (Mouse + Rabbit IgG in OR gate)	$5.3 \cdot 10^{-3}$ μS
X_{M0}^{OR}	Detection limit (Mouse IgG in OR gate)	57.4 μg/ml
X_{R0}^{OR}	Detection limit (Rabbit IgG in OR gate)	1000 μg/ml
X_{MR0}^{OR}	Detection limit (Mouse + Rabbit IgG in OR gate)	63 μg/ml
G^{AND}	Control conductance (Mouse + Rabbit IgG in AND gate)	110 μS
k_M^{AND}	Sensitivity (Mouse IgG in AND gate)	$2.4 \cdot 10^{-3}$ μS
k_R^{AND}	Sensitivity (Rabbit IgG in AND gate)	$2.7 \cdot 10^{-3}$ μS
k_{MR}^{AND}	Sensitivity (Mouse + Rabbit IgG in AND gate)	$2.6 \cdot 10^{-3}$ μS
X_{M0}^{AND}	Detection limit (Mouse IgG in AND gate)	4.6 μg/ml
X_{R0}^{AND}	Detection limit (Rabbit IgG in AND gate)	24 μg/ml
X_{MR0}^{AND}	Detection limit (Mouse + Rabbit IgG in AND gate)	1200 μg/ml

mulative effect of different noise sources affecting the output of the immunoassay. Table I summarizes the repository of model parameters estimated using experimental data obtained from the immunoassay logic circuits shown in Fig. 5. Note that the behavioral models could also be represented by its equivalent circuits as shown in Fig. 6(b). Noise variance was adjusted based on the regression error obtained when the linear models in Table I were built.

Next, FAST is used for simulating a 30×30 cross-bar circuit as shown in Fig. 5, where each cross-bar point instantiates one of the antibody logic circuits shown in Table I. The first topology is referred to as the "repetition" topology, each protein element is patterned multiple times on different cross-bar locations. The second topology is referred to as the "asymmetric" topology where a logical combination of the protein circuit (AND, OR and Equality) are used (see Fig. 1) at different cross-bar locations. For comparison purposes an "uncoded" topology is also simulated which only uses a single protein circuit (without encoding). In the simulation experiment, different concentration of analyte (rabbit IgG) is applied to the circuit and the objective is to determine the inference on the presence or absence of analyte in presence of random noise. Here the biosensor noise was modeled as a zero mean additive white Gaussian noise (AWGN).

Once the simulation starts, the density evolution module is used to trace the mean message probability density functions (pdfs) as shown in Fig. 8. The figure shows the pdfs of the messages propagated from variable-nodes to check-nodes are shown for different message-passing iterations. Note the number of message-passing iterations can be set by the user as one of several simulation parameters.

The evolution of the pdfs illustrates that starting from an equal likelihood of pathogens being "detected" or "not detected," the pdf curves shift to the right. This implies that the biomolecular circuit converges stochastically towards a state where detecting the pathogens is significantly higher than the starting state. This verifies the correct dynamical response of the biomolecular circuit.

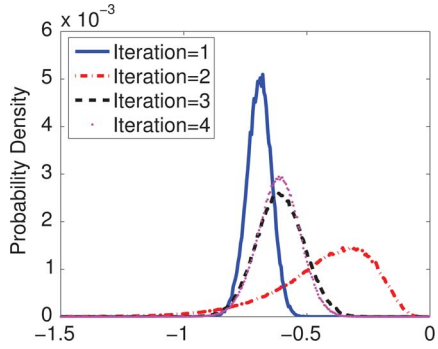


Fig. 8. The pdf of messages as reported by the FAST DE module.

At the end of the FAST simulation, the reporting module produces the detection error rate (DER) which summarizes the likelihood of incorrect detection for different concentrations of the analyte. Fig. 9 compares the DER corresponding to the three biomolecular circuit topologies for different concentrations of rabbit IgG. As expected, the FAST simulation shows that as the concentration of the pathogen increases the DER decreases for all the three topologies. However, the Fig. 9 also shows that a simple "asymmetric" topology outperforms the "repetition" topology by orders of magnitude in terms of DER. This result indicates that patterning of protein based circuit into different logical functions could provide significant reliability benefits especially since the "repetition" topology is most commonly used in state-of-the-art microarrays. Of-course, this simulation result needs to be verified using in-laboratory experiments where the optimized "asymmetric" biomolecular circuit can now be prototyped in silicon.

The last step in the FAST design flow (shown in Fig. 2) is to verify some key attribute of the DER simulations. However, verifying the DER of a 10^6 size protein circuit is experimentally cumbersome and laborious. Therefore, the benefits of the encoder topology will be verified indirectly based on the DER trends as summarized in Table II for a repetition encoder and

TABLE II
DER OBTAINED USING FAST FOR DIFFERENT CONCENTRATION OF RABBIT AND MOUSE IGG (CONCENTRATION UNIT: G/ML): SHOWING THAT FOR A REPETITION ENCODER AN INCREASE IN MOUSE IGG CONCENTRATION DOES NOT AFFECT THE DETECTION RELIABILITY OF RABBIT IGG

Mouse IgG \ Rabbit IgG	2×10^{-10}	6×10^{-10}	1.2×10^{-9}	1.2×10^{-8}	1.2×10^{-7}	1.2×10^{-6}	1.2×10^{-5}
2×10^{-10}	0.509	0.419	0.354	0.224	0.102	0.050	0.025
6×10^{-10}	0.501	0.422	0.380	0.216	0.119	0.051	0.020
1.2×10^{-9}	0.498	0.421	0.361	0.221	0.125	0.054	0.018
1.2×10^{-8}	0.503	0.415	0.371	0.222	0.109	0.052	0.019
1.2×10^{-7}	0.498	0.421	0.366	0.219	0.119	0.053	0.020
1.2×10^{-6}	0.502	0.414	0.368	0.221	0.115	0.051	0.020
1.2×10^{-5}	0.497	0.416	0.368	0.221	0.116	0.053	0.020

TABLE III
DER OBTAINED USING FAST FOR DIFFERENT CONCENTRATION OF RABBIT AND MOUSE IGG (CONCENTRATION UNIT: G/ML): SHOWING THAT FOR ASYMMETRIC ENCODER, PRESENCE OF HIGHER CONCENTRATION OF MOUSE IGG ENHANCES THE DETECTION RELIABILITY OF RABBIT IGG

Mouse IgG \ Rabbit IgG	2×10^{-10}	6×10^{-10}	1.2×10^{-9}	1.2×10^{-8}	1.2×10^{-7}	1.2×10^{-6}	1.2×10^{-5}
2×10^{-10}	0.245	0.181	0.145	0.055	0.016	0.004	0.0007
6×10^{-10}	0.208	0.155	0.124	0.049	0.014	0.003	0.0005
1.2×10^{-9}	0.188	0.140	0.119	0.044	0.013	0.003	0.0005
1.2×10^{-8}	0.129	0.095	0.075	0.033	0.009	0.002	0.0003
1.2×10^{-7}	0.079	0.056	0.044	0.019	0.006	0.001	0.0002
1.2×10^{-6}	0.045	0.030	0.024	0.009	0.003	0.0008	0.0001
1.2×10^{-5}	0.024	0.016	0.012	0.004	0.001	0.0004	7.7×10^{-5}

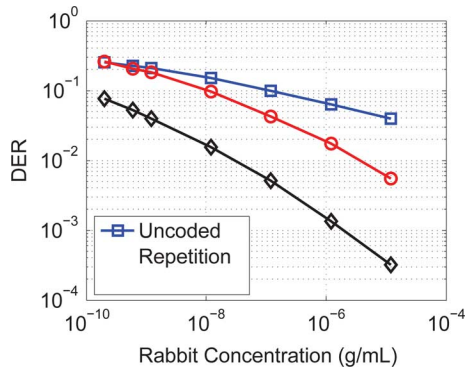


Fig. 9. The DER curves corresponding to "uncoded," "repetition," and "asymmetric" topology of the biomolecular circuit for different rabbit IgG concentration levels when mouse IgG concentration was set to 1.2×10^{-5}g/mL.

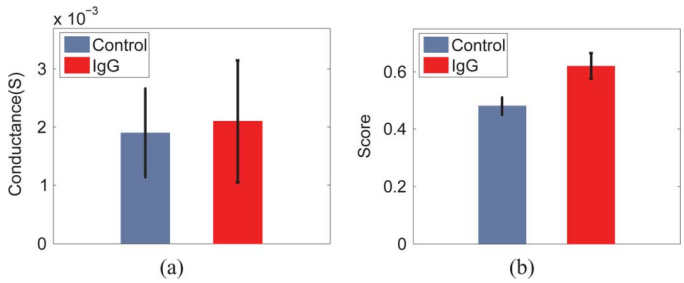


Fig. 10. Experimental verification (adapted from [6]) of responses predicted by FAST. (a) Results from a mouse IgG biochip when 100 pg/ml mouse IgG is used in presence of 10 μg/ml of rabbit IgG. (b) Results from the factor-graph decoder showing significant improvement in reliability of mouse IgG detection in the presence of background rabbit IgG.

in Table III for asymmetric encoder. Table II shows that as expected the DER of rabbit IgG decreases when the concentration of rabbit IgG in the sample increases. Also can be seen in Table II, the DER of rabbit IgG remains unaffected by the concentration of the mouse IgG. However the DER for the "asymmetric" code, as shown in Table III, shows that the increase in concentration of rabbit IgG increases the reliability of detection of the mouse IgG. This phenomena is a proof that the asymmetric encoder can indeed be used to design reliable immunoassays.

Our experiments reported in [6] provide the proof for the response of an FEC biosensor as predicted by FAST. In the experiment reported in [6], 100 pg/mL mouse IgG was added to the solution of 10 μg/mL rabbit IgG which served as the background interference. 10 μg/mL bovine IgG was used as the negative control experiment. For calibration purposes, a sample containing only 100 pg/mL mouse IgG was applied a biosensor specific to a single mouse IgG. Fig. 10(a) shows the result obtained from the single analyte biochip when a mixture of 100 pg/mL mouse IgG and 10 μg/mL rabbit IgG was applied. As expected, the results showed that anti-mouse IgG are non-specific to rabbit IgG. In the next experiment, the mixture with similar composition was applied to the six biochips and the measured conductances were process by the decoding algorithm. Fig. 10(b) shows the response obtained from the decoder and compares it with results obtained for the negative control experiments. Note that the output of the decoder is the normalized score indicating the presence or absence of mouse IgG. It can be clearly seen in Fig. 10(b) that not only has the magnitude of the scores have increased significantly compared to the control, the variance of the results have also decreased by orders of magnitude. Since there is no overlap between error bars, perfect detection of mouse IgG can be achieved in the presence of high background concentration of rabbit IgG. This experiment verified the reliability benefits of the asymmetric encoder as predicted by FAST simulations.

V. Conclusion

In this paper we presented a simulation tool, FAST, which can be used for analyzing protein-silicon hybrid circuits. The paper presented the software architecture of FAST and described its underlying factor-graph based decoding algorithm. The software provides sufficient flexibility to the user for defining different biomolecular circuit models and circuit topologies. Also, the software provides a detailed performance report which includes: (a) a reliability curve; and (b) a density evolution curve which can be used for system level debugging by observing the dynamic response of the circuit. The open-source version of FAST, written in C programming language is publicly available and can be downloaded from the web at *http://www.egr.msu.edu/aimlab/FAST*. The future work in this area includes: (a) populating the FAST database with different biomolecular circuit models corresponding to other emerging nano-biosensors; (b) parameterizing the circuit models to make it scalable with respect to the size and density of the nano-biosensor array; and (c) improving the speed of simulation by exploiting the convergence properties of message passing algorithms.

References

[1] A. Hassibi, H. Vikalo, and A. Hajimiri, "On noise processes and limits of performance in biosensors," *J. Appl. Phys.*, vol. 102, no. 1, p. 014909, 2007.

[2] A. Hassibi, S. Zahedi, R. Navid, R. W. Dutton, and T. H. Lee, "Biological shot-noise and quantum-limited signal-to-noise ratio in affinity-based biosensors," *J. Appl. Phys.*, vol. 97, no. 8, p. 084701, 2005.

[3] Y. Liu, S. Chakrabartty, and E. C. Alocilja, "Fundamental building blocks for molecular bio-wire based forward-error correcting biosensors," *Nanotechnol.*, no. 18, p. 4240172, 2007.

[4] Y. Liu and S. Chakrabartty, "Factor graph based biomolecular circuit analysis for designing forward error correcting biosensors," *IEEE Trans. Biomed. Circuits Syst.*, vol. 3, no. 3, pp. 150–159, Jun. 2009.

[5] W. E. Ryan, "An introduction to LDPC codes," in *CRC Handbook for Coding and Signal Processing for Recoding Systems*. Boca Raton, FL: CRC Press, 2004, ch. 36.

[6] Y. Liu, M. Gu, E. C. Alocilja, and S. Chakrabartty, "Co-detection: Ultra-reliable nanoparticle-based electrical detection of biomolecules in the presence of large background interference," *Biosens. Bioelectron.*, vol. 26, no. 3, pp. 1087–1092, Sep.. 2010.

[7] F. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[8] R. G. Gallager, "Low-density parity check codes," *IRE Trans. Inf. Theory*, vol. IT-8, no. 1, pp. 21–28, Jan. 1962.

[9] N. Wiberg, "Codes and decoding on general graphs," Ph.D. dissertation, Linkoping Univ., Linkoping, Sweden, 1996.

[10] T. Richardson, M. Shokrollahi, and R. Urbanke, "Design of capacity approaching irregular low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.

[11] T. Richardson and R. Urbanke, "The capacity of low-density parity check codes under message-passing decoding," *IEEE Trans. Inf. Theory*, vol. 47, pp. 599–618, 2001.

[12] D. J. C. MacKay, *Information Theory, Inference and Learning Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2003, ch. 6, pp. 557–572 [Online]. Available: http://www.inference.phy.cam.ac.uk/mackay/itila

[13] Y. Liu, D. Zhang, E. C. Alocilja, and S. Chakrabartty, "Biomolecules detection using a silver-enhanced gold nanoparticle-based biochip," *Nano Res. Lett.*, 2010, DOI: 10.1007/s11671- 010-9542-0.

[14] K. T. Liao and H. J. Huang, "Femtomolar immunoassay based on coupling gold nanoparticle enlargement with square wave stripping voltammetry," *Analytica Chimica Acta*, vol. 538, no. 1–2, pp. 159–164, 2005.

[15] S. Gupta, S. Huda, P. K. Kilpatrick, and O. D. Velev, "Characterization and optimization of gold nanoparticle-based silver-enhanced immunoassays," *Anal. Chem.*, vol. 79, no. 10, pp. 3810–3820, 2007.

[16] R. Almog, S. Zaitsev, O. Shtempluck, and E. Buks, "Signal amplification in a nanomechanical duffing resonator via stochastic resonance," *Appl. Phys. Lett.*, vol. 90, 2007.

[17] K. Wiesenfeld and F. Moss, "Stochastic resonance and the benefits of noise: From ice ages to crayfish and SQUIDs," *Nature*, vol. 373, pp. 33–36, 1995.

[18] D. F. Russell, L. A. Wilkens, and F. Moss, *Nature*, vol. 402, 1999.

[19] S. Jones, B. Jumarhon, S. McKee, and J. A. Scott, "A mathematical model of a biosensor," *J. Eng. Math.*, vol. 30, no. 3, pp. 321–337, 1996.

[20] C. Heitzinger, R. Kennell, G. Klimeck, N. Mauser, M. McLennan, and C. Ringhofer1, "Modeling and simulation of field-effect biosensors (BioFETs) and their deployment on the NanoHUB," in *J. Phys., Conf. Ser.*, 2008, vol. 107, p. 012004.

[21] F. Ivanauskas and R. Baronas, "Modelling an amperometric biosensor acting in a flowing liquid," *Int. J. Numer. Methods Fluids*, vol. 56, no. 8, pp. 1313–1319, 2007.

[22] Y. Liu and Q. Wang, "Mathematical and computational modeling of biosensors: Modeling for enzyme-substrate interaction and biomolecular interaction," in *Proc. 2nd WSEAS Int. Conf. Computational Chemistry Table of Contents, Recent Advances In Biology and Biomedicine*, 2008, pp. 38–42.

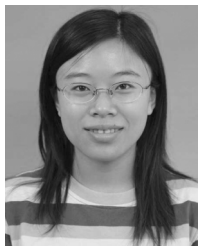
[23] N. D. Botkin, M. Schlensog, M. Tewes, and V. L. Turova, "A mathematical model of a biosensor," *Nanotech.*, vol. 1, pp. 231–234, 2001.

[24] B. M. Patre and V. G. Sangam, "Mathematical model of an amperometric biosensor for the design of an appropriate instrumentation system," *J. Med. Eng. Technol.*, vol. 31, no. 5, pp. 351–360, 2007.

[25] V. Rossokhaty and N. Rossokhataja, "A mathematical model of silicon-based thermobiosensors," *IMA J. Manage. Math.*, vol. 10, no. 1, pp. 41–53, 1999.

[26] P. Rachela, N. Amir, and S. D. Yosi, "Mathematical model of whole cell based bio-chip: An electrochemical biosensor for water toxicity detection," *J. Electroanal. Chem.*, vol. 602, no. 1, pp. 17–23, 2007.

[27] M. Schlosshauer and D. Baker, "Realistic protein-protein association rates from a simple diffusional model neglecting long-range interactions, free energy barriers, and landscape ruggedness," *Protein Sci.*, vol. 13, pp. 1660–1669, 2004.



Ming Gu (M'12) received the B.E. and M.E. degrees from the School of Automation Science and Electrical Engineering, Beijing University of Aeronautics and Astronautics, Beijing, China, in 2000 and 2003, respectively, and the M.E. and Ph.D. degrees in electrical and computer engineering from Michigan State University, East Lansing, in 2008 and 2012, respectively.

From 2003 to 2006, she worked as a Research Engineer in the Center for Space Science and Applied Research, Chinese Academy of Sciences, Beijing, China. Since May 2012, she has been with Iwatt Inc., Campbell, CA. Her research interests include mixed signal VLSI circuits design, error-control coding, and analog signal processing.

Shantanu Chakrabartty (SM'99–M'04–S'09) received the B.Tech. degree from the Indian Institute of Technology, Delhi, India, and the M.S. and Ph.D. degrees in electrical engineering from Johns Hopkins University, Baltimore, MD, in 1996, 2002, and 2005, respectively.

He is currently an Associate Professor in the Department of Electrical and Computer Engineering at Michigan State University (MSU), East Lansing. From 1996–1999, he was with Qualcomm Inc., San Diego, CA. During 2002, he was a Visiting Researcher at The University of Tokyo, Tokyo, Japan. His work covers different aspects of analog computing, in particular nonvolatile circuits, and his current research interests include energy harvesting sensors and neuromorphic and hybrid circuits and systems.

Dr. Chakrabartty was a Catalyst Foundation Fellow from 1999–2004 and is a recipient of the National Science Foundation's Career Award, the University Teacher-Scholar Award from MSU, and the 2012 Technology of the Year Award from MSU Technologies. He currently serves as an associate editor for IEEE Transactions of Biomedical Circuits and Systems and the *Advances in Artificial Neural Systems* journal, as well as a review editor for the *Frontiers of Neuromorphic Engineering* journal.