Mapping the Diffusion Network into a Stochastic System in Very Large Scale Integration

Chen-Han Chien, Chih-Chen Lu, and Hsin Chen, *Member, IEEE*

***Abstract*—The Diffusion Network (DN) is a probabilistic model capable of recognising continuous-time, continuous-valued biomedical data. As the stochastic process of the DN is described by stochastic differential equations, realising the DN with analogue circuits is important to facilitate real-time simulation of a large network. This paper presents the translation of the DN into analogue Very Large Scale Integration (VLSI). With extensive simulation, the dynamic ranges of parameters and their representation in VLSI are identified. The VLSI circuits realising the stochastic unit of the DN are further designed and interconnected to form a stochastic system using noise to induce stochastic dynamics in VLSI. The circuit simulation demonstrate that the VLSI translation of the DN is satisfactory and the DN system is capable of using noise-induced stochastic dynamics to regenerate various types of continuous-time sequences.**

I. Introduction

The Diffusion Network (DN) proposed by Movellan is a stochastic recurrent network whose stochastic dynamics can be trained to model the probability distributions of continuous-time sequences by the Monte-Carlo Expectation-Maximisation (EM) algorithm [1], [2]. As stochasticity is useful for generalising the natural variability in data [3][4], the DN is further shown suitable for recognising noisy, continuous-time biomedical data [5]. However, the stochastic dynamics of the DN is defined by a set of continuous-time, stochastic differential equations. The speed of simulating stochastic differential equations in a digital computer is inherently limited by the serial processing and numerical iterations of the computer. Translating the DN into analogue circuits is thus important for simulating a large DN in real time. The hardware implementation of the DN could further function as an intelligent embedded system capable of recognising multichannel, time-varying biomedical signals in real time. Such a system would be useful for implantable biomedical microsystems, which aim to deliver bio-feedbacks or to control prosthetic devices in real time [6].

This paper presents the translation of the DN into a stochastic VLSI system using noise-induced stochasticity. Following a brief review on the DN model, the dynamic ranges of parameters essential for modelling data satisfactorily are first identified with Matlab simulation. The representation of parameters in VLSI are then defined, and the component circuits for realising the DN are designed. Finally, the simulated dynamics of the VLSI system are compared with those obtained in Matlab simulation to exam the fidelity of the VLSI translation.

Chen-Han Chien, Chih-Chen Lu, and Hsin Chen are with the Institute of Electronics Engineering, National Tsing Hua University, No.101, Sec.2, Kuang-Fu Road, HsinChu, 30013, Taiwan (phone: +886 35162221; email: hchen@ee.nthu.edu.tw).

II. The Diffusion Network

The DN consists of continuous-time, continuous-valued, stochastic units with fully recurrent connections, as illustrated in Fig.1a [1][2]. Let $x_i(t)$ represent the state of the unit i at time t, and w_{ij} the coupling strength from the unit i to the unit j. The state x_i is a random variable whose stochastic dynamics is governed by the following equation [1].

$$dx_i(t) = \mu_i(t) \cdot dt + \sigma \cdot dB_i(t) \tag{1}$$

where $\mu_i(t)$ is a deterministic *drift* term given in (2), σ a constant, and $dB_i(t)$ the *Brownian* motion. The Brownian motion introduces the stochasticity, enriching greatly the representational capability of the DN [2].

$$\mu_i(t) = \frac{1}{C_i}\left(\sum_j w_{ij} s_j(t) - x_i(t)/R_i\right) \tag{2}$$

$C_i > 0$ and $R_i > 0$ in (2) are adaptable parameters called *input capacitance* and *transmembrane resistance*, respectively, and $s_i(t) = \tanh(a_i \cdot x_i)$ with parameter a_i controlling the slope of the hyperbolic tangent function.

The stochastic units of the DN are divided into *visible* and *hidden units* (Fig.1a). By adapting parameters $w_{ij}, C_i, \text{and} R_i$ according to the Monte-Carlo EM algorithm (Appendix), the DN learns to regenerate the dynamics of training data as the stochastic dynamics of its visible units [2]. The stochasticity is useful for representing the variability of the training dynamics. Therefore, the number of visible units simply equals the dimension of training data, while the minimum number of hidden units for satisfactory modelling is normally identified through experimental trials. After training, all stochastic units of the DN are given their initial values (normally zero) at $t = 0$ and then sampled according to (1). The agreement between the sampled (regenerated) dynamics at visible neurons and the dynamics of training data then indicate how well the data are modelled.

III. Mapping the Diffusion Network into VLSI

A. Circuit architecture

The stochastic unit of the DN can be translated into the equivalent-circuit model in Fig.1b The resistor and the capacitor correspond to R_i and C_i in (2), respectively, and the

[1]Discrete-time approximation to the stochastic differential equations was adopted in the numerical simulation

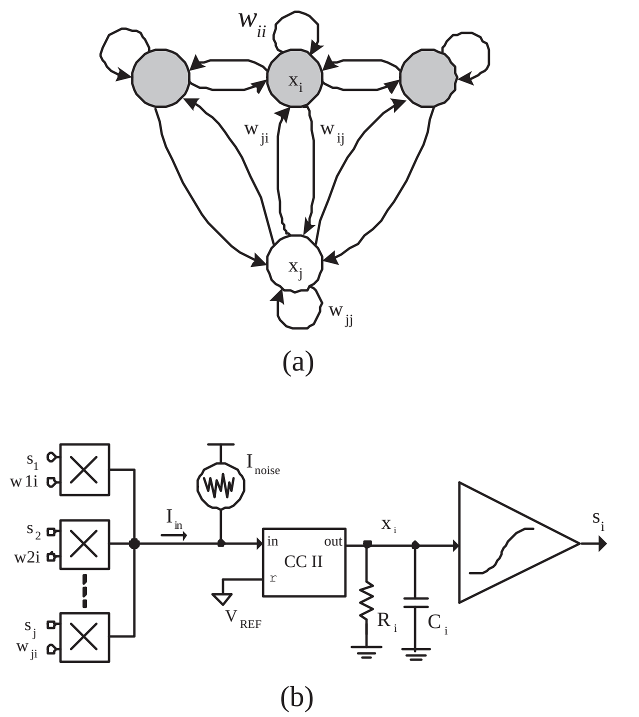


Fig. 1. (a)The diagram of a DN with one visible (white-coloured) and three hidden(grey-coloured) units. (b)The circuit architecture of a stochastic unit in the DN.

voltage at node x_i represent the state of the stochastic unit. The multipliers calculate $w_{ij} \cdot s_j$ and output a total current I_{in} proportional to $\sum_j w_{ij} s_j(t)$. I_{in} and the noise current I_{noise} are then summed up and buffered to R_i and C_i by the class II current conveyor [7], producing the stochastic dynamics at x_i according to (1). Finally, passing x_i through the sigmoid circuit gives s_i for the inputs of all stochastic units.

B. *Adapting w_{ij} and R_i only*

The simulation in [2] shows that adapting w_{ij} and C_i is sufficient for modelling various types of data. However, a variable resistor is easier to implement in VLSI than a variable capacitor. Fig.1b indicates that C_i and R_i cooperate to determine the "time constant" of the stochastic dynamics of each unit. The feasibility of modelling different data by adapting w_{ij} and R_i only is thus investigated.

With $C_i = 1$ and $\Delta t = 0.05$ for discrete-time iteration of (1) in Matlab, the DN was trained to model different types of data with their dynamic ranges normalised into $[-2, 2]$. A DN with one visible and one hidden units was first trained to model the bifurcating curves used in [2] (the black dashed lines in Fig.2). After 100 training epochs, the DN was capable of regenerating the bifurcating curves at its visible unit v_1 successfully, as shown by Fig.2. With two more hidden units, the DN was further trained to model ten QRS segments of the real heartbeat data extracted from the MIT-BIH database (the black lines in Fig.3). The reconstruction in Fig.3 indicated that the DN modelled not only the dynamics but also the

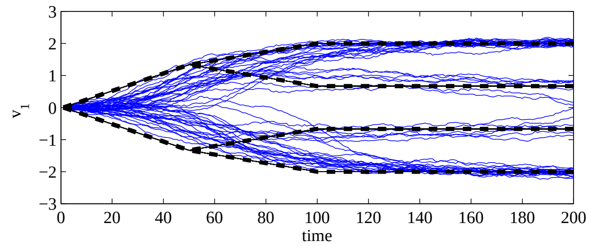


Fig. 2. The four bifurcating training data (black dashed lines) and 50 sequences (grey curves) regenerated by the DN after 100 training epochs. $a_i = 3$ and $\sigma = 0.1$ for all units. The numerical values of the horizontal axis correspond to the indexes of discrete-time samples, and the same representation is employed in Fig.3-Fig.5

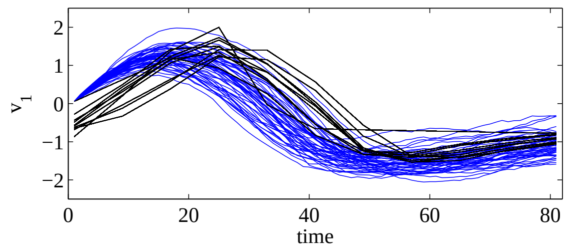


Fig. 3. 10 normal heartbeats (black curves) and 50 sequences (gray curves) regenerated by the DN after 100 training epochs. $a_i = 2$ and $\sigma = 0.1$ for all units.

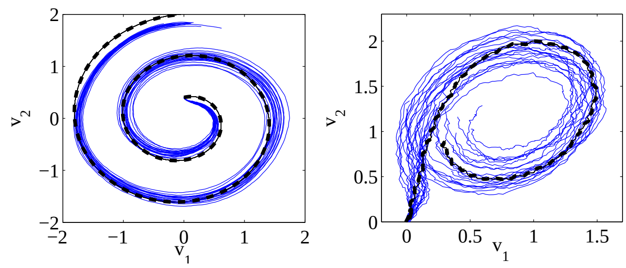


Fig. 4. (a)The spiral curve and (b) the handwritten ρ (black dashed lines) for training the DN. The gray curves in each subplot are 50 sequences regenerated by the DN after 120 training epochs. $a_i = 0.8$ and $\sigma = 0.01$ for modelling (a), while $a_i = 0.5$ and $\sigma = 0.04$ for modelling (b).

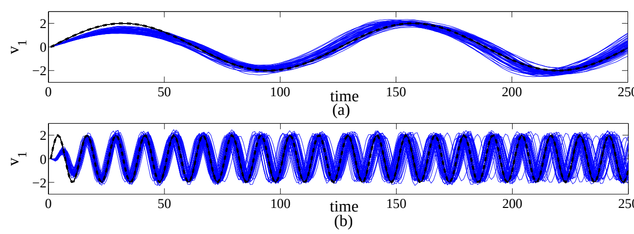


Fig. 5. The sinusoidal waves (black dashed lines) with frequency (a)f_0 and (b)$10f_0$. The gray curves in each subplot are 50 sequences regenerated by the DN after 100 training epochs. $a_i = 1$ and $\sigma = 0.05$ for all units.

variability of the real heartbeat data. Moreover, the spiral curve and the hand-written ρ shown by the black dashed lines in Fig.4 were used to examine the DN's capability to model two-dimensional data. As shown by Fig.4, a DN with two visible and five hidden units was able to regenerate the spiral curve and the handwritten ρ as the stochastic dynamics of the visible units v_1 and v_2. These results demonstrate that the DN can model both artificial and biomedical data satisfactorily by adapting w_{ij} and R_i only.

C. Dynamic ranges of parameters

As indicated by Fig.1b, a stochastic unit with a specific R_i and C_i needs to increase w_{ij} or s_j to increase the maximum changing rate of the dynamics at x_i. To identify the dynamic ranges required by all parameters, a DN with one visible and three hidden units is trained to model sinusoidal waves at different frequencies. Let the frequency of the sinusoidal wave in Fig.5a be f_0. The sinusoidal wave with $10f_0$ in Fig.5b is selected as the fastest dynamics to be modelled by the DN in VLSI. For data with even more faster dynamics, the data can always be slowed down by over-sampling and then expanding the samples along the time axis. After extensive simulations, the dynamic ranges required for modelling the sinusoidal waves with a frequency ranging from f_o to $10f_o$, as well as the artificial and biomedical data in Sec.III-B, are identified and summarised in Table.I. Fig.5 shows that the DN can model the sinusoidal waves satisfactorily within the parameter ranges.

D. Mappings between numerical simulation and VLSI implementation

The 0.35 μm CMOS technology provided by the Taiwan Semiconductor Manufacturing Company(TSMC) is used to realise the DN in VLSI. According to the architecture in Fig.1b, the parameters of the DN are further represented as currents or voltages in VLSI, as summarised by the third column in Table.I. Although the rating supply voltage of the technology is 3V, x_i is represented as a voltage between [0.9,2.1]V, in order to limit the maximum I_{in} required, as well as to ease the design of the variable resistor and the sigmoid circuit. As extensive simulation indicates that w_{ij} requires not only a dynamic range of $[-30, 30]$ but also a resolution to the first floating number, w_{ij} is represented as a current ranging from $-5\mu A$ to $5\mu A$ to avoid noise interferences. $\Delta t = 0.05$ is further set to be $5\mu s$ in VLSI, corresponding to a reasonable sampling rate (200kHz) at which most instruments can sample multiple channels(units) simultaneously.

Moreover, the mappings of C_i and R_i depend on the unit values of x_i, I_{in}, and Δt. According to (2), $R_i = 1$ and $x_i = 1$ result in $x_i/R_i = 1$, corresponding to a unit current of $I_{unit} = 660nA/30 = 22nA$ at node x_i in Fig.1b. As the unit voltage for x_i is $V_{unit} = 0.6V/3 = 0.2$ (Table.I), the unit value of R_i simply equals $R_{unit} = V_{unit}/I_{unit} \approx 9.1M\Omega$. Multiplying the unit resistance with the numerical ranges of R_i gives its dynamic range required in VLSI. Similarly, the unit value of C_i is calculated as $I_{unit} \cdot \Delta t_{unit}/V_{unit} =$

TABLE I
THE DYNAMIC RANGES OF PARAMETERS AND THEIR MAPPINGS BETWEEN MATLAB SIMULATION AND VLSI IMPLEMENTATION

Parameter	Matlab	VLSI
x_i	$-3 \sim 3$	$0.9V \sim 2.1V$
w_{ij}	$-30 \sim 30$	$-5\mu A \sim 5\mu A$
I_{in}	$-30 \sim 30$	$-660nA \sim 660nA$
Δt	0.05	$5\mu s$
R_i	$0.5 \sim 4$	$4.55M\Omega \sim 36.4M\Omega$
C_i	1	$11pF$
s_i	$-1 \sim 1$	$-2\mu A \sim 2\mu A$
a_i	$0.5 \sim 10$	$0.4\mu A \sim 11\mu A$
σ	$0.1 \sim 0.5$	$29.5nA \sim 147.5nA$

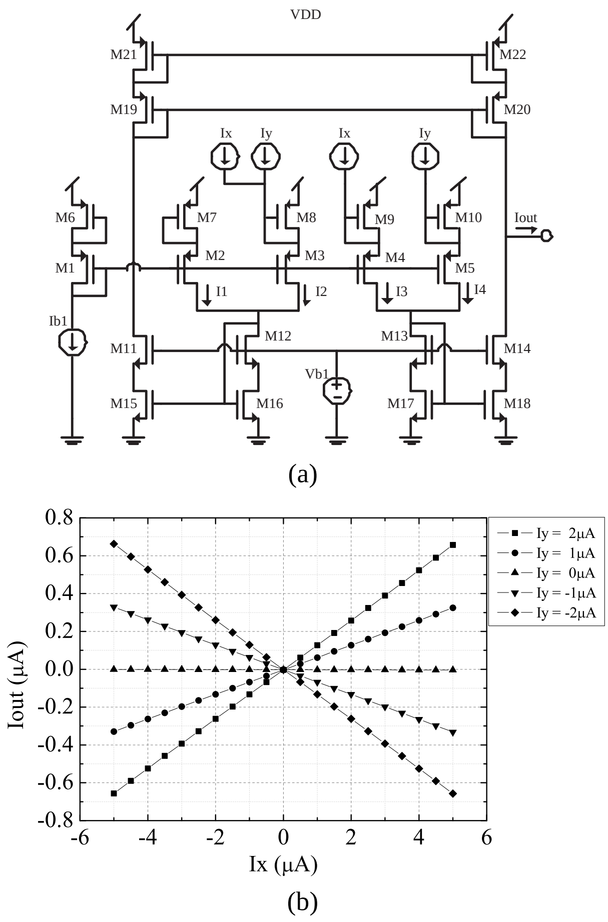
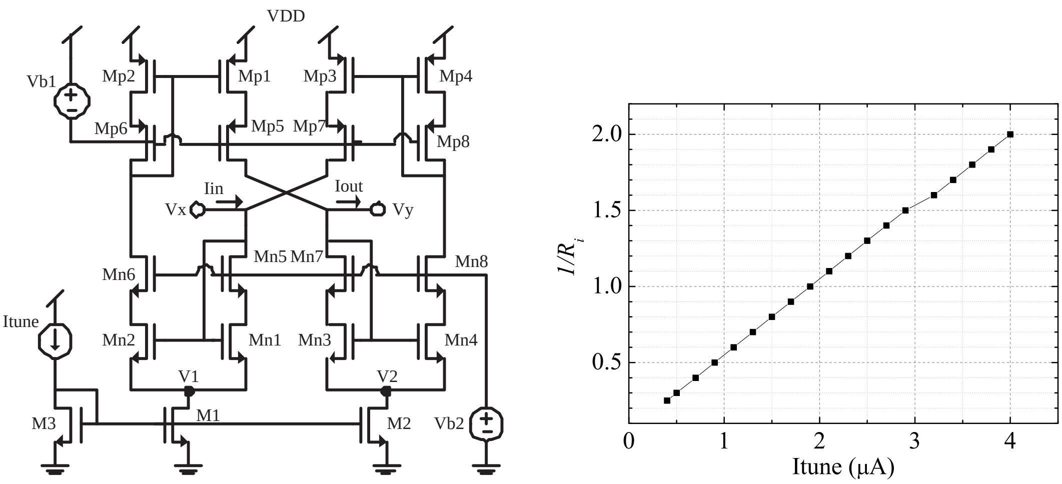


Fig. 6. (a)The four-quadrant multiplier and its (b)simulated DC characteristics.

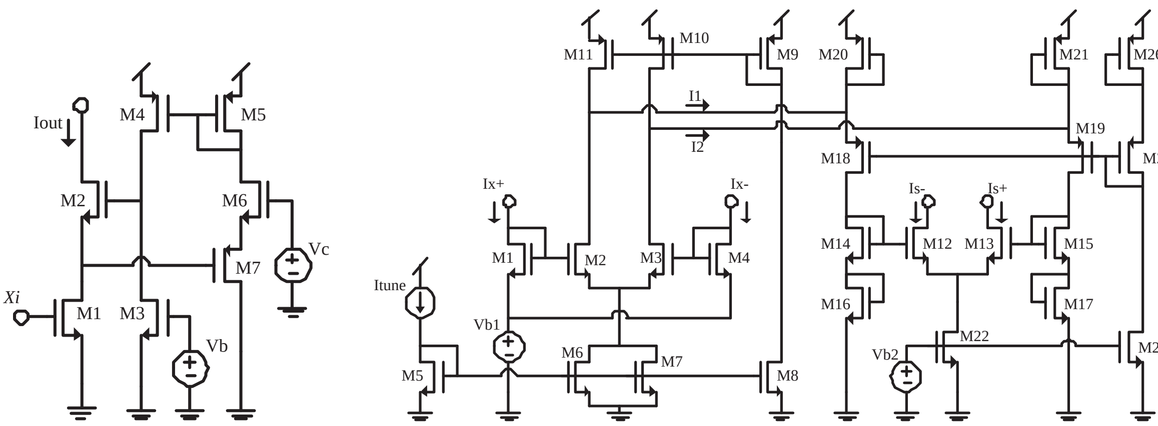
$11pF$, with $t_{unit} = 5\mu s/0.05 = 100\mu s$. Multiplying the unit capacitance with numerical ranges of C_i then gives its dynamic ranges in VLSI.

For σ, the noise term in (1) introduces a change of $dx_i = \sigma \cdot z \cdot \sqrt{dt}$ in discrete-time simulation (z represents an unit Gaussian). The noise current required to cause the same



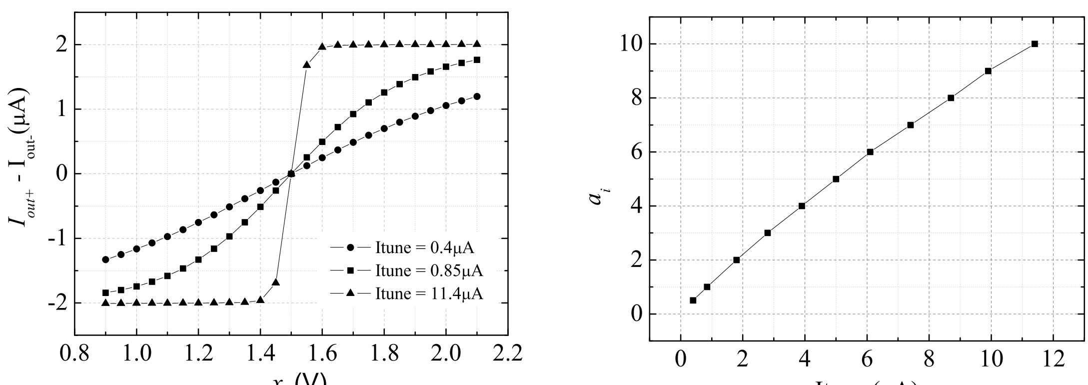
(a) (b)

Fig. 7. (a)The tunable active resistor (b)The simulated relationship between $1/R_i$ and I_{tune}.



(a) (b)

Fig. 8. (a)The current-to-voltage converter. (b)The sigmoid circuit.



(a) (b)

Fig. 9. (a)The simulated output of the sigmoid circuit and the V-I converter (b)The simulated a_i versus I_{tune}.

amount of dx_i is given as

$$i_n = \frac{C_i \cdot dx_i}{dt} = \frac{C_i \cdot \sigma \cdot z}{\sqrt{dt}} \qquad (3)$$

Substituting the maximum of $z = 3$, $dt = 0.05$ and $C_i = 1$ then gives the numerical relationship between i_n and σ. Multiplying i_n with I_{unit} then gives the mapping of σ.

IV. VLSI Design of the Stochastic Unit

The current-mode, four-quadrant multiplier proposed in [8] is employed to calculate $w_{ij} \cdot s_j$. Fig.6a shows the multiplier basing on the current squarers formed by M1-M10. I_2, I_3, and I_4 are proportional to the square of $(I_x + I_y)$, I_x, and I_y, respectively [8]. The current mirrors M11-M20 then calculate $I_1 + I_2 - I_3 - I_4$, resulting in $I_{out} = I_x \cdot I_y / 4I_{b1}$. The relationship holds as long as $|I_x + I_y| < 4I_{b1}$. Fig.6b shows the simulation result, indicating that the dynamic ranges defined in Table.I are met with negligible nonlinearity.

As shown in Fig.7a, the active resistor proposed in [9] is adopted to implement R_i. Let R represent the resistance between ground and V_1 or V_2. The resistance is tunable through $Itune$. Transistors Mn1-Mn8 and Mp1-Mp8 form four cascode current mirrors, resulting in an effective resistance of $2R$ between the nodes V_x and V_y. Fig.7b shows the simulated relationship between I_{tune} and $1/R_i$, demonstrating the required dynamic range is achieved.

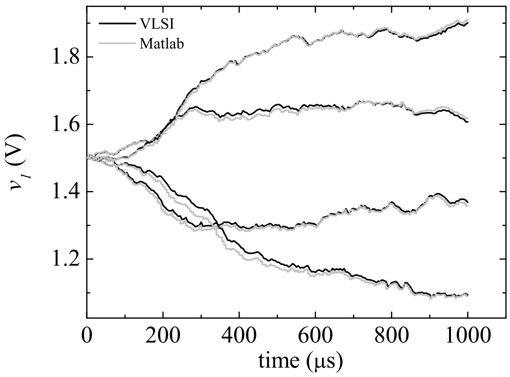
For the sigmoid function, the voltage x_i is first converted by the voltage-to-current(VI) converter in Fig.8a, and then delivered to the inputs of the current-mode sigmoid circuit in Fig.8b [10]. The VI conversion is simply achieved by operating M1 in triode region, and the operating mode is set by V_c which defines the drain voltage of M1. The converted current is then replicated by current mirrors and subtracted from a reference current to produce differential input currents ($I_{x+} \text{and} I_{x-}$) for the sigmoid circuit. Let $I_{tune} = kI_B$ and $I_{x+} - I_{x-} = xWI_B$ with $W = (I_{x+} + I_{x-})^{-1}$ in the sigmoid circuit. Transistors Ms1-Ms4 operate in the subthreshold region, forming a translinear loop that produces a differential current $I_2 - I_1 = xW2kI_B$. Transistors Ms12-M21 and M25-M26 further form translinear loops by operating in the subthreshold region, resulting in

$$\begin{aligned} I_{s+} - I_{s-} &= I_B \cdot f(x) \\ &= I_B \cdot \frac{xWk}{(xWk)^2 + 2} \sqrt{(xWk)^2 + 4} \end{aligned} \qquad (4)$$

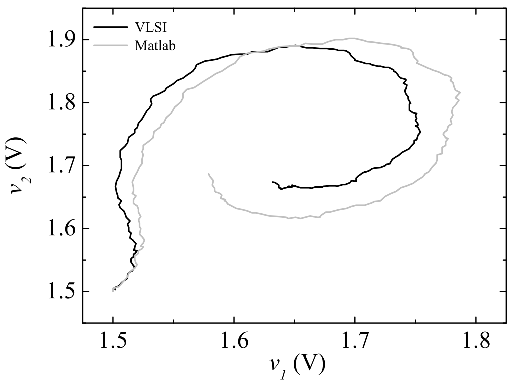
where $f(x)$ approximates $\tanh(a_i \cdot x_i)$ in (2) and k corresponds to a_i, adapting the slope of $f(x)$. Fig.9a shows the simulated input-output relationship of the sigmoid circuit connected with the V-I converter. The relationship between I_{tune} and a_i is further derived and shown in Fig.9b. Finally, the noise generator is implemented by the analogue random vector generator basing on cellular automata [11].

V. The Diffusion Network in VLSI

The designed VLSI stochastic units are interconnected to form a DN system and simulated with HSPICE. To facilitate the comparison between VLSI and Matlab simulation, the same noise sequence is employed in both simulations during the generation of specific stochastic dynamics. With one visible and one hidden neurons, the DN system can regenerate the bifurcating dynamics as shown by the black curves in Fig.10a, agreeing with the Matlab simulation (grey curves) satisfactorily. With two visible and five hidden neurons, the DN system can be further programmed to generate the hand-written ρ, as shown by Fig.10c. The promising results demonstrate the satisfactory mapping of the DN model into VLSI circuits. A chip containing a DN of seven neurons is thus fabricated with the TSMC 0.35 μm CMOS technology. The measurement results will be presented in the conference.



(a)



(b)

Fig. 10. (a)The bifurcating data and (b)The handwritten ρ reconstructed by the DN system (black) and the Matlab simulation (grey).

VI. Conclusion

With extensive simulation, the dynamic ranges of the parameters of the DN have been identified and mapped into VLSI. This underpins the realisation of the DN with analogue VLSI circuits. A DN system in VLSI has thus been designed and fabricated. The circuit simulation demonstrates that the DN model has been mapped into a stochastic VLSI system satisfactorily. Based on the DN theory, the VLSI system with noise-induced stochasticity will be able to adapt its stochastic dynamics towards modelling various continuous-time, continuous-valued data. This capability makes it a

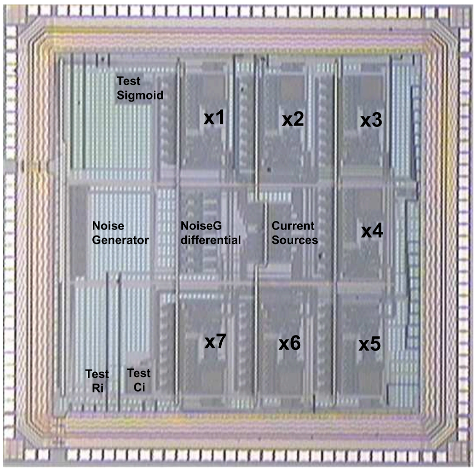


Fig. 11. The VLSI system of the Diffusion Network. The chip area is $3.5 \times 3.5mm^2$.

potential solution for recognising high-dimensional, time-varying biomedical signals in implantable microsystems.

Appendix : The Monte-Carlo EM Algorithm

To train a DN with n visible and m hidden units, the dynamics of visible units are forced to follow the desired sequences (training data), while the corresponding dynamics of hidden units are sampled, in accordance with (1) and (2), for l times. Let $x_i^l(t) : [0,T] \rightarrow \mathbb{R}$ denote the l-th sampled dynamics of the unit i. A weighting factor proportional to the likelihood of the l-th Monte-Carlo sample is calculated according to

$$\begin{aligned} \pi(l) = \exp\{ &\frac{1}{\sigma^2} \sum_{i=1}^{n} \int_0^T \mu_i(t) dx_i(t) \\ &- \frac{1}{2\sigma^2} \sum_{i=1}^{n} \int_0^T \mu_i(t)^2 dt \} \end{aligned} \tag{5}$$

To obtain the maximum-likelihood estimate of the connection matrix $\hat{\mathbf{w}} = \{w_{ij}\}$, the $\hat{a}$ and $\hat{b}$ matrixes are first computed for each Monte-Carlo sample according to

$$a_{ij}(l) = \int_0^T \varphi(x_i^l(t)) \varphi(x_j^l(t)) dt, \tag{6}$$

$$\begin{aligned} b_{ij}(l) = &C_j \int_0^T \varphi(x_i^l(t)) dx_j^l(t) \\ &+ \frac{1}{R_j} \int_0^T \varphi(x_i^l(t)) x_j^l(t) dt, \end{aligned} \tag{7}$$

where $\varphi(\cdot)$ represents the sigmoid function. The connection matrix is then estimated as $\hat{\mathbf{w}} = \bar{\mathbf{a}}^{-1}\bar{\mathbf{b}}$ with $\bar{\mathbf{a}}$ and $\bar{\mathbf{b}}$ given in (8) and (9), respectively.

$$\bar{\mathbf{a}} = \frac{\sum_{l=1}^{m} \pi(l) a(\hat{l})}{\sum_{l=1}^{m} \pi(l)}, \tag{8}$$

$$\bar{\mathbf{b}} = \frac{\sum_{l=1}^{m} \pi(l) b(\hat{l})}{\sum_{l=1}^{m} \pi(l)}, \tag{9}$$

On the other hand, the maximum likelihood estimates for C_i and R_i are calculated for each Monte-Carlo sample according to (10) and (11), respectively.

$$\hat{C}_i = \frac{C_i \int_0^T \mu_i^l(t)^2 dt}{\int_0^T \mu_i^l(t) dx_i^l(t)} \tag{10}$$

$$\hat{R}_i = \frac{\int_0^T x_i^l(t)^2 dt}{\int_0^T \sum_{j=1}^{n} \varphi(x_j^l(t)) w_{ij} x_i^l(t) dt - C_i \int_0^T x_i^l(t) dx_i^l(t)} \tag{11}$$

The opinions of the l estimates for R_i and C_i are also weighted by $\pi(l)$ to obtain their optimum estimates.

Acknowledgments

The authors would like to devote their greatest thanks to the National Science Council in Taiwan for funding this project (NSC 96-2221-E-007-167-MY3).

References

[1] J. R. Movellan, "A learning theorem for networks at detailed stochastic equilibrium," *Neural Computation*, vol. 10, no. 5, pp. 1157–1178, 1998.

[2] J. R. Movellan, P. Mineiro, and R. J. Williams, "A Monte-Carlo EM approach for partially observable diffusion processes: Theory and applications to neural networks," *Neural Computation*, vol. 14, no. 7, pp. 1507–1544, 2002.

[3] H. Chen and A. F. Murray, "A continuous restricted boltzmann machine with an implementable training algorithm," *IEE Proceedings of Vision, Image and Signal Processing*, vol. 150, no. 3, pp. 153–158, 2003.

[4] D. Specht, "Probabilistic neural networks," *Neural Networks*, vol. 3, no. *, pp. 109–118, 1990.

[5] Y. Hsu, T. Chiu, and H. Chen, "Real-time recognition of continuous-time biomedical signals using the diffusion network," in *Proceedings of the International Joint Conference on Neural Networks*, 2008, pp. 2628–2633.

[6] M. Lebedev and M. Nicolelis, "Brain-machine interfaces: past, present and future," *Trends in Neuroscience*, vol. 29, no. 9, pp. 536–546, 2006.

[7] C. Toumazou, F. J. Lidgey, and D. G. Haigh, *Analogue IC Design: The Current-Mode Approach*, 1st ed. Institution of Engineering and Technology, 1990.

[8] K. Tanno, O. Ishizuka, and Z. Tang, "Four-quadrant cmos current-mode multiplier independent of device parameters," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 47, no. 5, pp. 473–477, 2000.

[9] M. Al-Sarawi, "A novel technology for grounded-to-floating resistor coversion in cmos technology," *Microelectronics Journal*, vol. 33, no. 12, pp. 1059–1069, 2002.

[10] M. Valle, *Smart Adaptive Systems on Silicon*. Kluwer Academic Publishers, 2004, ch. Supervised Learning Systems oon Silicon, pp. 139–156.

[11] G. Cauwenberghs, "Delta-sigma cellular automata for analog vlsi random vector generation," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 46, no. 3, pp. 240–250, 1999.