A Scalable and Programmable Probabilistic Generative Model in Very Large Scale Integration

Journal:	*Transactions on Neural Networks*
Manuscript ID:	Draft
Manuscript Type:	Paper
Date Submitted by the Author:	
Complete List of Authors:	Lu, Chih; NTHU, ENE Chen, Hsin; National Tsing Hua University, Electrical Engineering
Keywords:	VLSI Implementation, Noise, Probabilistic Model, Boltzamnn Machine, Scalable and Programmable Systems, Stochastic systems

A Scalable and Programmable Probabilistic Generative Model in Very Large Scale Integration

C.C. Lu and H. Chen, *Member, IEEE*

Institute of Electronics Engineering

National Tsinghua University, Hsinchu, Taiwan

ABSTRACT

An intelligent embedded system capable of preprocessing noisy, sensory data has been demanded in many biomedical applications. The paper presents the Very-Large-Scale-Integration (VLSI) implementation of a scalable and programmable Continuous Restricted Boltzmann Machine (CRBM), a probabilistic model proved useful for recognising biomedical data. The scalability allows the network size to be expanded by interconnecting multiple chips, and the programmability allows all parameters to be remained at their optimum values. Each chip contains 10 stochastic neurons, 25 synapses, and a 10-channel noise generator for the neurons. Basing on the CRBM algorithm, the VLSI system is able to use noise-induced, continuous-valued stochasticity to model both artificial and real biomedical data. With the noise generator turned off, the system can also function as a multi-layer perceptron. By interconnecting multiple chips to form particular CRBM networks, the ability to classify both artificial and biomedical data in real time is further demonstrated. The proposed CRBM system thus provides a potential solution for the intelligent system in demand.

Index terms: VLSI Implementation, Noise, Probabilistic Model, Boltzamnn Machine, Scalable and Programmable Systems, Stochastic systems.

Corresponding author: H. Chen (email: hchen@ee.nthu.edu.tw)

Acknowledgement: The authors would like to thanks the National Science Council (NSC) in Taiwan for funding this project (Grant code : NSC 96-2221-E-007-167-MY3), and to the Chip Implementation Center in Taiwan and the Taiwan Semiconductor Manufacturing Co. for fabricating the chip.

I. INTRODUCTION

In the development of implantable devices and bio-electronic interfaces [18][24][26][27][28], exposing electronic systems to noisy environments has become inevitable. Although sensory data could be transmitted wirelessly out of implanted devices and then processed by sophisticated algorithms, transmitting all the raw data is power-consuming and unfavourable for long-term monitoring. Applications like neural prostheses further look for the possibility of recognising biomedical signals on line, so as to deliver bio-feedbacks or to control prosthetic limbs in real time. Therefore, an intelligent embedded system which is robust against noise and capable of extracting useful information from noisy, high-dimensional biomedical signals is becoming essential. Moreover, the robustness against the intrinsic electronic noise is important as the transistor size shrinks toward the deep-submicron scale [15].

Probabilistic models are able to use stochasticity to generalise the natural variability of data. Many probabilistic models have been shown promising for reasoning biomedical data or for solving weakly-constrained problems such as pattern recognition. Therefore, realising probabilistic models in VLSI is attractive for applications like intelligent sensor fusion in implantable devices [9][26]. However, only a few probabilistic models are amenable to VLSI implementation [7][10][14], and most of which relies greatly on precise computation of Bayesian rules or vector products. Maintaining the precision becomes difficult as transistor noise and hardware non-ideality grow.

Contrarily, the probabilistic model called the Continuous Restricted Boltzmann Machine (CRBM) has been shown capable of classifying biomedical data reliably, as well as realised in VLSI with noise-induced stochasticity [6][7]. The noise is not only used to model data variability basing on the CRBM, but also proved useful to enhance the robustness against interferences [18]. The CRBM system in VLSI is thus potential for intelligent sensor fusion in implantable devices. However, the prototype system with only six neurons is limited to model two-dimensional data, while real biomedical signals are normally high-dimensional and complex.

To alleviate this limitation, this paper presents a scalable and programmable CRBM system in VLSI. The full system is designed and fabricated with the TSMC 0.35μm CMOS technology. The scalability allows the network size to be expanded by interconnecting multiple chips. The programmability allows all parameters on-chip to be refreshed to optimum values, or to be trained by the chip-in-a-loop configuration [4]. A multi-channel noise generator is further included on-chip to induce continuous-valued stochasticity required by the CRBM. Following a brief introduction to the CRBM model, the architecture, the circuit design, and the measurements of the modular CRBM (mCRBM) system are presented. The ability of the mCRBM system to regenerate different continuous-valued distributions, as well as to classify artificial data is then tested. Finally, the mCRBM system's capability of modelling and classifying real biomedical data is examined in the context of sorting neuronal spikes.

II. THE CRBM MODEL

The CRBM consists of one visible and one hidden layers of stochastic neurons with inter-layer connections only, as shown in Fig.1. As a generative model, the CRBM learns to "regenerate" the training data as the states of its visible neurons. The number of visible neurons thus equal to the dimension of the training data, while the minimum number of hidden neurons required to model data satisfactorily is indentified by experimental trials [7]. Let w_{ij} represent the bi-directional connection between neurons v_i and h_j, and let s_i denote either v_i or h_j in the following content. The stochastic state of a neuron s_i is defined as

$$s_i = \varphi\left(a_i \cdot \left(\sum_j w_{ij} \cdot s_j + N_i(\sigma,0) \right) \right) \tag{1}$$

where $N_i(\sigma,0)$ represents a Gaussian noise with zero mean and variance σ^2, and $\varphi(\cdot)$ a sigmoid function (e.g. $tanh(\cdot)$) with asymptotes at ±1. Parameter a_i controls the slope of the sigmoid function and thus the variance of s_i. Therefore, the stochastic behaviour of a neuron is either near-deterministic (small a_i), or continuous-stochastic (moderate a_i), or binary-stochastic (large a_i). Let λ represent the parameter $\{w_{ij}\}$ or $\{a_i\}$. Parameters in a CRBM system are trained by the simplified minimising-contrastive-divergence (MCD) algorithm [6]

$$\Delta\lambda = \eta_\lambda \cdot sign\left(\langle s_i \cdot s_j \rangle_4 - \langle \hat{s}_i \cdot \hat{s}_j \rangle_4 \right) \tag{2}$$

where $\hat{s}_i$ and $\hat{s}_j$ denotes the one-step, Gibbs-sampled states, η_λ the updating rate, and $<\cdot>_4$ the expectation over four training data. For training $\{a_i\}$, s_j and $\hat{s}_j$ in Eq.(2) are replaced by s_i and $\hat{s}_i$, respectively.

The data distribution learnt by the CRBM can be obtained by initialising visible neurons with random values, and then Gibbs sampling hidden and visible neurons alternatively for multiple steps. The N-th step sample of the visible neurons is called the *N-step reconstruction*, approximating the distribution modelled by the CRBM when N is large (normally N>10 is sufficient [7]). The similarity between the N-step reconstruction and the training data then indicates how well the data is modelled. After training, testing data can be categorised according to the responses of hidden neurons [7].

III. SYSTEM ARCHITECTURE

Fig.2 shows the architecture of the scalable and programmable CRBM system [17], containing neuron modules (v_i and h_j), synapse modules (w_{ij}), a noise generator, and digital control circuits. The programming unit is realised by a microcontroller off-chip. Each synapse module (Fig.2b) contains two multipliers, calculating $w_{ij}h_j$ and $w_{ij}v_i$ as the current inputs for neurons v_i and h_j, respectively. Each visible (hidden) neuron then sums up the currents on the same row (column) at terminal I, and outputs a voltage representing s_i at terminal O (Fig.2c). In addition, a noise input is included to makes s_i probabilistic. Without noise, the neurons become deterministic perceptrons in the multi-layer perceptron [25].

The modular design enables the CRBM system to have 5M visible and 5N hidden neurons by interconnecting an MxN chip array. The synapse modules in the same row (column) transmit output currents to the left- (bottom-) most neurons in the row (column). Each neuron then transmits its voltage output back to the synapse modules in the same row (column). As shown by Fig.2(c), the control signal N (determines whether a neuron is enabled. With N=1, the current inputs at terminal I are passed through the sigmoid circuit to generate the neuron's output, while with N=0, the current inputs are simply directed to terminal X and the neuron output is buffered from terminal S into terminal O.

Moreover, a current normaliser is included to avoid the saturation of the sigmoid circuit as a large number of synapses are connected.

The parameters $\{w_{ij}\}$ and $\{a_i\}$ are stored locally as voltages across capacitors in the synapse and the neuron modules, respectively. The updating circuit proposed in [3] is then employed to tune the capacitor voltages periodically according to the digital input P. In training mode, P is calculated according to Eq.(2). As soon as optimum levels are obtained, the programming unit stores $\{w_{ij}\}$ and $\{a_i\}$ into its digital memory by using an analogue-to-digital converter (ADC) (Fig.2(d)). In refreshing mode, parameter values on the capacitors are sampled periodically by the ADC, compared with their optimum levels stored in the memory, and then updated according to the P determined by comparison. It is notable that one ADC can be shared by all parameters.

IV. Circuit Design and Measurements

Fig.3 shows the layout of the mCRBM system, fabricated with the TSMC 0.35µm CMOS technology. The circuit area excluding the pads is 4200µm×4200µm, and the power consumption is 20.4mW. The synapse module employs the "modified Chible multiplier" proposed in [5] to calculate $w_{ij}s_j$, and the analogue random vector generator proposed in [2] is implemented to generate 10 channels of uncorrelated noise on-chip. The following subsections describe the circuits and the measurement results of the neuron module and the programmable parameter array.

A. *Continuous Stochastic Neurons in VLSI*

Fig.4 shows the circuit diagram realising the CRBM neuron described by Eq.(1). The four-quadrant multipliers of synapses modules output a total current proportional to $i_{sum} = \sum_j w_{ij}s_j$ [5]. The differential pair, Mna and Mnb, then transforms noise voltage

V_{ni} into a noise current $i_n = g_m(V_{ni}-V_{nr})$. The transconductance g_m is controlled by V_{sig} through M7, scaling the noise current as σ in Eq.(1). Afterwards, i_{in} representing ($\sum_j w_{ij}s_j+\sigma\cdot N_i(0,1)$) is converted into V_x by the operational amplifier with an active resistor. The sigmoid circuit basing on transistors (M_{siga} and M_{sigb}) in subthreshold operation produces an output of

$$i_o = i_{c1} - i_{c2} = I_b \cdot \varphi\left(\frac{i_{in} \cdot R(V_{asi})}{V_t}\right) \tag{3}$$

where $R(V_{asi})$ denotes the resistance of the active resistor controlled by V_{asi}, and V_t represents the thermal voltage (kT/q). Finally, the resistor R_T converts i_o into a voltage V_o, representing s_i in Eq.(1), and the switch samples a continuous-valued output state, V_{si}. The dynamic ranges of the parameters and their mappings between numerical simulation and VLSI implementation are summarised in Table.1. The unit values in the last column are obtained by dividing the VLSI to the numerical values.

Fig.5(a) showed the measured characteristics of a sigmoid circuit. With i_{in} swept from -10μA to 10μA, the output V_o was measured. Different curves corresponded to different V_{asi}, controlling the sigmoid slope as a_i in Eq.(1). According to [6], the mapping between V_{asi} and a_i was derived and shown in Fig.5(b). Fig.6(a) showed the scatter plot of the noise voltages measured from two channels of the noise generator. The nearly-uniform distribution of data points indicated that the correlation between the two channels was negligible. Similar results were obtained for any two channels. Fig.6(b) further showed the statistical distribution of the noise amplitudes in one channel, which approximated an uniform distribution within [1, 2]V. Although circuits for converting uniform into Gaussian distributions were available [22], the

CRBM simply used noise to model data variance. The distribution of noise was not necessary to be Gaussian, as demonstrated in [15]. The uniform-distributed noise was thus sent to the CRBM neurons directly. According to [6], the mapping between V_{sig} and σ was derived and shown in Fig.6(c).

The transient response of the neuron H0 was tested by setting w_{i0} = 3 V and sweeping all v_i between 1V and 2V (corresponding to $v_i = \pm 1$). With V_{asi} =1.75V and V_{sig} =0.55V, the neuron generated the continuous-valued, stochastic dynamics shown in Fig.7. As the clock went low, an analogue voltage was sampled and held as the Gibbs-sampled state of the neuron.

B. Programmable Parameter Array

Each mCRBM system contains 35 parameters (25 w_{ij} and 10 a_i), which are arranged into a 5x7 array and multiplexed by the architecture shown in Fig.8(a). The corresponding digital-control signals are shown in Fig.8(b). With C0-C4 decoded from CK[0:2], five parameter values in the same column (w_{0j}-w_{4j}) are selected sequentially by the MUX_j, and then transmitted to the off-chip ADC. In refreshing mode, w_{0j} is first compared with its target value during C0=1. The signal $\overline{INC}/DEC[:0]$ representing the update direction is then obtained and stored in a register. Following the same procedure, the update directions for w_{1j}- w_{4j} are obtained as C1-C4 becomes high sequentially. CKup=1 then triggers the updating circuits (Fig.8(c)) to tune all parameter once. The refreshing step is controlled by the pulse-width of the signal VPLS [3]. In training mode, update directions are calculated according to Eq.(2), and then used to adapt the parameter array in a similar manner.

Fig.9(a) showed the measured characteristic of the updating circuit in refreshing mode. With VP=2.46V, VN=0.57V, and a pulse width of 320ns for VPLS, an updating step of 12mV was easily achieved for both incremental ($\overline{INC}/DEC=0$) and decremental ($\overline{INC}/DEC=1$) updates. The updating step could be further decreased by reducing the pulse width, but the background and the switching noise made the updating step hardly visible. The programmability of parameters in the same column was further tested and shown in Fig.9(b). After $w_{00} \sim w_{30}$ were read out sequentially, CKup triggered update circuits to refresh the parameters once. The refreshing frequency was around 770 Hz.

V. Regenerating Continuous-Valued Data Distributions

The mCRBM system's ability to regenerate different data distributions was tested by (1) training the CRBM in Matlab to derive parameter values for modelling a dataset, (2) programming the parameters in VLSI to the derived values, and (3) generating multi-step reconstructions from the mCRBM system. During multi-step reconstruction, initial data was transmitted to visible neurons with their N=0. Hidden and visible neurons were then sampled alternatively with N=1 for all neurons. The experimental results are presented in the following subsections.

A. *Regenerating Two-Dimensional Data with a Symmetric Distribution*

A CRBM with two visible and four hidden neurons was trained to model the two-dimensional data in Fig.10(a), consisting of two clusters of 200 Gaussian-distributed data points. According to Table.I and Fig.5(b), the learnt parameters values were translated into voltages as

$$\{w_{ij}\} = \begin{bmatrix} \times & 1.4723 & 1.5614 & 1.7435 & 1.4626 \\ 1.6107 & 1.8579 & 1.4199 & 1.4816 & 1.5497 \\ 1.5157 & 1.2823 & 1.4605 & 1.5858 & 1.5215 \end{bmatrix} \text{(Volts)}$$

$$\{a_{vi}\} = \begin{bmatrix} 1.75 & 1.75 \end{bmatrix} \text{(Volts)}$$

$$\{a_{hi}\} = \begin{bmatrix} 2.25 & 1.75 & 1.85 & 1.25 \end{bmatrix} \text{(Volts)} \tag{4}$$

With $\{w_{ij}\}$ and $\{a_i\}$ refreshed to the voltage levels in Eq.(4), the mCRBM system generated the 15-step reconstruction in Fig.10(b). The similarity between Fig.10(a) and (b) demonstrated that the CRBM system was able to regenerate the training data satisfactorily. In addition, the noise in VLSI was used to regenerate the data variance, causing each reconstructed cluster to exhibit a square shape due to the uniform-distributed noise.

B. Regenerating Two-Dimensional Data with a Non-symmetric Distribution

A CRBM with two visible and four hidden neurons was further trained to model the non-symmetric distribution in Fig.11(a), comprising of 400 data points sampled from one elliptic Gaussian and one circular Gaussian. After 4000 training epochs, the parameters were derived and slightly adjusted to be

$$\{w_{ij}\} = \begin{bmatrix} \times & 1.4042 & 1.6019 & 1.5847 & 1.6215 \\ 1.6107 & 1.8000 & 0.8000 & 2.0000 & 1.5677 \\ 1.5157 & 1.7000 & 2.3000 & 1.9000 & 1.5782 \end{bmatrix} \text{(Volts)}$$

$$\{a_{vi}\} = \begin{bmatrix} 1.50 & 1.30 \end{bmatrix} \text{(Volts)}$$

$$\{a_{hi}\} = \begin{bmatrix} 1.78 & 2.30 & 1.95 & 1.30 \end{bmatrix} \text{(Volts)} \tag{5}$$

The adjustments were mainly for compensating the nonlinearity of multipliers, as had been done in [6]. With $\{w_{ij}\}$ and $\{a_i\}$ refreshed to the voltage levels in Eq.(5), the mCRBM system regenerated the 15-step reconstruction in Fig.11(b), agreeing with the training data satisfactorily.

C. Regenerating Three-Dimensional Data with a Symmetric Distribution

To demonstrate the difference from the CRBM system with only two visible neurons in [6], the mCRBM system was programmed to regenerate the three-dimensional (3D) data in Fig.12(a), consisting of two clusters of 200 Gaussian-distributed points. A CRBM with three visible and four hidden neurons was trained to model the data for 4000 epochs. The learnt parameters after slight adjustment were

$$\{w_{ij}\} = \begin{bmatrix} \times & 1.5010 & 1.4915 & 1.5021 & 1.5107 \\ 1.5102 & 1.4920 & 2.1231 & 1.4897 & 1.4898 \\ 1.4982 & 1.5021 & 0.9892 & 1.5101 & 1.4953 \\ 1.5001 & 1.4997 & 2.0012 & 1.5003 & 1.5201 \end{bmatrix} \text{(Volts)}$$

$$\{a_{vi}\} = \begin{bmatrix} 1.7 & 1.75 & 1.70 \end{bmatrix} \text{ (Volts)}$$

$$\{a_{hi}\} = \begin{bmatrix} 1.10 & 1.10 & 2.2 & 1.10 \end{bmatrix} \text{ (Volts)} \qquad (6)$$

With $\{w_{ij}\}$ and $\{a_i\}$ refreshed to the voltage levels in Eq.(6), the mCRBM system regenerated the 15-step reconstruction in Fig.12(b). The similarity between Fig.12(a) and (b) demonstrated again the mCRBM system's ability to regenerate the training data satisfactorily, by the use of noise-induced stochasticity in VLSI.

VI. Classifying Artificial, Three-Dimensional Data

To exploit the scalability of the mCRBM system, two mCRBM chips were interconnected to form a CRBM model with a single-layer perceptron (SLP). Fig.13(a) showed the network architecture, and Fig.13(b) the connection between the two chips. One chip was programmed to model the 3D data in Fig.12, while the other provides a visible neuron to function as the SLP. The SLP was connected to the hidden neurons of the CRBM, classifying data according to the outputs of the hidden neurons. The unused synapses had w_{ij} = 1.5V, and the unused neurons (the blank circles in Fig.13(b)) had N=0. V0 and H0 with N=0 functioned as the biasing neurons whose outputs were set to be 2V (i.e. $s_i = 1$) constantly via their terminal S.

The SLP was trained in Matlab to classify the responses of hidden neurons as the data in Fig.12(a) were clamped to the visible neurons. By back-propagation learning, the weight vector was obtained as $\mathbf{w_S}$ **=** [1.5, 2.25, 1.5, 1.5] (V). With V_{asi} = 2V for the SLP, $\{w_{ij}\}$ of the SLP refreshed to $\mathbf{w_S}$**,** and $\{w_{ij}\}$ and $\{a_{si}\}$ of the CRBM refreshed to Eq.(6), the responses of the full system to the testing data in Fig.14(a) was measured. The testing data contained 5 points for the cluster A and 495 points for the cluster B. All of the 500 testing data were mixed together and presented to the visible neurons one-by-one at a sampling rate of 500Hz. As the noise for all neurons were turned off, the measured output of the SLP was shown in Fig.14(b). The five spikes peaking at 2V corresponded to the five data of the cluster A, while all other data resulted in an output around 1V. This result demonstrated that the mCRBM system was able to classify the 3D data reliably and in real time. The reliability came from the use of noise to model the data variance. Once the noise was removed during classification, the hidden neurons (and thus the SLP) responded to data from the same cluster with negligible differences. In addition, by the merit of analogue circuits, the classification was achieved in real time.

VII. Modelling and CLASSIFYING REAL BIOMEDICAL DATA

On-line spike sorting has been demanded by many brain-machine interfaces [8][21], in order to reduce data size for wireless transmission or to control prosthetic devices in real time. The ability of the mCRBM system to model and to classify neuronal spikes was thus tested.

The data were collected by a 16-channel microelectrode array implanted in the layer V of the primary motor cortex (M1) of an awake rat [12]. The rat was free to move in a box during recording. A multi-channel processor (MAP, Plexon) was then used to record neuronal activity at 40kHz/channel with bandpass filters (450-5kHz). The recorded signals were normalised into [-1, 1] to fit the dynamic range of the CRBM neuron. Fig.15 showed the three types of spikes (labeled as A, B, and C) recorded by a single channel, corresponding to the activity of different neurons in affinity to the recording electrode. A total of 500 spikes with equal number for each type was divided into a training dataset of 100 spikes and a testing dataset of 400 spikes. To reduce hardware complexity, the 64-dimensional spikes were down-sampled to 9-dimensional spikes, as shown by the black lines with squares in Fig.15. A CRBM with nine visible and two hidden neurons was then trained on the dataset in Matlab. After 60000 training epochs , the learnt parameters were given as

$$\{w_{ij}\} = \begin{bmatrix} \times & 1 & 1.5 \\ 1.5 & 1.5 & 1.5 \\ 1.6 & 1.7 & 1 \\ 2.225 & 1.65 & 1.375 \\ 1.55 & 1.7 & 1.8 \\ 1.05 & 1.1 & 1.6 \\ 1.1 & 1.1 & 1.5 \\ 1.2 & 1.25 & 1.5 \\ 1.4 & 1.4 & 1.5 \\ 1.6 & 1.6 & 1.5 \end{bmatrix} \text{(Volts)}$$

$$\{a_{vi}\} = \begin{bmatrix} 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 \end{bmatrix} \text{(Volts)} \tag{7}$$

$$\{a_{hi}\} = \begin{bmatrix}1.95 & 1.95 & 1.95\end{bmatrix} \text{ (Volts)}$$

Two mCRBM chips were interconnected to form a mCRBM system with nine visible and two hidden neurons. The network architecture and the chip connection were shown in Fig.16(a) and (b), respectively. The unused synapses had w_{ij} = 1.5V, and the unused neurons (the blank circles in Fig.13(b)) had N=0. With $\{w_{ij}\}$ and $\{a_i\}$ programmed to the voltage levels in Eq.(7), the mCRBM system was able to reconstruct the three types of spikes, as shown by the gray lines with circles in Fig.15. The ability to model real biomedical data with noise-induced stochasticity in VLSI was clearly demonstrated. Furthermore, as the testing dataset was presented to the mCRBM system with noise turned off, the responses of hidden neurons were measured and plotted in Fig.17(a). Different types of spikes resulted in responses in different clusters, and the clusters were easily separable among each other. The separation could be further enhanced by increasing the sigmoid slope of the hidden neurons. For example, as V_{asi} was increased to 2.25V for all hidden neurons, the responses to the testing dataset were shown in Fig.17(b). Compared with Fig.17(a), the variance of each cluster was significantly reduced. With two voltage comparators having their thresholds in the grey region, the responses could be sorted with 100% accuracy easily. Compared to the principle component analysis (PCA) employed in most spike-sorting systems [16][20][28], the mCRBM system had at least comparable performance, while the analogue implementation facilitates real-time spike sorting.

VIII. CONCLUSION

A scalable and programmable CRBM system in VLSI have been designed, fabricated and tested. By the merit of programmability, the mCRBM system is proved capable of regenerating various artificial data, as well as real biomedical data. By the merit of scalability, the mCRBM system is further shown capable of forming a multi-layer network for data classification, or a large CRBM network for modelling high-dimensional data. The promising experiment results demonstrate that, basing on the CRBM algorithm, large-scale computation with noise-induced stochasticity in VLSI is feasible. In addition, the advantage of probabilistic modelling in VLSI has been demonstrated as the reliability of classifying artificial data and sorting neuronal spikes. The implementation with analogue VLSI further facilitates real-time classification and avoids the need for data converters. These features make the mCRBM system attractive for preprocessing noisy and high-dimensional data in many biomedical applications. The slight imperfection is that parameters mapped into VLSI still require adjustments to compensate for hardware nonlinearity. This problem could be solved by training the mCRBM system with a chip-in-a-loop configuration, as will be tested in the near future. Moreover, the possibility of using the intrinsic noise of transistors to realise the CRBM system will be explored. The exploration would suggest the feasibility of using instead of suppressing noise for computation in the deep-sub-micron era.

[References]

[1] Cauwenberghs, G., "An Analog VLSI Recurrent Neural Network Learning a Continuous-Time Trajectory," *IEEE Transactions on Neural Networks*, vol. 7, no. 2 , pp. 346-361, 1996.

[2] Cauwenberghs, G., "Delta-sigma cellular automata for analog VLSI random vector generation," *Circuit and System II: Analog and Digital Signal Processing, IEEE Transaction on*, vol. 46, no.3, pp. 240-250, 1999.

[3] Cauwenberghs, G., "An Analog VLSI Recurrent Neural Network Learning a Continuous-Time Trajectory," *IEEE Transactions on Neural Networks*, vol. 7, no. 2, pp. 346-361, Mar.2003.

[4] G. Cauwenberghs and M. Bayoumi, Eds., *Learning on Silicon-Adaptive VLSI Neural Systems*. Norwell MA: Kluwer Academic, 1999.

[5] Chen, H., Fleury, P., and Murray, A.F., "Minimizing Contrastive Divergence in Noisy, Mixed-mode VLSI Neurons," *in Advances in Neural Information Processing Systems (NIPS*2003).*

[6] Chen, H., Fleury, P., and Murray, A. F., "Continuous-Valued Probabilistic Behaviour in a VLSI Generative Model," *IEEE Transactions on Neural Networks*, vol. 17, no. 3, pp. 755-770, 2006.

[7] Chen, H. and Murray, A. F., "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.

[8] Donoghue, J. P., "Bridging The Brain to The World: A Perspective on Neural Interface Systems," *Neuron*, Vol. 60, Issue 3, pp. 511-521, 2008.

[9] Erik. A. Johannessen, Lei Wang, Cathy Wyse, David R. S. Cumming, and Jon M. A. Cooper, "Biocompatibility of a Lab-on-a-Pill Sensor in Artificial Gastrointestinal Environments," *Biomedical Engineering, IEEE Transactions on*, vol. 53, no. 11, pp. 2333-2340, 2006.

[10] Genov, R. and Cauwenberghs, G., "Kerneltron: support vector machine in silicon," *IEEE Transactions on Neural Networks*, vol. 14, no. 8, pp. 1426-1433, 2003.

[11] Hinton, G. E., "Training Products of Experts by Minimizing Contrastive Divergence," *Neural Computation*, vol. 14, no. 8, pp. 1771-1800, Aug.2002.

[12] C. F. Liu, T. Y. Pu, Y. C. Huang, K. F. Lin, Y. Y. Chen, W. T. Huang, Y. J. Chang, and C. H. Chen, "Wireless Systems for the Sorting of Neural Signals Using Onboard Processing," *Biomedical Engineering Symposium on Biosignal, Biosensor, Bioelectronic, and Bioengineering*, 2009.

[13] N. H. Hamid, A. F. Murray, D. Laurenson, S. Roy, and C. Binjie, "Probabilistic Computing with Future *Deep* sub-micrometer Devices: a Modelling Approach," *IEEE International Symposium on Circuits and Systems*, pp. 2510-2513, 2005.

[14] Hsu, D., Bridges, S., Figueroa, M., and Diorio, C., "Adaptive Quantization and Density Estimation in Silicon," *in Advances in Neural Information Processing Systems (NIPS*2002).*

[15] N. H. Hamid, Alan F. Murray and Scott Roy, "Time-Domain Modeling of Low-Frequency Noise in Deep-Submicrometer MOSFET," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 1, pp. 245-257, 2008.

[16] Harrison, R. R., Watkins, P. T., Kier, R. J., Lovejoy, R. O., Black, D. J., Greger, B., and Solzbacher, F. "A Low-Power Integrated Circuit for a Wireless 100-Electrode

Neural Recording System," *Solid-State Circuits, IEEE Journal of,* vol.42, no.1, pp.123-133, Jan. 2007.

[17] Lu, C. C., Hong, C. Y., and Chen, H., "A Scalable and Programmable Architecture for the Continuous Restricted Boltzmann Machine in VLSI," *IEEE International Symposium on Circuits and Systems, 2007.*

[18] C. C. Lu, C. C. Li and H. Chen, "How Robust Is a Probabilistic Neural VLSI System Against Environmental Noise," *The 3rd Inter. Workshop on Artificial Neural Networks for Pattern Recognition*, p 44-53, 2008.

[19] Lebedev, M. A. and Nicolelis, M. A. L., "Brain-machine interfaces: past, present and future," TRENDS in Neuroscience 29 [9], 536-546. 2006.

[20] Moosung Chae, Wentai Liu, Zhi Yang, Tungchien Chen, Jungsuk Kim, Sivaprakasam, M., and Yuce, M. "A 128-Channel 6mW Wireless Neural Recording IC with On-the-Fly Spike Sorting and UWB Tansmitter," *Solid-State Circuits Conference, 2008. ISSCC 2008. Digest of Technical Papers. IEEE International*, pp.146-603, 3-7 Feb. 2008.

[21] Olsson, R H , III and Wise, K. D., "A three-dimensional neural recording microsystem with implantable data compression circuitry," *IEEE J. Solid-State Circuits*, vol. 40, no. 12, pp. 2796-2804, 2005.

[22] Reid, M. S., Brown, E. A. and DeWeerth, S. P., "Subthreshold CMOS Array for Generating a Gaussian Distribution of Currents," *Circuit and System II: Analog and Digital Signal Processing, IEEE Transaction on*, vol. 53, no.10, pp. 1057-7130, 2006.

[23] Salinas, E., “Noisy Neurons Can Certainly Compute," *Nature Neuroscience*, vol.9, no.11, pp.1349-1350, 2006.

[24] Schwartz, A. B., "Cortical Neural Prothetics," *Annual Review Neuroscience*, pp. 487-507, Mar.2004.

[25] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd Edition, Prentice-Hall, 1999.

[26] Tang, T. B., Johannessen, E., Wang, L., Astaras, A., Ahmadian, M., Murray, A. F., Cooper, J. M., Beaumont, S. P., Flynn, B. W., and Cumming, D. R. S., "Towards a Miniature Wireless Integrated Multisensor Microsystem for Industrial and Biomedical Applications," *IEEE Sensors Journal: Special Issue on Integrated Multisensor Systems and Signal Processing*, vol. 2, no. 6, pp. 628-635, 2002.

[27] Wise, K. D., Anderson, D. J., Hetke, J. F., D.R., K., and Najafi, K., "Wireless Implantable Microsystems: High-Density Electronic Interfaces to the Nervous System," *Proceedings of the IEEE*, vol. 92, no. 1, pp. 76-96, Jan.2004.

[28] Z. S. Zumsteg, C. Kemere, S. O'Driscoll, G. Santhanam, R. E. Ahmed, K. V. Shenoy, and T. H. Meng, "Power feasibility of implantable digital spike sorting circuits for neural prosthetic systems," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 13, no. 3, pp. 272-279, 2005.

[Table Captions]

TABLE I: The mapping of parameters between Matlab simulation and VLSI implementation.

[Figure Captions]

Fig.1: The architecture of a CRBM model with four visible and four hidden neurons. v_0 and h_0 represent biasing units with constant outputs v_0=h_0=1.

Fig.2: The architecture of a scalable and programmable CRBM system and its functional modules.

Fig.3: The chip photo of the scalable and programmable CRBM system

Fig.4: The circuit diagram of a continuous-valued stochastic neuron of the CRBM

Fig.5: (a) The measured DC characteristic of a sigmoid circuit with different V_{asi}. (b) The mapping between a_i and V_{asi}.

Fig.6: (a) The scatter plot of the noise voltages measured from channel A and J of the noise generator. (b) Statistical histograms of the noise amplitudes recorded from channel A. (c) The mapping between σ and V_{sig}.

Fig.7: The measured output (V_{si}) of a continuous-valued, stochastic neuron (upper trace) and the clock (lower trace) that samples V_{si}.

Fig.8: (a) The architecture for multiplexing and programming the parameters of the mCRBM system. (b) The digital-control signals. (c) The updating circuit.

Fig.9: (a) The measured updating stepsize of 12mV with VP=2.46V, VN=0.57V, and a pulse width of 320ns. (b) The measured programming process of the parameters in the first column (w00, w10, w20, w30, and w40).

Fig.10: (a) The training data with a symmetric distribution. (b) 400 data points reconstructed by the mCRBM system with parameters refreshed to Eq.(4)..

Fig.11: (a) The training data with a non-symmetric distribution. (b) 400 data points reconstructed by the mCRBM system with parameters refreshed to Eq.(5).

Fig.12: (a) The three-dimensional training data. (b) 400 data points reconstructed by the mCRBM system with parameters refreshed to Eq.(6).

Fig.13: (a) The architechture and (b) the chip connection of a CRBM with a SLP for classifying the data in Fig.14a.

Fig.14: (a) Three-dimensional testing dataset for classification. (b) The measured output of the SLP in response to the 500 testing data.

Fig.15: (Upper)**:** The recorded and normalised neuronal spikes with 64 dimensions. (Lower): The down-sampled spike waveform (black lines with squares) and the spike waveform reconstructed by the mCRBM system with parameters refreshed to Eq.(7). (gray lines with circles)

Fig.16: (a) The architechture and (b) the chip connection of a CRBM for modelling and classifying the spikes in Fig.15

Fig.17: The measure responses of the hidden neuron H1 and H2 to the testing dataset as (a) $\{a_{hi}\}$ = 1.95V (b) $\{a_{hi}\}$ = 2.25V.

[Tables]

TABLE I

	MATLAB	VLSI	UNIT VALUE
s_i	[-1.0, 1.0]	[1.0, 2.0] (V)	0.5 (V)
w_{ij}	[-3.0, 3.0]	[0.0, 3.0] (V)	0.5 (V)
i_{in}	[-15, 15]	[-15, 15] (μA)	1μA
a_i	[0.5, 9.0]	[0.5, 2.4] (V)	Fig.5(b)
σ	[0.1, 0.35]	[0.53, 0.64] (V)	Fig.6(c)

[Figures]

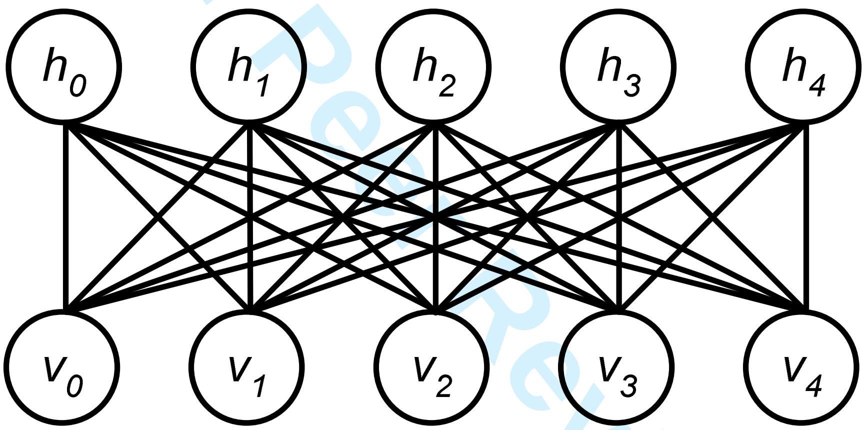


Fig.1

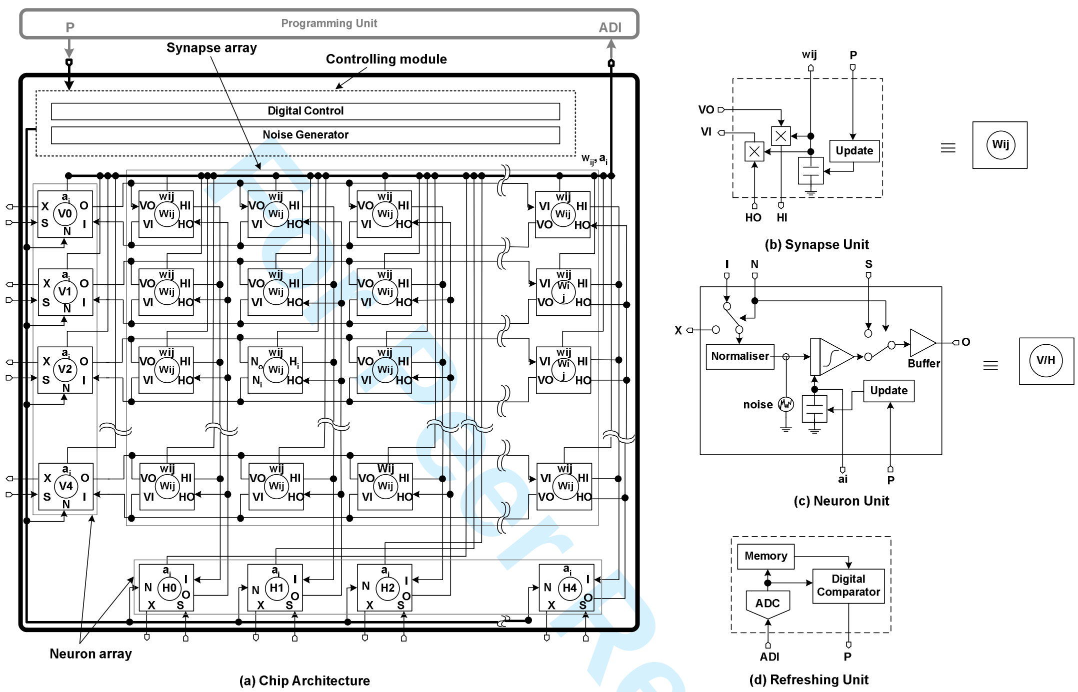

Programming Unit
P
ADI
Synapse array
Controlling module
Digital Control
Noise Generator
Neuron array
(a) Chip Architecture
wij
P
VO
VI
Update
HO
HI
Wij
(b) Synapse Unit
I
N
S
X
Normaliser
noise
Buffer
Update
ai
P
O
V/H
(c) Neuron Unit
Memory
Digital Comparator
ADC
ADI
P
(d) Refreshing Unit

Fig.2

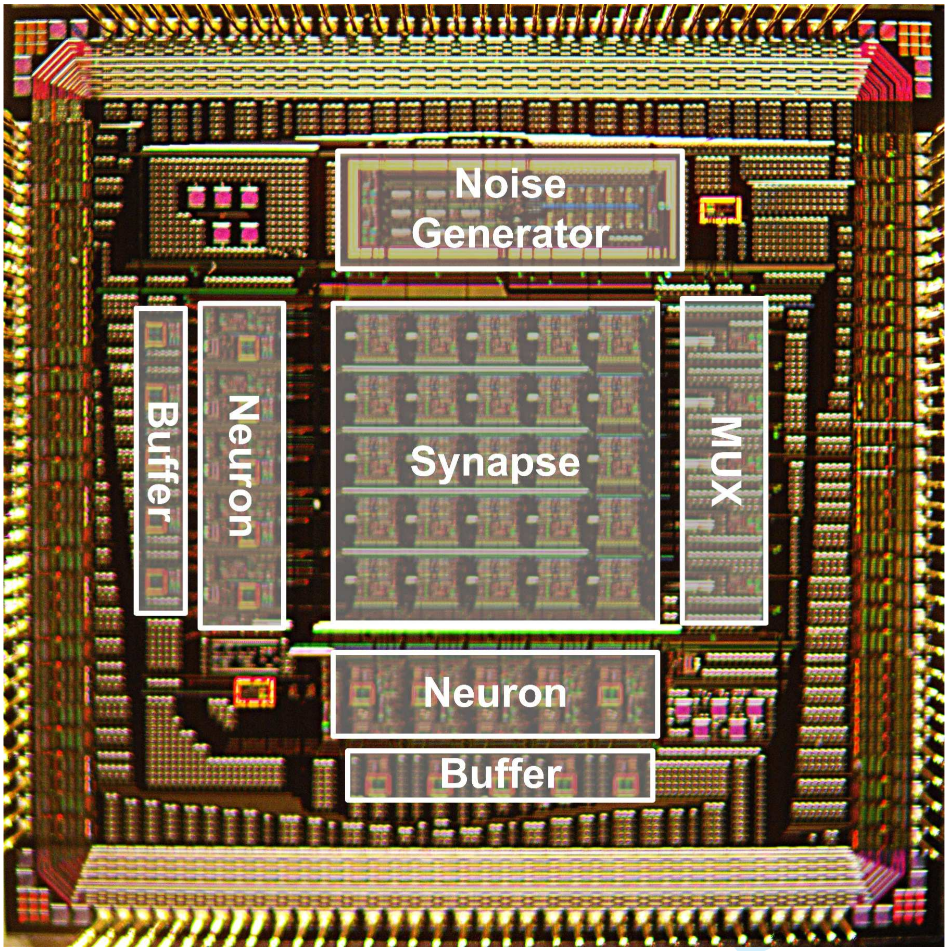


Fig.3

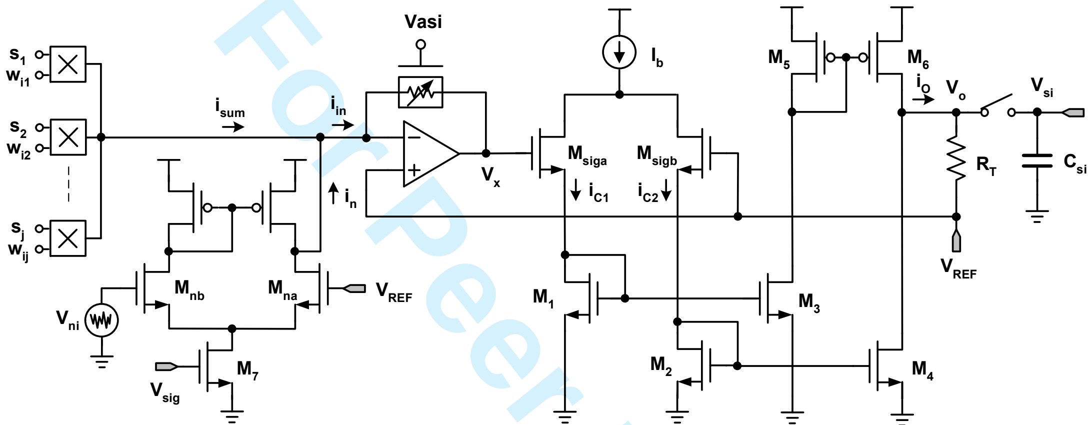
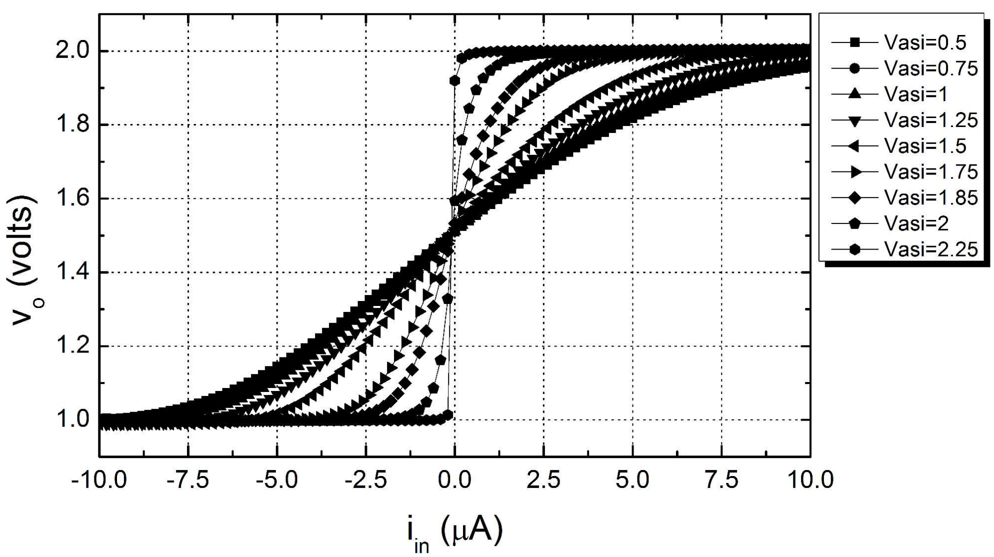

Vasi
Ib
M5
M6
s1
wi1
s2
wi2
sj
wij
isum
iin
in
Vx
Msiga
Msigb
iC1
iC2
iO
Vo
Vsi
RT
Csi
VREF
Mnb
Mna
VREF
Vni
M1
M3
M2
M4
M7
Vsig

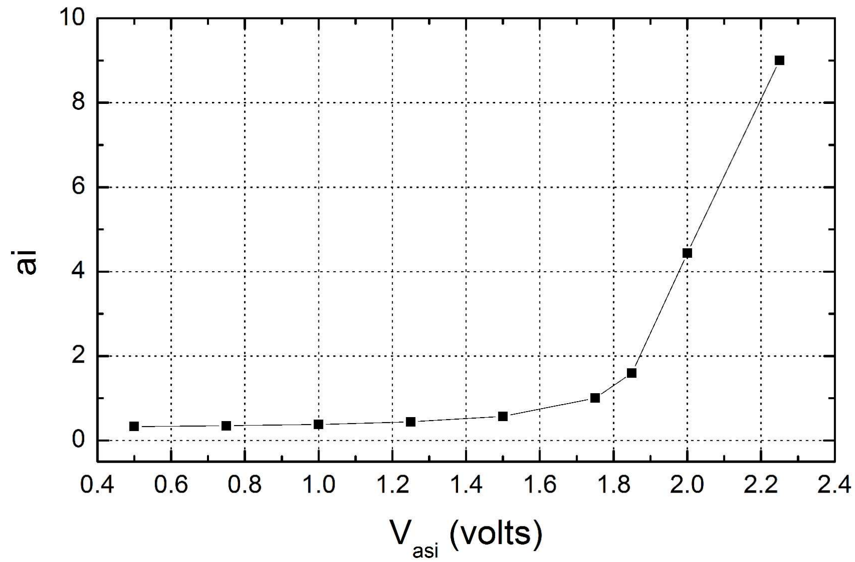
Fig.4



2.0
1.8
1.6
1.4
1.2
1.0
-10.0
-7.5
-5.0
-2.5
0.0
2.5
5.0
7.5
10.0
v_o (volts)
i_{in} (μA)
Vasi=0.5
Vasi=0.75
Vasi=1
Vasi=1.25
Vasi=1.5
Vasi=1.75
Vasi=1.85
Vasi=2
Vasi=2.25

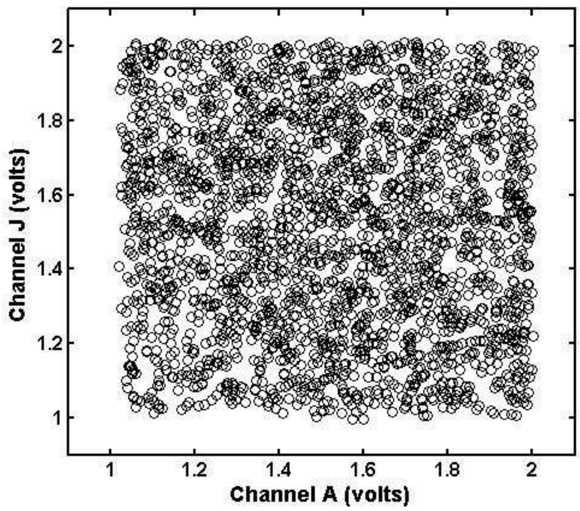
(a)

Fig.5



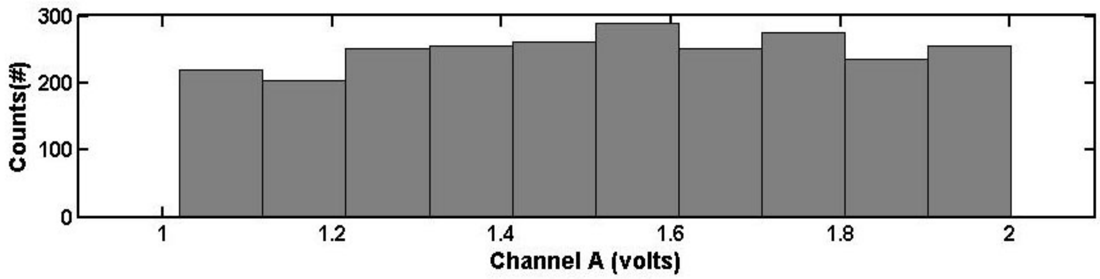
(b)

Fig.5

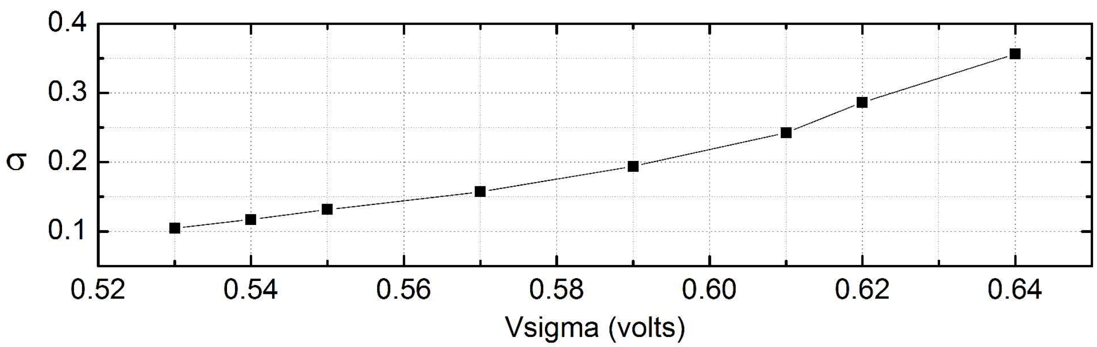


(a)

Fig.6


Counts(#)
300
200
100
0
1
1.2
1.4
1.6
1.8
2
Channel A (volts)

(b)


σ
0.4
0.3
0.2
0.1
0.52
0.54
0.56
0.58
0.60
0.62
0.64
Vsigma (volts)

(c)

Fig.6

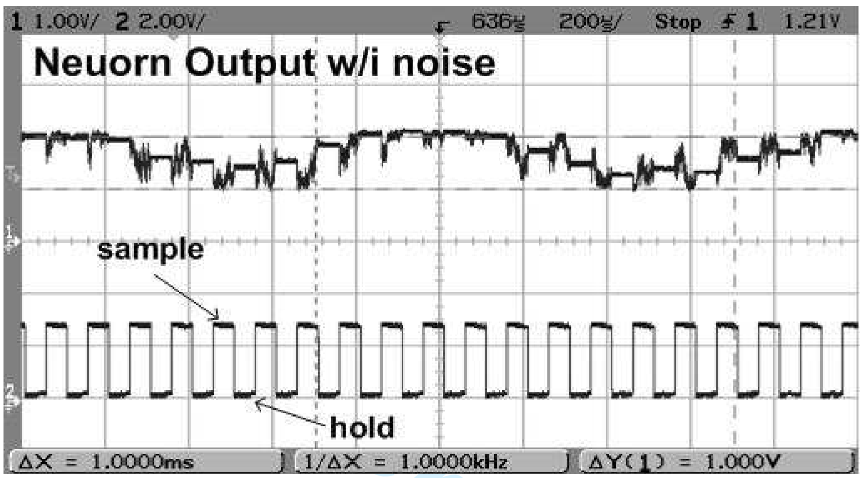
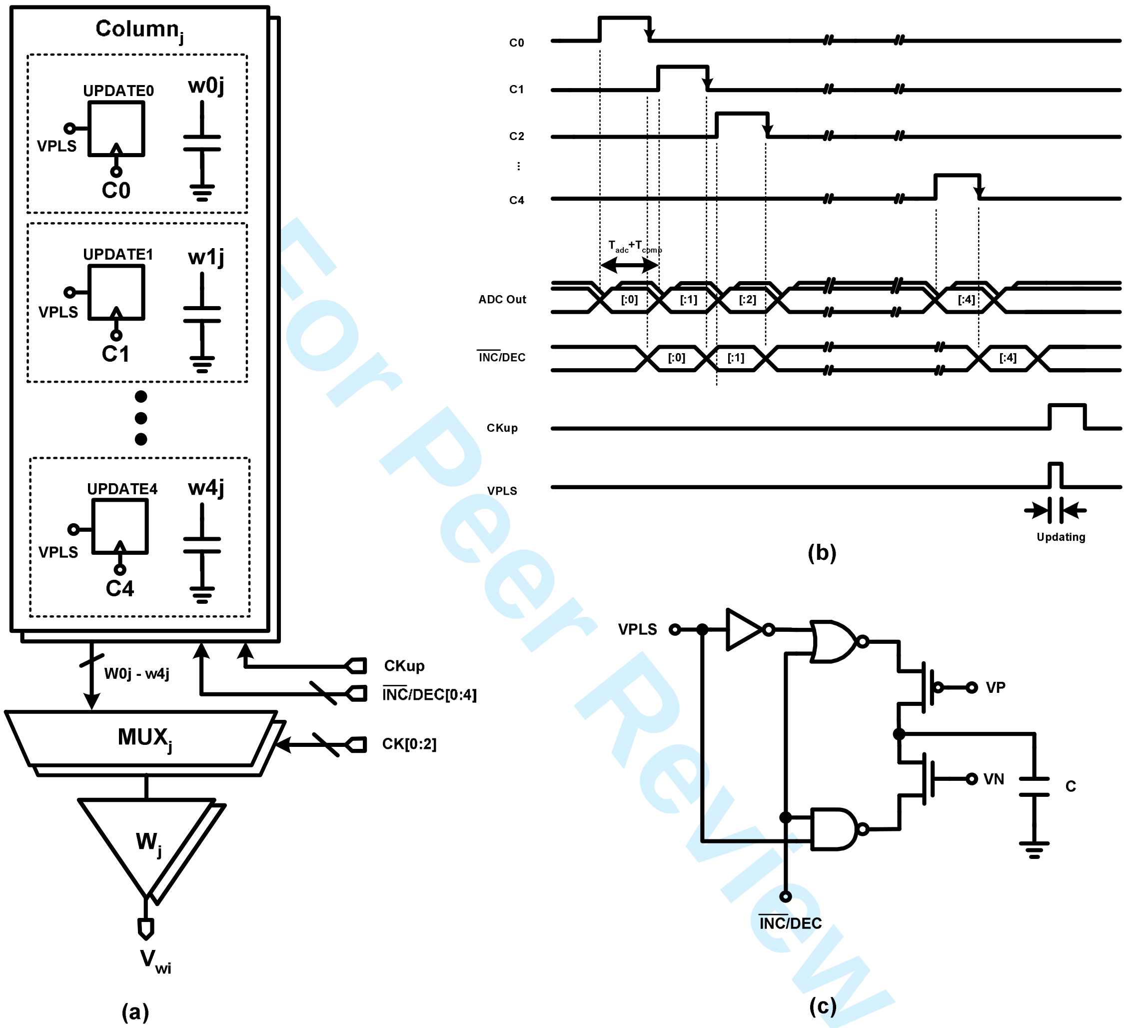
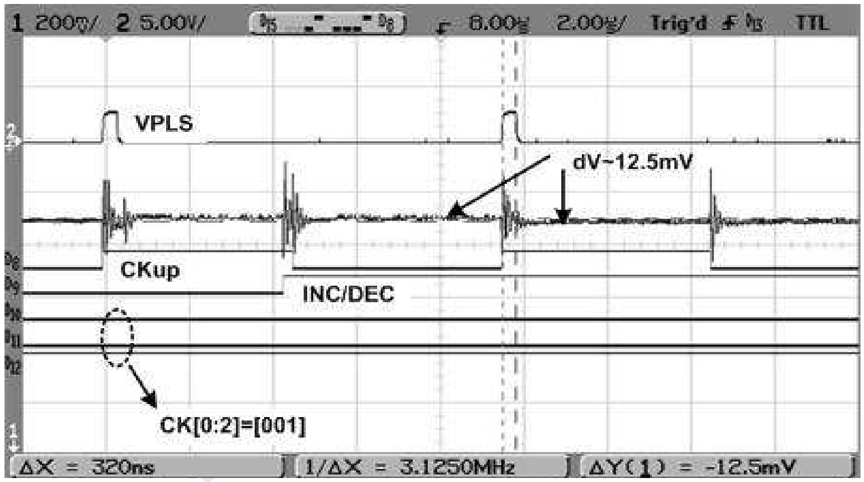

1 1.00V/ 2 2.00V/
636$
200$/
Stop
1
1.21V
Neuorn Output w/i noise
sample
hold
ΔX = 1.0000ms
1/ΔX = 1.0000kHz
ΔY(1) = 1.000V

Fig. 7

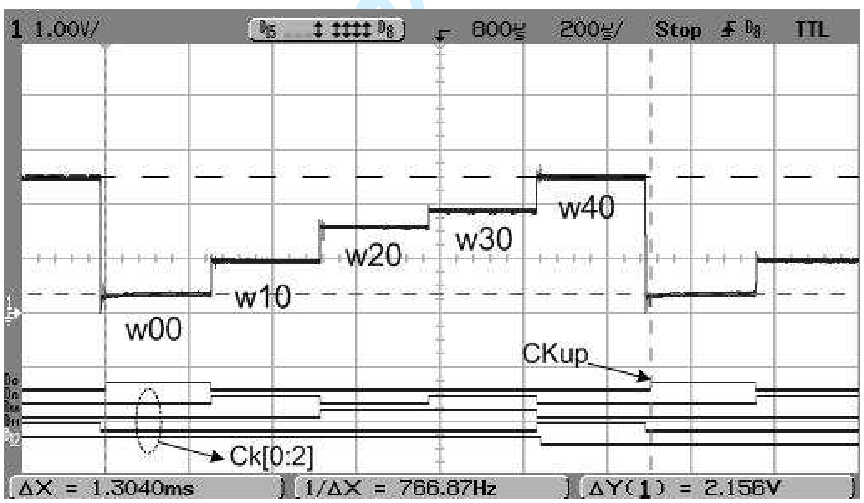


Columnj
UPDATE0
w0j
VPLS
C0
UPDATE1
w1j
VPLS
C1
UPDATE4
w4j
VPLS
C4
W0j - w4j
CKup
INC/DEC[0:4]
MUXj
CK[0:2]
Wj
Vwi
(a)
C0
C1
C2
C4
Tadc+Tcomp
ADC Out
[:0]
[:1]
[:2]
[:4]
INC/DEC
[:0]
[:1]
[:4]
CKup
VPLS
Updating
(b)
VPLS
VP
VN
C
INC/DEC
(c)

Fig.8

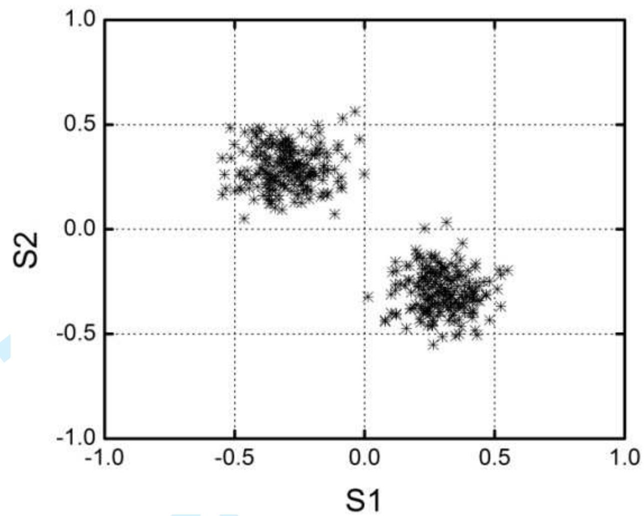

1 200mV/ 2 5.00V/
8.00us 2.00us/ Trig'd TTL
VPLS
dV~12.5mV
CKup
INC/DEC
CK[0:2]=[001]
ΔX = 320ns
1/ΔX = 3.1250MHz
ΔY(1) = -12.5mV

(a)

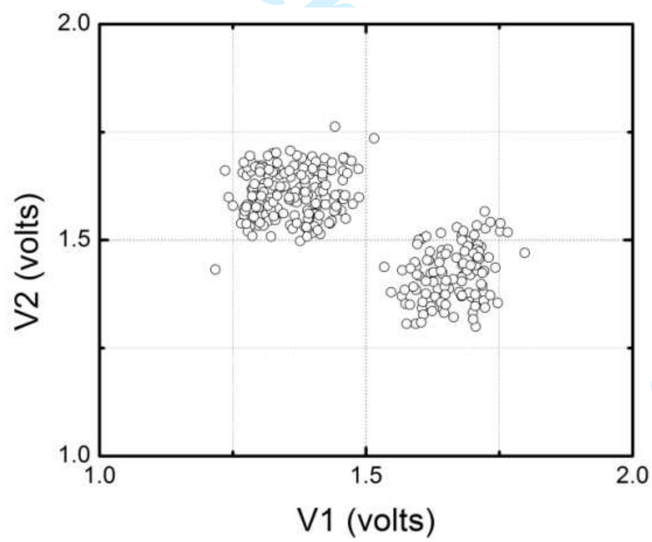

1 1.00V/
800us 200us/ Stop TTL
w40
w30
w20
w10
w00
CKup
Ck[0:2]
ΔX = 1.3040ms
1/ΔX = 766.87Hz
ΔY(1) = 2.156V

(b)

Fig.9

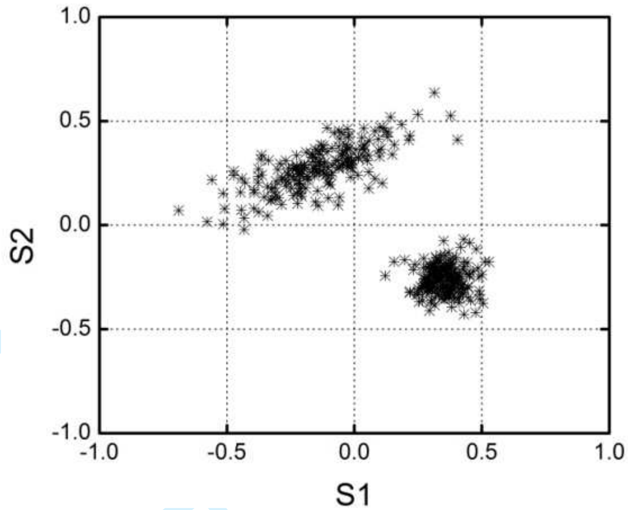

1.0
0.5
0.0
-0.5
-1.0
S2
-1.0
-0.5
0.0
0.5
1.0
S1

(a)

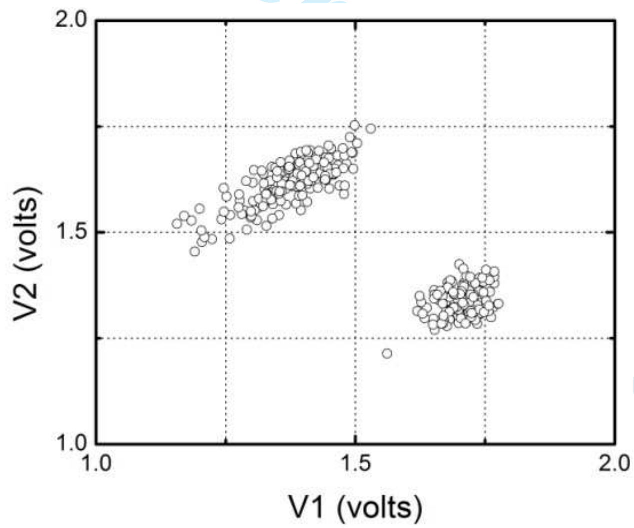

2.0
1.5
1.0
V2 (volts)
1.0
1.5
2.0
V1 (volts)

(b)

Fig. 10

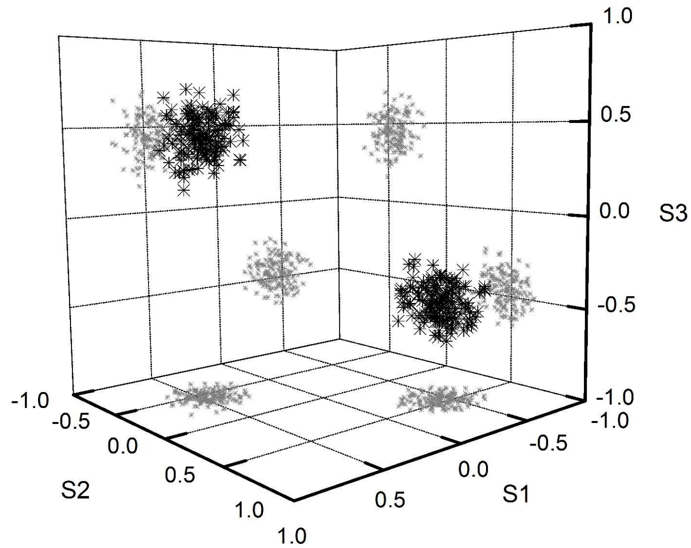

1.0
0.5
0.0
-0.5
-1.0
S2
-1.0
-0.5
0.0
0.5
1.0
S1

(a)

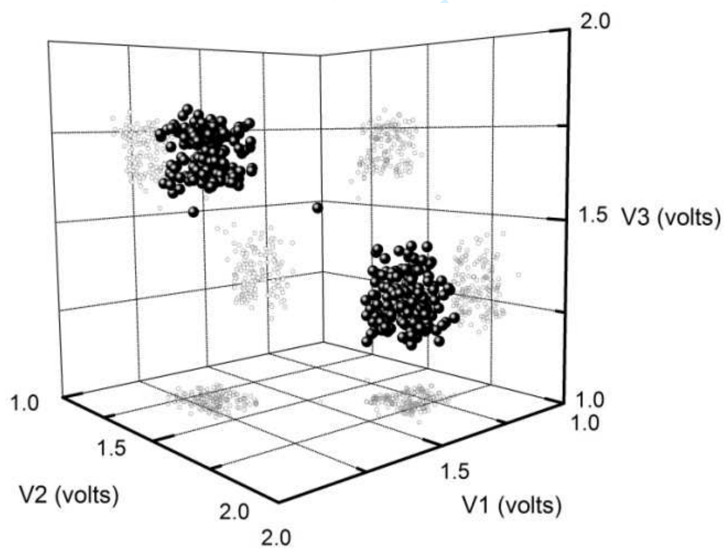

2.0
1.5
1.0
V2 (volts)
1.0
1.5
2.0
V1 (volts)

(b)

Fig. 11


1.0
0.5
0.0
S3
-0.5
-1.0
-1.0
-0.5
0.0
0.5
1.0
S2
-1.0
-0.5
0.0
0.5
1.0
S1

(a)


2.0
1.5
V3 (volts)
1.0
1.0
1.5
2.0
V2 (volts)
1.0
1.5
2.0
V1 (volts)

(b)

Fig. 12

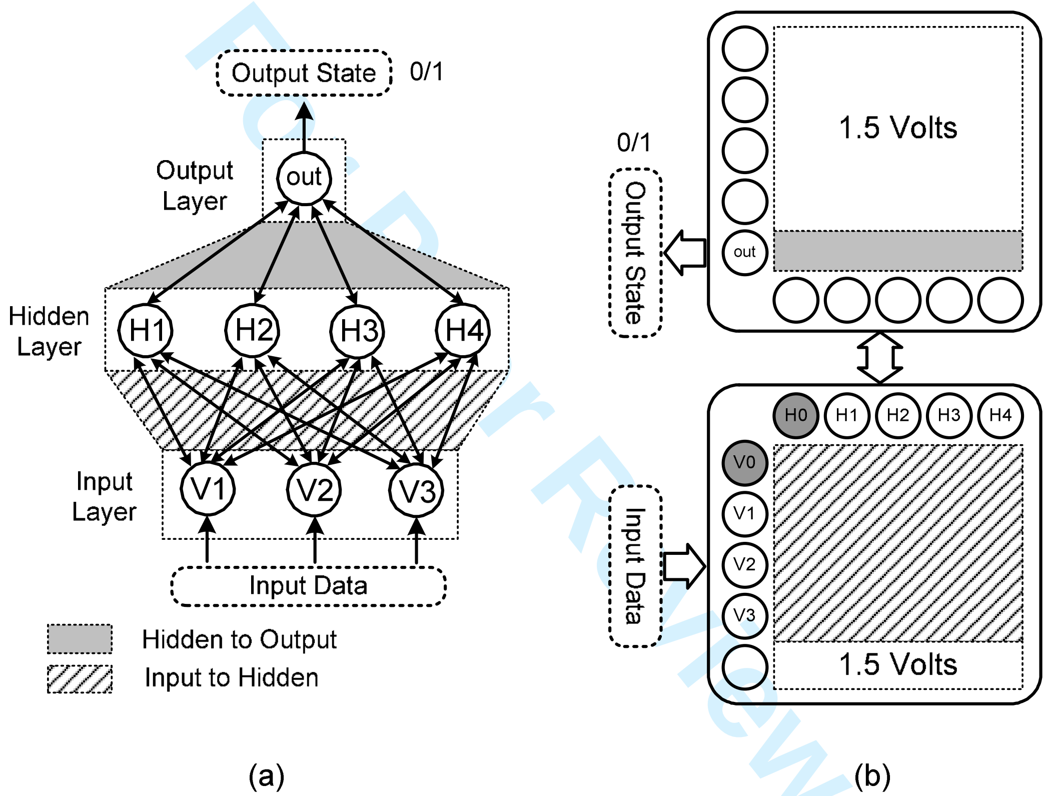
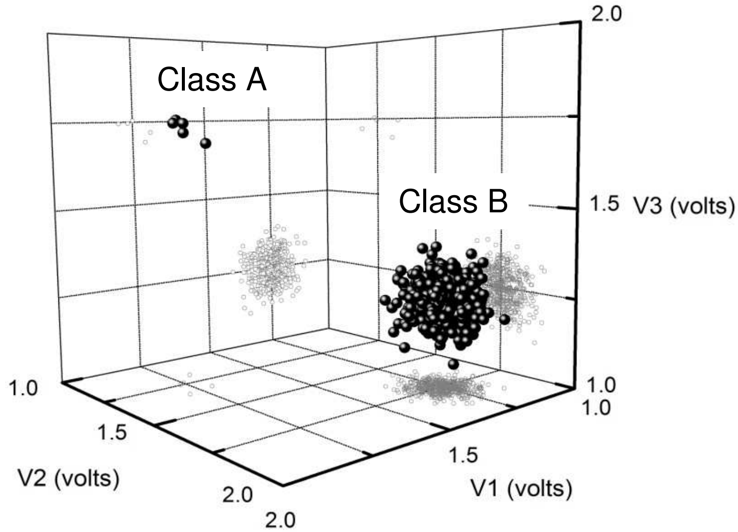
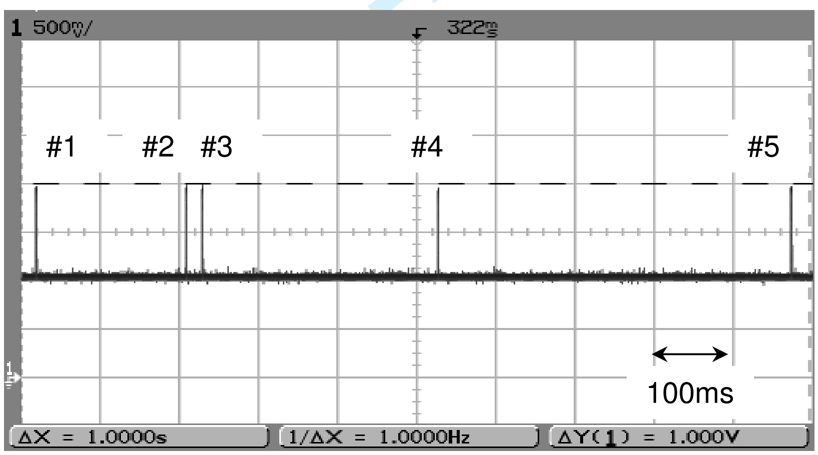

Output State
0/1
Output
Layer
out
Hidden
Layer
H1
H2
H3
H4
Input
Layer
V1
V2
V3
Input Data
Hidden to Output
Input to Hidden
(a)
1.5 Volts
0/1
Output State
out
H0
H1
H2
H3
H4
V0
V1
V2
V3
Input Data
1.5 Volts
(b)

Fig. 13



Class A
Class B
2.0
1.5
V3 (volts)
1.0
1.0
1.5
V2 (volts)
2.0
1.0
1.5
V1 (volts)
2.0

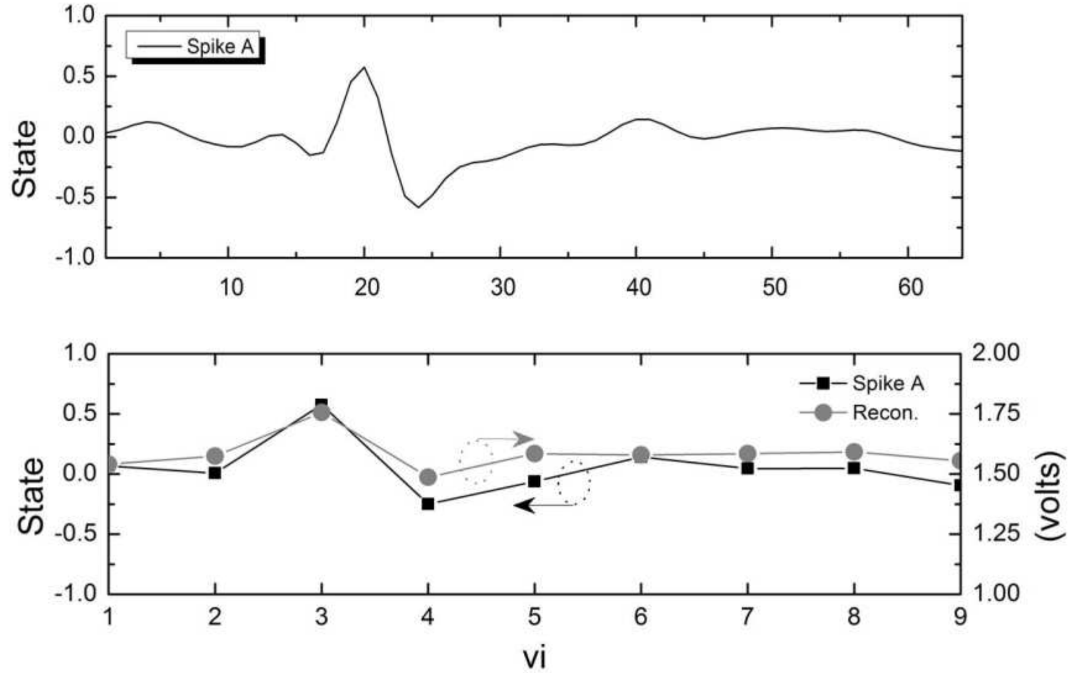
(a)



1 500mV/
322ms
#1
#2
#3
#4
#5
100ms
ΔX = 1.0000s
1/ΔX = 1.0000Hz
ΔY(1) = 1.000V

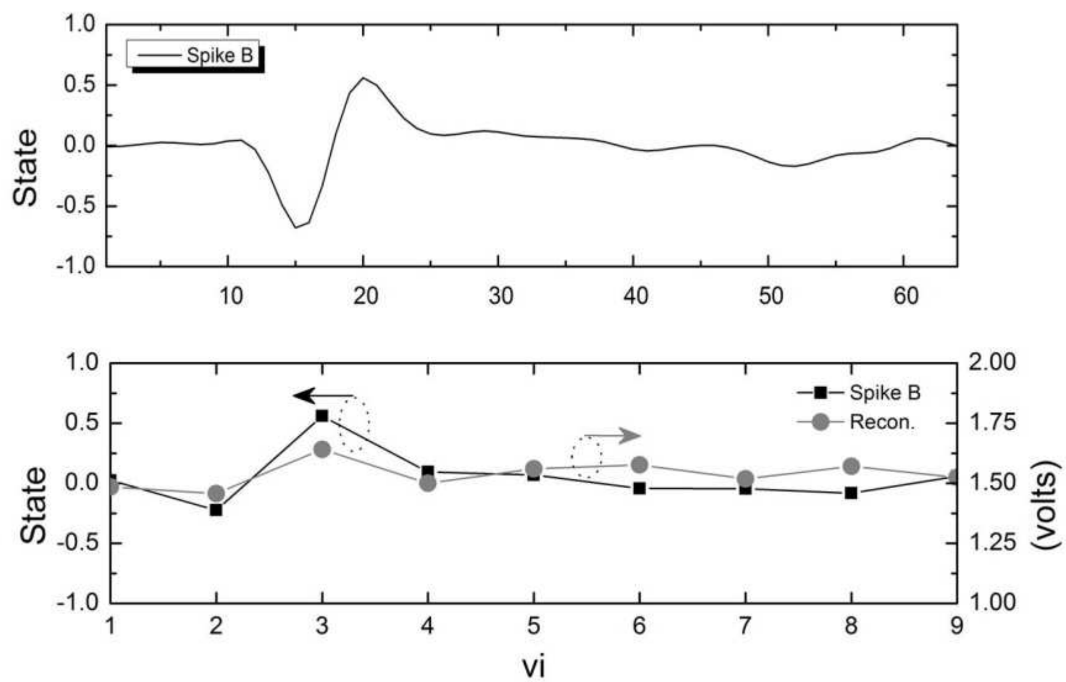
(b)

Fig. 14


Spike A
State
1.0
0.5
0.0
-0.5
-1.0
10
20
30
40
50
60
Spike A
Recon.
2.00
1.75
1.50
1.25
1.00
(volts)
1
2
3
4
5
6
7
8
9
vi

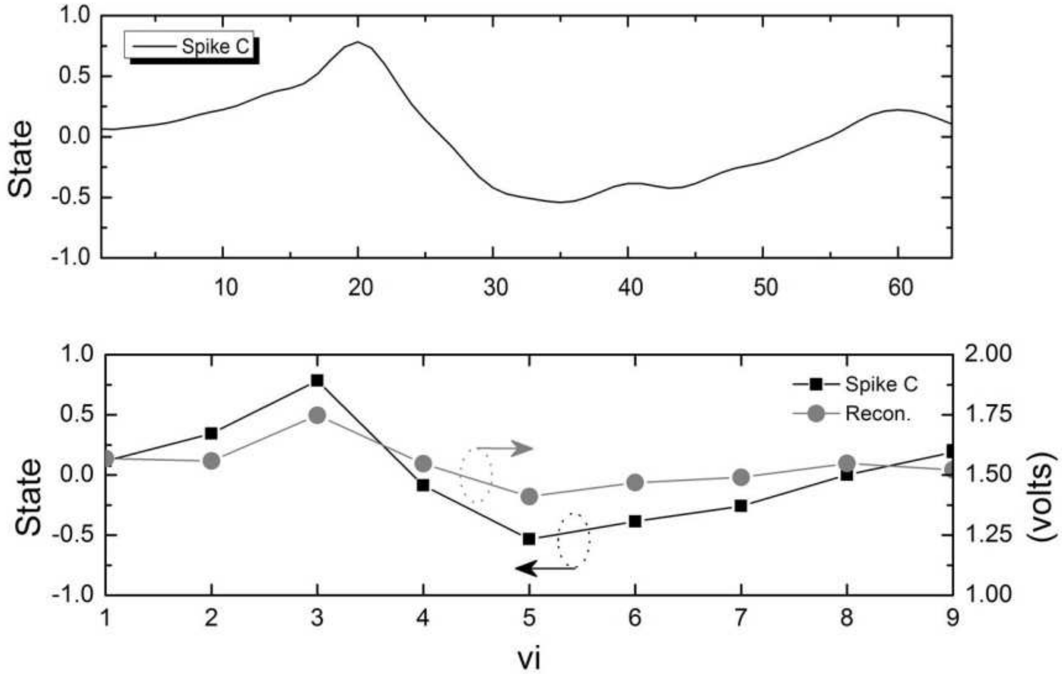
(a)

Fig. 15


Spike B
State
1.0
0.5
0.0
-0.5
-1.0
10
20
30
40
50
60
Spike B
Recon.
State
(volts)
2.00
1.75
1.50
1.25
1.00
1
2
3
4
5
6
7
8
9
vi

(b)

Fig. 15



Spike C
State
1.0
0.5
0.0
-0.5
-1.0
10
20
30
40
50
60
Spike C
Recon.
State
(volts)
2.00
1.75
1.50
1.25
1.00
1
2
3
4
5
6
7
8
9
vi

(c)

Fig. 15

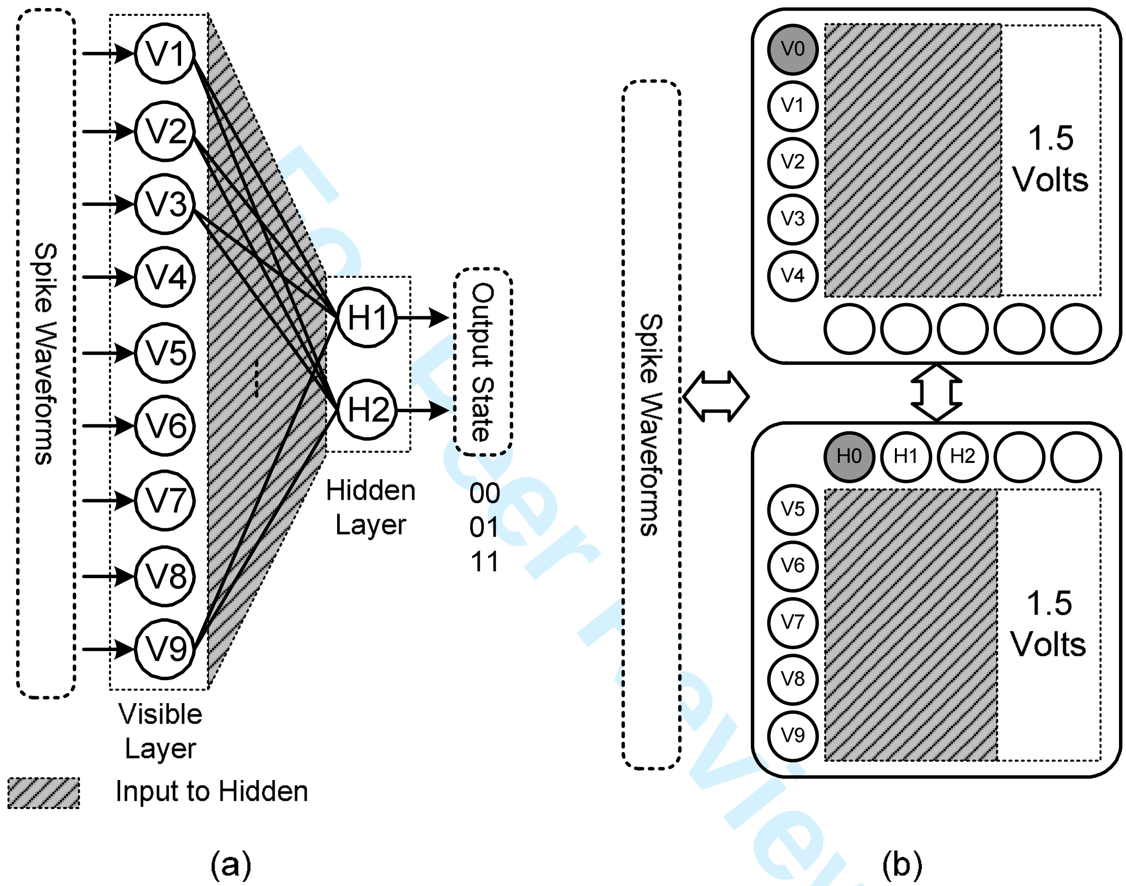
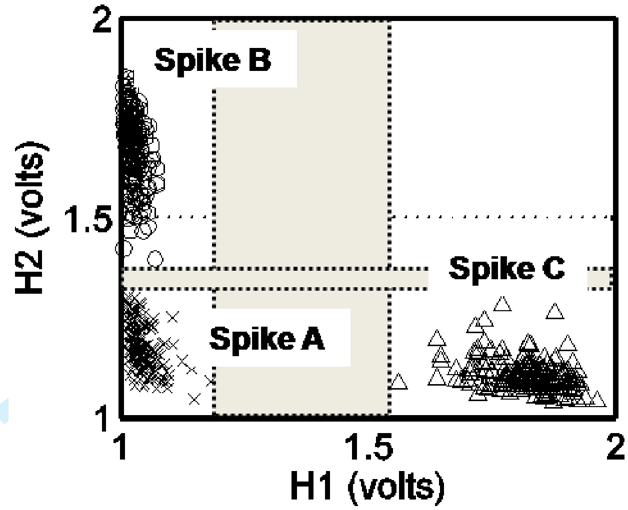
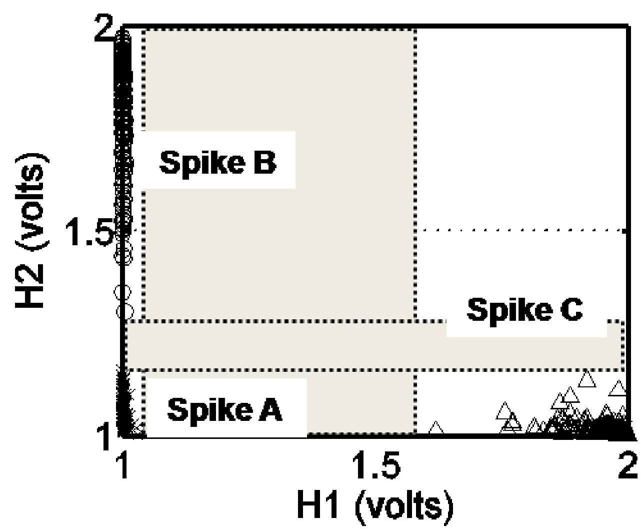

Spike Waveforms
V1
V2
V3
V4
V5
V6
V7
V8
V9
H1
H2
Output State
Hidden Layer
00
01
11
Visible Layer
Input to Hidden
(a)
Spike Waveforms
V0
V1
V2
V3
V4
1.5 Volts
H0
H1
H2
V5
V6
V7
V8
V9
1.5 Volts
(b)

Fig. 16


Spike B
Spike C
Spike A
H2 (volts)
H1 (volts)
2
1.5
1
1
1.5
2

(a)


Spike B
Spike C
Spike A
H2 (volts)
H1 (volts)
2
1.5
1
1
1.5
2

(b)

Fig. 17