Can deep-sub-micron device noise be used as the basis for probabilistic neural computation?

Nor Hisham Hamid

A thesis submitted for the degree of Doctor of Philosophy.
The University of Edinburgh.
March 2006

Abstract

This thesis explores the potential of probabilistic neural architectures for computation with future nanoscale Metal-Oxide-Semiconductor Field Effect Transistors (MOSFETs). In particular, the performance of a Continuous Restricted Boltzmann Machine (CRBM) implemented with generated noise of Random Telegraph Signal (RTS) and $1/f$ form has been studied with reference to the 'typical' Gaussian implementation. In this study, a time domain RTS based noise analysis capability has been developed based upon future nanoscale MOSFETs, to represent the effect of nanoscale MOSFET noise on circuit implementation in particular the synaptic analogue multiplier which is subsequently used to implement stochastic behaviour of the CRBM. The result of this thesis indicates little degradation in performance from that of the typical Gaussian CRBM. Through simulation experiments, the CRBM with nanoscale MOSFET noise shows the ability to reconstruct training data, although it takes longer to converge to equilibrium. The results in this thesis do not prove that nanoscale MOSFET noise can be exploited in all contexts and with all data, for probabilistic computation. However, the result indicates, for the first time, that nanoscale MOSFET noise has the potential to be used for probabilistic neural computation hardware implementation. This thesis thus introduces a methodology for a form of technology-downstreaming and highlights the potential of probabilistic architecture for computation with future nanoscale MOSFETs.

Declaration of originality

I declare that this thesis has been completed by myself and that, except where indicated to the contrary, the research documented in this thesis is entirely my own.

Nor Hisham Hamid

Acknowledgements

First of all, I would like to thank God, the Almighty, for having made everything possible by giving me the strength and courage to pursue this PhD.

I would like to thank my supervisors Prof. Alan F. Murray and Dr. Martin Reekie for their advise, guidance, and encouragement, throughout the course of this PhD. I am particularly indebted to Prof. Alan Murray, whose expertise, understanding, dedication and patience has greatly contributed toward the completion of this thesis.

I would also like to thank my family for their inspiration, love, moral and financial support, and prayers over these past four years. In particular, I am forever indebted to my wife Mazni, without her love, encouragement, patience, and understanding, I would never have completed this thesis.

I greatly appreciate Dr. David Laurenson and Prof. Steve McLaughlin (IDCOM) of University of Edinburgh, and Prof. Asen Asenov, Dr. Scott Roy, Dr. Jeremy Watling, and all members of Device Modelling Group of University of Glasgow, for their technical support and expertise toward the understanding and implementation of noisy nanoscale MOSFET model.

A very special thank to Dr. Thomas Koickal, Tong Boon, Mark, and Katherine who carefully proof read my papers and thesis, and politely pointed out mistakes. I also would like to thank all SEE computing staff in particular David Stewart for their dedication in ensuring a first class IT support. Finally, I would like to thank all whose direct and indirect support helped me completing this thesis.

I acknowledge the Universiti Teknologi Petronas for providing scholarships to pursue this PhD work.

I dedicate this thesis to my wife, my son Haziq, my daughters Aqeela and Faqeeha.

Contents

List of figures

List of tables

Chapter 1
Introduction

This thesis explores the prospects of using future nanoscale MOSFETs to implement probabilistic computation in hardware, based on modelled future nanoscale MOSFETs and specific probabilistic neural computation architecture. The word 'future' refers to MOSFETs which are not yet available although their performance is predictable through current research efforts, while the word 'nanoscale' refers to MOSFETs with dimensions less than 10nm (sub-10nm) in physical gate length. The motivation of this research is described in Sec.1.1, and the contribution to knowledge is clarified in Sec.1.2. Finally, the structure of this thesis is described in Sec.1.3.

1.1 Motivation

The success of metal-oxide-semiconductor field-effect transistors (MOSFETs) as the basic building block of most digital and analogue very large scale integrated (VLSI) circuits is predicted to continue for some years. Currently, 90nm (with a physical gate length of 50nm) is the state-of-the-art MOSFET process technology and it is projected that by 2018, sub-10nm physical gate length MOSFETs will be available [6]. The drive toward miniaturisation is led by the promise of improved circuit performance, reduced chip sizes, and the potential of higher levels of integration. However, as MOSFET dimensions continue to shrink, considerable challenges arise in the area of device performance and reliability uncertainty [7, 8].

One of the contributing factors towards performance and reliability uncertainty is the increase in low frequency drain current noise. Drain current noise in MOSFETs is predicted to increase as the channel length shrinks [9–12]. Random Telegraph Signal (RTS) noise and $1/f$ noise are the primary forms of low frequency noises that are predicted in future nanoscale MOSFETs. In current technology these are minimised or suppressed through design and/or additional fabrication steps. As MOSFET dimensions continue to shrink, their presence will become increasingly significant. Recent studies show low frequency drain current noise amplitudes in excess of 60% in Deep Sub-Micrometer (DSM) MOSFET[1] [13]. As MOSFETs continue to scale, the low frequency drain current noise in nanoscale MOSFETs[2] is expected to become a serious issue [14], leading to severely limited functionality, performance, and compromised reliability. A conventional solution would avoid or minimise nanoscale MOSFET noise through additional fabrication processes. For large MOSFETs, fabrication processes may be controlled to reduce noise [6]. However, in nanoscale fabrication process, precise fabrication process control may be impossible and the performance gain of nanoscale MOSFETs may not justify the enormous fabrication cost. Furthermore, many of the sources of noise and unreliability in DSM MOSFETs are fundamental and will not yield to improved or more careful processing. Solutions based on alternative architectural paradigms, such that the unreliable performance of these nanoscale MOSFETs could be tolerated or useful, become very attractive. For example, the architectures proposed in [15–17] use redundant circuits to form error correction to deal with this uncertainty. Other approaches are adaptive (neural network and probabilistic computing), forcing errors introduced by these nanoscale MOSFETs noise to adapt to a known (trained) system outcome or acceptable-error marked [18–20]. These fault tolerant architectural approaches provide reliability via redundancy, at the expense of circuit area and speed. An unconventional architectural approach that allows for stochasticity, or that even exploits nanoscale MOSFET noise, is therefore an exciting alternative.

Solving the nanoscale MOSFET noise issue requires something of a paradigm shift, wherein noise is viewed as a necessary element of useful computation. This is different from the approach reported in [16–18] where artificial neural networks are used because of their inherent hardware redundancy, error tolerance and self-organisation features, offering a very effective means to counteract the inherent weaknesses of nanoscale MOSFETs. Naturally, it would be unwise to claim that the proposed approach will solve all conventional computing problems. Rather, it provides an opportunity for new computation architectures such as a probabilistic neural architecture to be implemented in hardware more efficiently, extending

[1] Deep Sub-Micrometer (DSM) refers to MOSFET with physical gate length less than 100nm but greater than 10nm.

[2] Low frequency noise in nanoscale MOSFETs will henceforth be referred to as nanoscale MOSFET noise.

the compatibility of nanoscale MOSFETs to specific real-world application.

The Continuous Restricted Boltzmann Machine (CRBM) aims to deal with real-time data in a noisy environment; potentially for a complex multi-sensory micro-system implementation such as a Lab-On-Chip system [21,22], coincidentally an area where nanoscale technology may be highly desirable for reasons of size. Data often encodes biological or chemical information, of relatively low bandwidth. Probabilistic neural systems are arguably well-positioned to address nanoscale MOSFET noise as they use stochasticity to extract and to classify important features in real-world data. This project explores the use of nanoscale MOSFET noise in a probabilistic neural architecture which has been shown great potential for realising intelligent embedded system [21–24].

The work therefore must build a bridge between future nanoscale MOSFET physics and this probabilistic neural computation, in order to determine whether intrinsic low frequency drain current noise in future nanoscale MOSFETs is useful for probabilistic computation. If unreliable nanoscale MOSFETs can be shown to be useful in such an application, the technological and economic consequences of their practical implementation may become extremely significant. Furthermore, the methodology developed to make this study has more generic usefulness, as will be discussed in this thesis.

1.2 Contribution to knowledge

This project sets out to explore the suggestion that :-

Low frequency drain current noise in future nanoscale MOSFETs can underpin useful probabilistic computation.

In examining this hypothesis, the project will develop new methods to link DSM device physics, through compact circuit models, to behavioural-level simulations of a relatively-well-understood probabilistic paradigm.

The Continuous Restricted Boltzmann Machine (CRBM) has been chosen as an experimental platform. While both nanoscale MOSFET noise [10–12,25–29] and CRBM [3,21–24,30–

32] have been the subject of extensive research, the use of nanoscale MOSFET noise for computation in the CRBM has not been studied, and it is hoped that this project will point the way towards hardware implementations of nano-embedded intelligent systems.

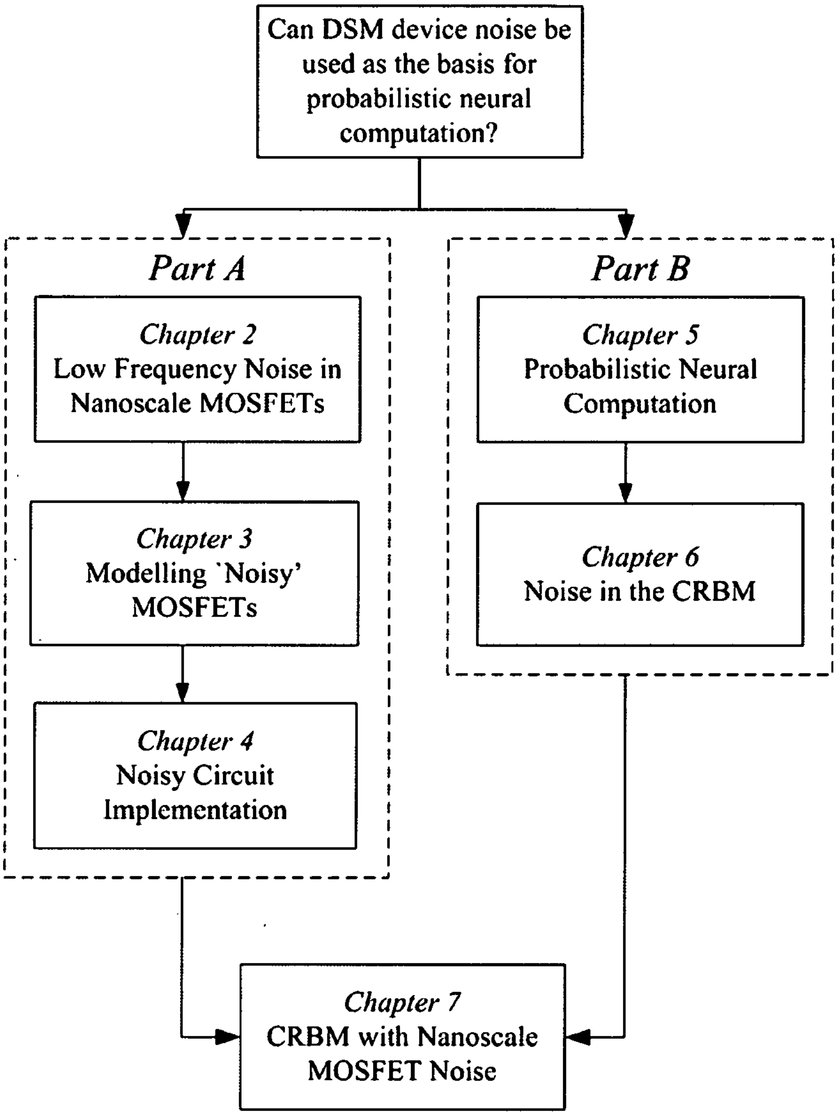
To achieve the objective of this project, temporal fluctuations of nanoscale MOSFET noise must be incorporated into the CRBM. Unfortunately, nanoscale devices are at least a decade away from everyday reality [6]. To pursue this project, temporal fluctuations in nanoscale MOSFETs will be simulated, based upon theoretical compact models of nanoscale MOSFETs extracted from atomistic simulation [25–27, 33]. Current simulators cannot support the requirement for time domain noise analysis based on nanoscale MOSFET noise characteristics (RTS). Therefore, in order to still pursue on the main objective of this project, a time domain RTS noise based simulation capability must be developed.

1.3 Chapter layout

This thesis can be separated into two relatively independent sections, A and B as illustrated by Fig.1.1. Part A deals with nanoscale MOSFET noise while part B discusses the chosen probabilistic neural model, the CRBM. Chapter 7 brings these together, linking nanoscale MOSFET characteristics and probabilistic neural computation into one common goal, providing useful computation. The chapters are:-

- Chapter 2 reviews the nanoscale MOSFET drain current low frequency noise: RTS and $1/f$ noise.
- Chapter 3 discusses the modelling of a noisy MOSFET for time domain noise analysis capability.
- Chapter 4 analyses the time domain output noise of the noisy analogue multipliers implementation, based on the capability developed in Chapter 3.
- Chapter 5 reviews the CRBM algorithm and architecture.
- Chapter 6 analyses the effect of noise on the CRBM performance.

- Chapter 7 presents the CRBM with nanoscale MOSFET noise implementation, and explore the performance of this implementation.
- Chapter 8 concludes the contribution and the future work of this research.


Can DSM device noise be used as the basis for probabilistic neural computation?
Part A
Chapter 2
Low Frequency Noise in Nanoscale MOSFETs
Chapter 3
Modelling `Noisy' MOSFETs
Chapter 4
Noisy Circuit Implementation
Part B
Chapter 5
Probabilistic Neural Computation
Chapter 6
Noise in the CRBM
Chapter 7
CRBM with Nanoscale MOSFET Noise

DSM : Dccp-Sub-Micrometcr

CRBM : Continuous Restrictcd Boltzman Machinc

Figure 1.1: *The flowchart illustrating the chapter flow in this thesis.*

Chapter 2
Low Frequency Noise in Nanoscale MOSFETs

2.1 Introduction

Low frequency noise becomes a dominant limiting factor in the practical use of MOSFETs in a circuit implementation as the devices enter nanoscale dimensions. It sets a lower limit to the level of signal that can be reliably processed by the circuit. Excessive low frequency noise could lead to serious performance and functionality limitations. Therefore, low frequency noise in MOSFETs has been studied extensively in the last few decades [4, 9, 11–14, 27, 34, 35].

Random Telegraph Signal (RTS) noise and $1/f$ noise are the two forms of low frequency noise that are predicted to dominate future nanoscale MOSFETs [9, 11, 12, 14]. In current technology, their existence is insignificant, and in most cases minimised or suppressed through either design or additional fabrication processes [6]. As MOSFET dimensions continue to shrink, their presence (noise) is predicted to become increasingly significant. Recent studies have shown low frequency drain current noise amplitude in excess of 60% in Deep-Sub-Micrometer (DSM) MOSFETs [13].

Fig.2.1 shows an example of simulated time domain 1-trap RTS and $1/f$ noise for a $35nm$ gate-length NMOS transistor based on a noisy MOSFET model developed in [2]. In reality, $1/f$ noise in general is more relevant to larger MOSFETs (>5-10 μm^2[11]), while as MOSFETs shrink to nanometer scale, RTS noise becomes dominant [4]. It is commonly agreed that the superposition of multiple RTS noise sources gives rise to $1/f$ noise. Understanding the microscopic origin of RTS noise therefore contributes to the understanding of the origin of $1/f$ noise.

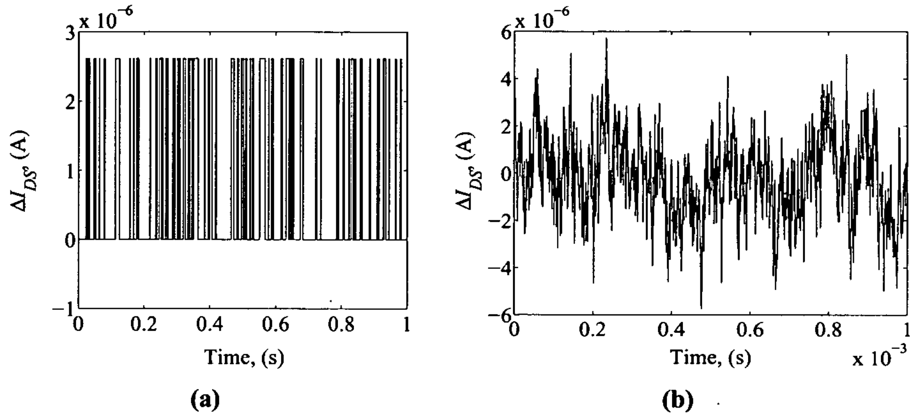


Figure 2.1: *Sample of (a) single trap RTS and (b) flicker noise.*

2.2 Random Telegraph Signal (RTS) Noise

Recent studies show that the low frequency performance of nanoscale MOSFETs is dominated by RTS noise [11, 13]. RTS noise arises from the capture and emission (trapping and detrapping) of hot electrons in the channel by traps (defects) at the interface of $Si - SiO_2$, causing discretised drain current fluctuations, as seen in Fig.2.1(a) [2].

RTS noise is characterised by three parameters: the average amplitude of fluctuation ΔI_D, the mean capture time $\bar{\tau}_c$, and the mean emission time $\bar{\tau}_e$. All these parameters vary over wide ranges with devices sizes, temperature, and bias conditions, where models to describe their dependencies have been developed [4, 27, 28, 36].

2.2.1 Origin of traps

There are four kinds of defects or traps commonly associated with a $Si - SiO_2$ interface: mobile ions, fixed charges, interfacial traps, and induced charges [1]. Illustrated by Fig.2.2, the defects/traps are briefly described below based on the detailed descriptions in [1].

- Mobile ions, typically Na^+ and K^+, lying within the SiO_2 interface are introduced through contamination during the fabrication process. They move around or redistribute under bias-temperature stressing, producing instability in the MOSFET's char-

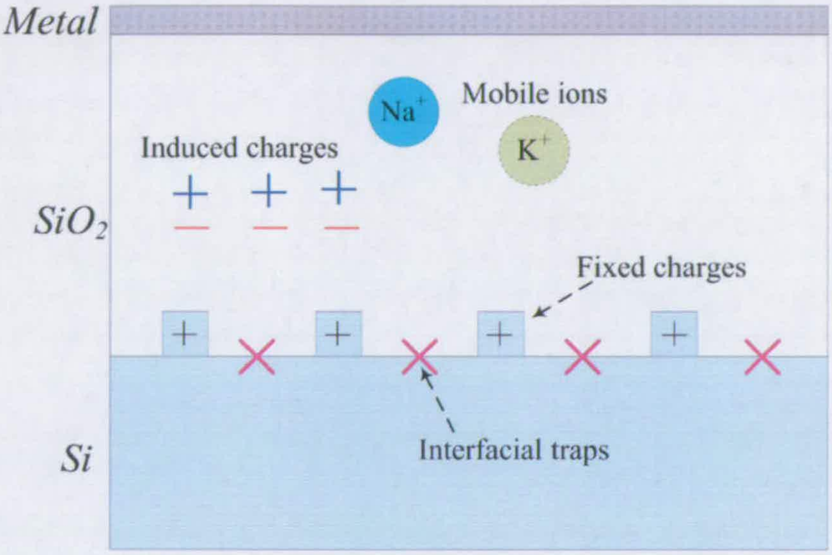


Figure 2.2: *An illustration of the traps and charges in $Si - SiO_2$ structures. (Adapted from [1]).*

acteristics.

- Fixed charges exist due to excess ionic silicon that has broken away during the oxidising reaction at the $Si - SiO_2$ interface. This explains the location of fixed charges in SiO_2. Unlike mobile ions, these fixed charges are consistent for a given set of fabrication conditions.

- Interfacial traps, which are situated at the $Si - SiO_2$ interface, are believed to arise from unsatisfied chemical bonds, or so-called "dangling bonds", at the surface of the Si during thermal formation of the SiO_2 layer. The interfacial traps introduce energy levels in the forbidden band gap at the $Si - SiO_2$ interface and remain fixed in energy relative to the conduction band and valence band energies.

- Induced charges are introduced into the SiO_2 due to ionising radiation, or hot carrier stress. The induced charges may be positively or negatively charged. They influence the MOSFET's characteristics by increasing or reducing the threshold voltage.

Interface traps and induced charges, unlike the mobile ions and fixed charges, are readily influenced by the bias conditions of the MOSFET. If they are within the tunnelling distance

($\leq$2 nm) of the hot electrons, they can be charged or discharged, creating fluctuations in the MOSFET's characteristics (i.e. the drain current I_{DS}), of the form of RTS.

2.2.2 RTS noise amplitude

The discretised drain current noise in a nanoscale MOSFETs is the combined effect of carrier number fluctuations and carrier mobility fluctuations [9, 37]. The normalised amplitude of this discrete fluctuation is described in the following general relation:

$$\frac{\Delta I_D}{I_D} = \frac{\Delta N}{N} \pm \frac{\Delta \mu}{\mu}, \tag{2.1}$$

where N is the number of channel carriers per unit area and μ is carrier mobility. The term $\left(\frac{\Delta\mu}{\mu}\right)$ in Eq.(2.1) describes the effect of mobility fluctuations caused by Coulombic scattering of the charged traps [4, 37–39]. The sign ($\pm$) indicates the electronic state of the traps (i.e. charged (+) or neutral (-)) [36], after capturing electrons [37]. A trap that is charged after capturing an electron increases the scattering effect which subsequently increases the noise amplitude ΔI_D [37]. If a trap becomes neutral after capturing an electron (charged when empty), the Coulombic scattering becomes weaker, reducing the noise amplitude ΔI_D [37]. As the MOSFET enters nanoscale dimensions, the carrier number fluctuations $\left(\frac{\Delta N}{N}\right)$ become dominant [9]. Therefore, the term $\left(\pm\frac{\Delta\mu}{\mu}\right)$ in Eq.(2.1) can be dropped. $\left(\frac{\Delta N}{N}\right)$ describes the carrier number fluctuations caused by the capture and emission of electrons by the trap. When a trap captures an electron from the channel, the effective drain to source current drops. When the trapped electron is released into the channel, the effective drain to source current increases. The normalised amplitude ($\Delta I_D/I_D$) dominated by $\left(\frac{\Delta N}{N}\right)$ can therefore be described by [40, 41]:

$$\frac{\Delta I_D}{I_D} = \alpha\frac{g_m}{I_D} \cdot \frac{q}{WLC_{ox}}\left(1 - \frac{x_t}{t_{ox}}\right), \tag{2.2}$$

where g_m is channel transconductance, W is channel width, L is channel length, C_{ox} is gate oxide capacitance, t_{ox} is gate oxide thickness, x_t is trap depth measured from the $Si - SiO_2$ interface. α in Eq.(2.2) is a semi-empirical parameter used to account for the wide variation

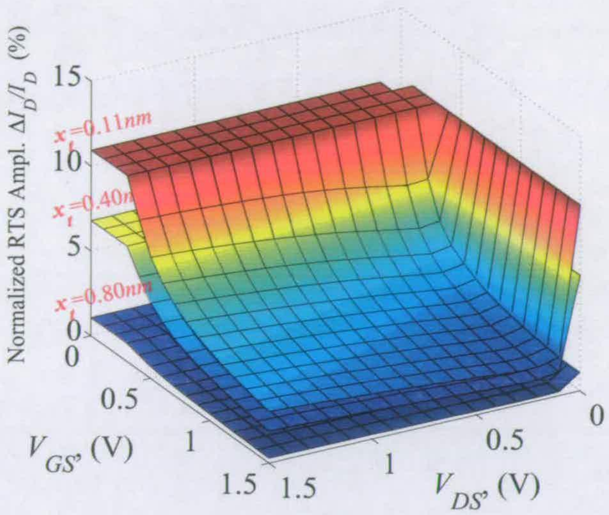


Figure 2.3: *RTS noise amplitude variation with gate V_{GS} and drain V_{DS} voltages for three different trap depths (x_t= 0.11, 0.40, 0.80 nm) located in the middle of gate (y_t= 16nm) for an implementation based on 35nm NMOS modelled in [2] .*

of RTS amplitude [40] caused by the short channel effect in MOSFETs operating in weak inversion [41]. α is in the range of 0.1 to 100 [42].

The dependence of typical RTS noise amplitudes on bias is shown in Fig.2.3. The noise amplitudes peak at low gate voltages (weak inversion), while at higher gate voltages, the amplitudes decrease. Recent studies have shown that RTS noise amplitudes vary by some 40% in weak inversion compared to 5% for strong inversion [27]. In addition, it has been reported that shallower traps (small x_t) produce a larger RTS noise amplitude [37], which is in agreement with Eq.(2.2).

2.2.3 RTS average capture and average emission time

The capture and emission of electrons cause fluctuations in channel conductance, which in turn causes the drain current to fluctuate. Capture and emission are stochastic events, obeying Poisson statistics, and are normally described by the average values of $\bar{\tau}_c$ and $\bar{\tau}_e$, respectively [36]. $\bar{\tau}_c$ represents the mean time that a trap is empty before capturing an electron and $\bar{\tau}_e$ represents the mean time in which a trapped electron is freed. Eq.(2.3) and Eq.(2.4) [4] describe the basic empirical model for $\bar{\tau}_c$ and $\bar{\tau}_e$ that are used in this work.

$$\bar{\tau}_c = \frac{\exp\left(\frac{\Delta E_B}{kT}\right)}{\sigma_0 v_{th} n} \tag{2.3}$$

$$\bar{\tau}_e = \frac{\exp\left[\left(\Delta E_B + \Delta E_{CT}\right)/kT\right]}{g \sigma_0 v_{th} n} \tag{2.4}$$

In Eq.(2.3) and Eq.(2.4), σ_0 is the trap cross-section pre-factor, $\Delta E_{{}_B}$ is the barrier energy for capture (also known as activation energy for capture), ΔE_{CT} is the trap binding energy, $\Delta E_B + \Delta E_{CT}$ is the emission activation energy, v_{th} is the average thermal velocity, n is the channel electron concentration, k is the Boltzmann constant, T is the absolute temperature, and g is the degeneracy factor which is normally set to 1 [4].

The capture and emission time are thermally activated processes which are inversely proportional to temperature, as described in both Eq.(2.3) and Eq.(2.4). For a fixed operating temperature, the capture and emission time are affected by bias conditions through σ_0, $\Delta E_{{}_B}$, and ΔE_{CT}. It has been reported that σ_0 and ΔE_{CT} depend strongly and positively on gate voltage, V_{GS} while $\Delta E_{{}_B}$ does not depend on either V_{GS} or V_{DS} [4, 5, 28]. The channel electron concentration's, dependence on drain voltage, V_{DS}, is strongly influenced by the position along the channel at which the trap is located. For an n-MOSFET, n near the source is not affected by V_{DS} while n near the drain shows a strong inverse relationship with V_{DS} [2]. An example of mean capture and emission time variation with bias conditions for a trap near the $Si - SiO_2$ interface (x_t= 0.11nm) located in the middle of the channel (y_t= 16nm) is shown in Fig.2.4.

2.2.4 Discussion

The capture and emission (trapping and de-trapping) of channel hot electrons by a trap (defect) at the interface of $Si - SiO_2$, causes discretised drain current fluctuations with amplitude exhibiting a non-Gaussian distribution [11]. The power spectral density (PSD) of 1-trap RTS noise exhibits a Lorentzian spectrum, described by [4, 28, 29]:

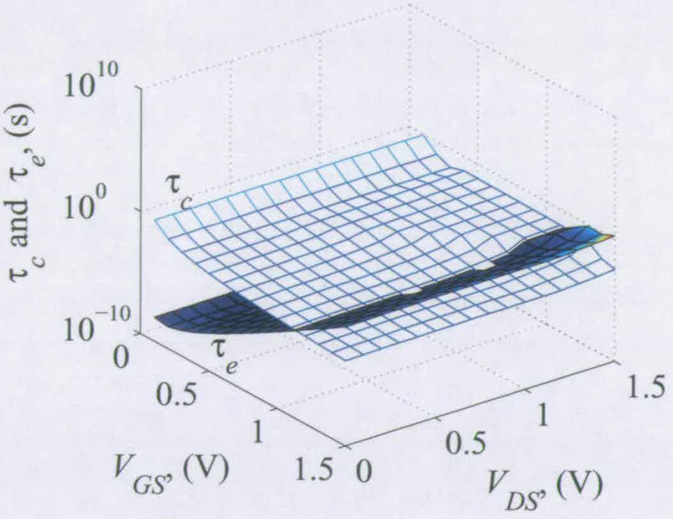


Figure 2.4: *Average capture $\bar{\tau}_c$ and average emission $\bar{\tau}_e$ vs gate V_{GS} and drain V_{DS} voltages for trap located at x_t= 0.11nm and y_t= 16nm for an implementation based on 35nm NMOS modelled in [2].*

$$S(f) = \frac{4\left(\Delta I_D\right)^2}{\left(\bar{\tau}_c + \bar{\tau}_e\right)\left[\left(\frac{1}{\bar{\tau}_c} + \frac{1}{\bar{\tau}_e}\right)^2 + \left(2\pi f\right)^2\right]}, \tag{2.5}$$

The number of discrete levels of drain current clearly depends on the number of traps. In practical MOSFETs of $1.0 \times 0.15\ \mu m^2$ dimensions, the number of interface trap densities are usually in the order of $10^{10}\mathrm{eV}^{-1} \cdot \mathrm{cm}^{-2}$ [43], of which several could be active (i.e. energy level within *kT*) causing multi-level RTS noise. For multi-trap RTS, Eq.(2.5) is generalised to [4]:

$$S_I(f) = \sum_{k=1}^{N_{traps}} \frac{4\left(\Delta I_D\right)_k^2}{\left(\bar{\tau}_c + \bar{\tau}_e\right)_k\left[\left(\frac{1}{\bar{\tau}_c} + \frac{1}{\bar{\tau}_e}\right)_k^2 + \left(2\pi f\right)^2\right]} \tag{2.6}$$

where N_{traps} represents the total number of active traps contained within the $Si - SiO_2$ interface and $S_I(f)$ is the current noise power spectral density summed over all traps $k = 1, 2, 3, \ldots, N_{traps}$. For a large number of active traps, N_{traps}, uniformly distributed (both throughout the oxide and in energy [44]) with a wide distribution of time constant ($\bar{\tau}_c$ and $\bar{\tau}_e$), the superposition of Lorentzian spectra in Eq.(2.6) gives rise to $S_I(f)$ with a $1/f$ form [4, 12].

This provides a physical justification that the superposition of several RTS noises gives rise to $1/f$ noise in MOSFETs.

2.3 Flicker ($1/f$) Noise

The origin of $1/f$ noise in MOSFETs has been extensively studied for more than a decade [9, 14, 44–46]. It has been agreed recently that $1/f$ noise in MOSFETs is associated with both carrier number fluctuations and correlated carrier mobility fluctuations [4, 47]. Carrier number fluctuations come from the random trapping and detrapping of free carriers in the oxide traps near the $Si - SiO_2$ interface, where the trapped carriers limit the mobility of the free carriers near the interface by Coulombic scattering [4].

2.3.1 $1/f$ noise model

In practice, MOSFET drain current $1/f$ noise is commonly described by its Power Spectral Density (PSD). A typical MOSFET drain current $1/f$ noise PSD has the following form [48]:

$$S(f) = \frac{K_f I_{DS}^{af}}{C_{ox} L_{eff}^2 f^{ef}}, \tag{2.7}$$

where K_f and a_f are process-dependent constants and may vary from sample to sample. The noise exponent a_f is typically between 0.5 and 2 [49]. The $1/f$ noise coefficient K_f was claimed in [47] to be bias-dependent but in common practice [48], K_f is considered constant. The frequency exponent, e_f, is a bias-dependent parameter with a typical value ranging between 0.7 and 1.2 [50]. C_{ox} is the gate oxide capacitance and L_{eff} is the effective channel length of a MOSFET.

A unified model of a MOSFET's drain current $1/f$ noise PSD incorporating both carrier number fluctuation and mobility fluctuation has been proposed by Hung [45]. The widely adopted analytical expression of the unified model in strong inversion is given by [48]

$$\begin{aligned} S(f) &= \frac{kTq^2 I_{DS}\mu_{eff}}{\gamma f^{ef} L^2 C_{ox}}\left[A\ln\frac{N_O+N^*}{N_L+N^*}+B\left(N_O-N_L\right)+\frac{1}{2}C\left(N_O^2-N_L^2\right)\right] \\ &+\Delta L_{clm}\frac{kTI_{DS}^2}{\gamma f^{ef} W L^2}\times\frac{A+BN_L+CN_L^2}{\left(N_L+N^*\right)^2}, \end{aligned} \qquad (2.8)$$

where

$$qN_O = C_{ox}\left(V_{GS}-V_{th}\right) \qquad (2.9)$$

$$qN_L = C_{ox}\left(V_{GS}-V_{th}-V_{DS}\right) \qquad (2.10)$$

and $N^* = \left(kT/q^2\right)\left(C_{ox}+C_d+C_{it}\right)$. C_d and C_{it} are depletion layers and interface trap capacitance respectively. γ is the attenuation coefficient of the electron wave function in the oxide, with a typical value of $10^8 cm^{-1}$[48]. A, B, and C are technology-dependent model parameters. ΔL_{clm} refers to the electrical channel length reduction due to channel length modulation. N_O and N_L are the charge densities at the source and the drain ends of the channel respectively.

2.3.2 Discussion

Eq.(2.7) is a simple model of the MOSFET's drain current $1/f$ noise PSD, used largely for 'long channel' MOSFETs [47]. Nevertheless, it is able to explain the general frequency and size dependence of $1/f$ noise in any MOSFET correctly. In practice, Eq.(2.7) is much easier to understand and implement, especially for the initial low frequency noise characteristic approximation of a MOSFET.

The approach to carrier trapping-detrapping $1/f$ noise provides a better physical explanation of $1/f$ noise in MOSFETs. Eq.(2.8) is a complex model based on this approach and is considered to be a 'complete' model, capable of expressing $1/f$ noise in all MOSFET operation regions in strong inversion. In the absence of technology-dependent model parameters and charge density estimations, Eq.(2.8) may not be immediately useful to describe 'accurate' low frequency noise in MOSFETs. In this case, Eq.(2.7) provides a reasonable approximation of Eq.(2.8).

2.4 Summary

This chapter has drawn together results from the extensive research and literature of Deep-Sub-Micrometer (DSM) noise modelling. It has been established that an approach based on the capture and emission of electrons by traps near the vicinity of the $Si - SiO_2$ interface provides a clear and simple explanation of dominant low frequency noise sources (RTS and $1/f$) in nanoscale MOSFETs, which accounts for both the effect of carrier number fluctuation and carrier mobility fluctuation. In general, low frequency noise is inversely proportional to MOSFET gate area. For a small MOSFET, RTS noise dominates the low frequency, producing discretised drain current. For large MOSFETs, the superposition of RTS noise gives rise to $1/f$ noise.

Chapter 3
Modelling 'Noisy' MOSFETs

This thesis aims to demonstrate the principle that nanoscale MOSFET low frequency noise can underpin useful probabilistic behaviour. The low frequency noise is used as a source of probabilistic behaviour in a modelled nanoscale silicon 'neuron' that adapts to the natural variability in input data. For this purpose, a noisy MOSFET model that accurately represents the temporal fluctuation of a real nanoscale MOSFET is required. This chapter describes the method used to model the noisy MOSFET. The capability to represent the dominant nanoscale MOSFET low frequency noises (RTS and $1/f$) will be demonstrated and verified. The noisy MOSFET model will be used to implement a key circuit for the silicon 'neuron' in Chapter 4.

3.1 Introduction

The aim of this study is to develop a SPICE-based behavioural model of a 'noisy' MOSFET, preserving the underlying, essential MOSFET transfer characteristics, as illustrated in Fig.3.1.

A credible, computationally-simple noisy MOSFET model is critical to this work. At this concept-proving stage, however, simplicity and simulation speed are more important than great accuracy. It is not, therefore, the objective of this study to develop a complex, arbitrarily accurate noisy DSM MOSFET model. Rather, the focus is to develop a model that produces a correct form of noise behaviour, valid in all operating regimes, and is easy to implement in a readily-available circuit simulator.

The work described in this chapter focuses on two low frequency MOSFET noise models: $1/f$ noise and RTS noise. $1/f$ noise in general is more relevant to the larger area MOSFETs

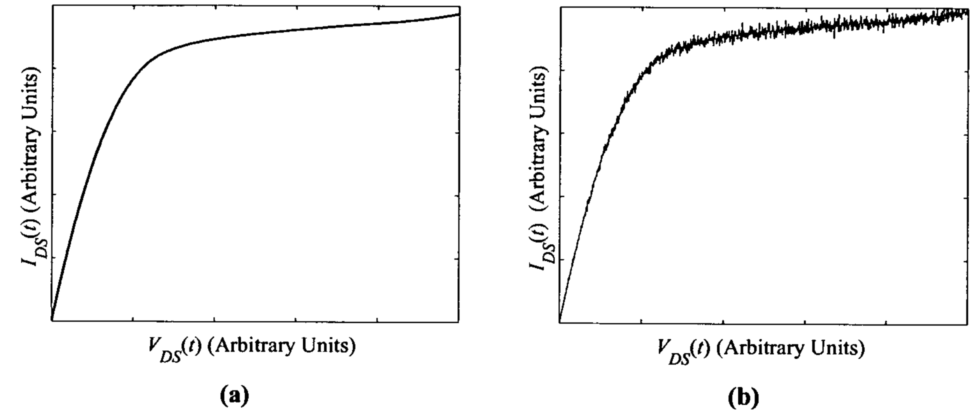


Figure 3.1: *(a) Standard n-channel MOS drain-source current I_{DS} for a given V_{GS} voltage. (b) noisy n-channel MOS drain current for a given V_{GS} voltage.*

($> 5-10\mu m^2$ [11]), while as the MOSFET shrinks to nanometer scale, the RTS noise become dominant [11, 36]. Current models predict that the superposition of RTS noise produces $1/f$ noise, and that the transition from a $1/f$ -dominated regime to that of true RTS noise is gradual—not in the form of a first-order 'phase transition' [4, 11, 45] .

3.2 Methodology Overview

The noisy MOSFET was modelled by augmenting a conventional noiseless MOSFET with a noise source, $n(t)$ (Fig.3.2). In this study, the initial MOSFET model was that for a $0.35\mu m$ AMS CMOS technology. However, the overall and final implementation of the noisy MOSFET was based upon a $35nm$ gate length MOSFET, for $90nm$ CMOS technology node, extracted from atomistic simulation [25, 26, 33].

The noiseless MOSFET is used to generate the I_{DS} response of the noisy MOSFET at a particular time instance t, for the given bias conditions (V_{GS} and V_{DS}). As bias conditions change in time, the noisy MOSFET drain current therefore changes correspondingly. The methodology uses I_{DS} static model in transient analysis. This is valid as the I_{DS} reaches steady state in pico-seconds [51] while the low frequency noise fluctuates in micro-seconds.

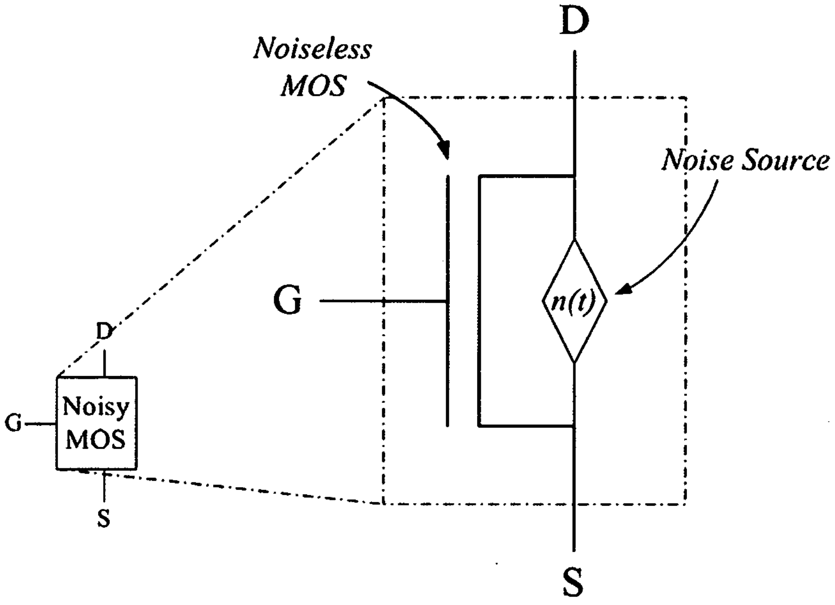


Figure 3.2: *Noisy MOSFET components. The noiseless MOSFET is a standard MOSFET having a typical I-V characteristic corresponding to the technology used. The noise source $n(t)$ generates noise data with the specific physical characteristics of the low frequency noise it models.*

The noise source $n(t)$ represents the $1/f$ or RTS behaviour of the noisy MOSFET. Time-domain models of $1/f$ noise are calculated from the $1/f$ Power Spectral Density (PSD) (described in Sec.2.3.1) generated for each bias point of the MOSFET. For RTS noise, time-domain models are developed using RTS parameters (amplitude and time statistics) calculated at each bias point of the MOSFET. Details are illustrated in Sections 3.3 and 3.4.

To ensure that the methodology used for generating noise does indeed produce time-domain noise with the intended spectral characteristics, the PSD of the generated time-domain noise is calculated using a PSD extraction algorithm.

In general, the DC drain-source current, I_{DS}, generated at a given time t, is used by the noise source $n(t)$ to generate *one* noise datum. The noise data generated in time space have the correct*form* of noise behaviour that the noise source $n(t)$ is supposed to model. If bias conditions change in time, the noise source $n(t)$ generates non-stationary noise data corresponding to the changing DC response of the noiseless MOSFET.

The noisy MOSFET model is used as a 'standard' component in circuits. It is essential that the method of generating noisy drain current does not otherwise disrupt the use of the circuit simulation under transient analysis. Therefore, a high-level analogue behavioural language, called Verilog-A, is used to model the noisy MOSFET. Implementation using Verilog-A allows full control of the noisy MOSFET behaviour, in addition to implementation flexibility for future modification.

3.3 Generating $1/f$ Noise

The objective of generating time-domain $1/f$ noise is somewhat different from that of most other work [52, 53]. Although the method chosen has been used to develop commercial transient noise analysis [53], this is not the aim of this work. The aim is to model a noisy MOSFET in a circuit implementation of a particular computational architecture, in order to explore the effects of noise on its performance.

3.3.1 Methodology: $1/f$ noise

The critical part in modelling the noisy MOSFET is the implementation of the $1/f$ noise source, $n(t)$. The $1/f$ noise source should exhibit the correct *form* of noise characteristic corresponding to the MOSFET technology and bias conditions.

In this work, $1/f$ noise was generated based on the Sum-of-Sinusoids technique adapted from [52]. This is the best approach for the use with Verilog-A implementation, even though other approaches are possible. The technique generates the time domain $1/f$ noise data, $n(t)$, by summing a fixed number N_f of random phase sinusoids in a specified frequency band [52]:

$$n(t) = \sum_{i=1}^{N_f} a_i(t)\sin(2\pi f_i t + \varphi_i), \tag{3.1}$$

where a_i is the magnitude, f_i is the frequency and φ_i is the random phase defining the i^{th} sinusoid.

N_f represents the number of sinusoids used to approximate noise data at a given time t. For a fixed frequency band (band-limited) PSD $S(f)$ (Eq.(2.7)), N_f depends on the division of the frequency band into frequency steps, Δf. A smaller Δf means more sinusoids, better $n(t)$ approximation, and longer transient data representation. Depending on the shape of $S(f)$, the frequency band may be divided linearly or logarithmically.

For each frequency interval f_i and $f_i + \Delta f$ of a given PSD $S(f)$ shown in Fig.3.3, the magnitude a_i is approximated as:

$$a_i = \sqrt{2 \int_{f_i}^{f_i+\Delta f} S(f) df}. \tag{3.2}$$

During simulation, as bias conditions change, $S(f)$ changes accordingly. For $1/f$ noise, the PSD is proportional to the drain-source current, I_{DS}, as described by Eq.(2.7) in Sec.2.3.1. Consequently, according to Eq.(3.2), the magnitude a_i will also change. The values for a_i are updated at each time step to cater for changes in bias conditions. Changes in bias conditions therefore affect generated $1/f$ noise magnitude and spectrum characteristics through Eq.(3.2) and Eq.(3.1). This enables the modelled noise generated to be of non-stationarity characteristics in the case where bias conditions change with time, at the expense of lengthy simulations.

f_i represents the frequency of the i^{th} sinusoid. φ_i is a random phase angle, uniformly distributed between 0 and 2π. $\varphi_i$values are unique for each sinusoid across t.

3.3.2 Implementation: $1/f$ noise

The initial n-channel MOSFET (NMOS) used to establish our methodology was based upon a $0.35\mu m$ AMS CMOS model. Gate length L, width W, and oxide thickness T_{ox} were set to $0.35\mu m$, $1\mu m$, and $7.7nm$, respectively. The $1/f$ noise coefficient K_f, exponent a_f, and frequency exponent e_f values were set to 2.81e-27 $A \cdot F$, 1.4, and 1, respectively. These values were taken from the AMS CMOS technology file used in simulation. In order to generate statistically significant noise amplitudes useful for this work, K_f was arbitrarily increased to 2.81e-23 $A \cdot F$.

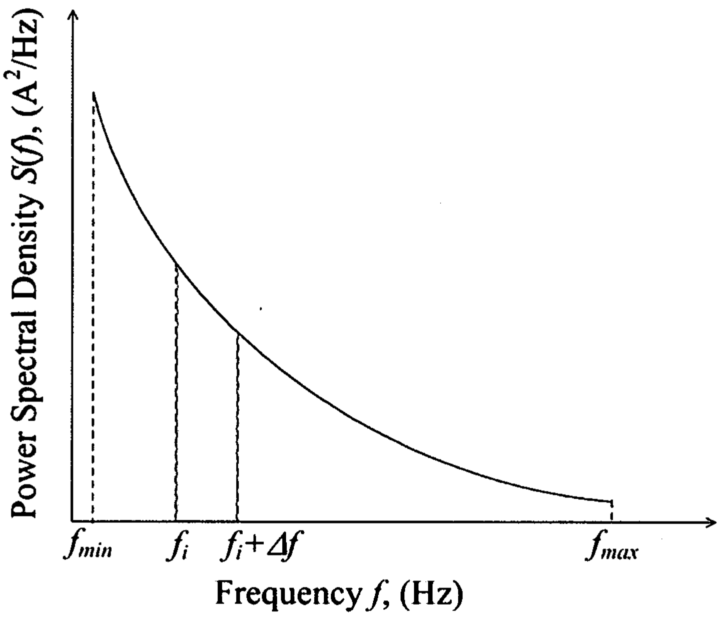


Figure 3.3: *Illustration of* $1/f$ *power spectral density* $S(f)$.

DSM implementation was based upon a $35nm$ gate length NMOS model developed using atomistic simulation. We set L=$35nm$, W=$0.1\mu m$, T_{ox}=$0.88nm$. The flicker noise parameters were not available in this $35nm$ NMOS model. Therefore, in this study, the values used for $0.35\mu m$ CMOS technology were re-used for the $35nm$ NMOS implementation.

These values and this approach are not acceptable for a thorough exploration of noise in $35nm$ MOSFETs. However as explained in Sec.3.3, they are more than adequate for the aims of this study, capturing as they do the most important characteristics of DSM noise at circuit level.

Using these parameters, the $1/f$ noise source $n(t)$ was implemented based on the method in Sec.3.3.1. It is thought that $1/f$ noise is important below 10kHz [9, 46], and we chose to generate $1/f$ noise between 100 Hz (f_{min}) and 10 kHz (f_{max}). So, for a given bias conditions (i.e. I_{DS}), frequency band, and MOSFET parameters, a corresponding PSD can be generated based on Eq.(2.7).

This PSD is then used to generate the amplitudes of the sinusoids in Eq.(3.1) using Eq.(3.2). The frequency band was initially divided linearly with Δf set to 100 Hz. Therefore, based upon the set frequency band and frequency step, there are 99 terms in equation Eq.(3.1).

It should be noted that this calculation is performed to generate *one* noise datum at a particular time instance t and then the noise datum, $\Delta I_{DS}(t)$, is added to generate $(I_{DS}(t) + \Delta I_{DS}(t))$. At the next simulation time step, the process is repeated.

The implementation steps for generating time domain $1/f$ noise are summarised as follows:

1. At $t = 0$ (initialisation)
   - Set the lowest frequency f_{min} and highest frequency f_{max} .
   - Set the frequency step Δf for dividing the band-limited PSD $S(f)$.
   - Calculate the number of sinusoids: $N_f = \frac{f_{max}-f_{min}}{\Delta f}$.
   - Generate N_f random phase angle using random number generator.
2. Probe the noiseless MOSFET drain-source current I_{DS} at time t.
3. Use the I_{DS} to generate band-limited ($f_{min} \rightarrow f_{max}$) PSD $S(f)$ using Eq.(2.7).
4. Approximate the magnitude a_i for each frequency interval f_i and f_i+Δf using Eq.(3.2).
5. Calculate the noise datum $n(t)$ using Eq.(3.1).
6. Add the noise datum $n(t)$ to I_{DS} to produce noisy I_{DS}.
7. Repeat 1-6 until the end of simulation time.

3.4 Generating RTS Noise

RTS noise in MOSFETs has been accepted and studied for some time [4,37,54,55]. Most agree that the physical origin of RTS noise is the trapping and de-trapping of electrons by traps located at or near the vicinity of the $Si - SiO_2$ interface [28,43] and RTS noise causes discretised drain current fluctuations. However, a 'standard' and generally accepted model for RTS noise that is valid for all operating regimes does not yet exist [28].

The number of discrete values of I_{DS} depends on the number of traps [38]. Most RTS noise models are based upon single trap capture and emission activity [4, 28], and restricted to room temperature operation in ohmic regimes at strong inversion [4, 28, 43].

In this work, we assembled a more wide-ranging RTS model based upon previously-published work [4, 28, 36, 37, 41, 55–57], and made some pragmatic assumptions to enable the models to be used in our circuit and target architecture.

3.4.1 Assumptions: RTS noise

The RTS noise is generated based upon existing models, which were developed and extended, with the following assumptions and restrictions:-

1. Only electron traps are considered and the noisy-MOSFET modelling was limited to n-type devices at this stage, as very few models of hole traps have been reported.
2. All traps are considered neutral when empty. Attractive and neutral traps have larger cross-sections (in the range of $10^{-14} - 10^{-12} cm^2$ and $10^{-18} - 10^{-14} cm^2$, respectively) than do repulsive (negative) traps with a cross-section smaller than $10^{-18} cm^2$, and a concomitantly low capture probability [58]. Neutral traps produce larger RTS amplitudes compared to the attractive traps and therefore representing the normally observable, hence analyzable RTS noise [37].
3. Active traps reside in the volume between the $Si - SiO_2$ interface and oxide at depth within the tunnelling distance ($\leq 2\ nm$) of any hot electrons, limited by gate oxide thickness t_{ox}.
4. Adjacent traps are at least $2nm$ apart. This is important to ensure that traps are electrostatically isolated [34], and thus there is no interaction such as tunnelling between traps [4].
5. The capture and emission of an electron by a single trap are mutually exclusive events.
6. Only one electron can be trapped by each trap at any particular time. It was reported

by [4] that RTS noise due to multi-electron trapping by a single trap can occur for traps located in Si rather than the $Si - SiO_2$ interface.

7. No Coulombic scattering effect (channel blocking). Coulombic scattering would cause trapped electrons to become a mid-filter that repels the further capturing process of electrons. This effect is mainly prominent in strong inversion and in very weak inversion [4].

8. It is assumed that capture and emission time are not affected by electron temperature. However, according to [4], by applying a drain-source voltage, the electron temperature can be raised above the underlying lattice temperature, giving rise to temperature-dependent capture and emission as evident by Eq.(2.3) and Eq.(2.4). By ignoring the effects of electron temperature, this assumption introduces inaccuracy to capture and emission time approximation especially for high bias conditions.

9. The amplitude of RTS is not affected by lateral trap position along the channel region.

With these assumptions, a suitably accurate *form* of RTS noise, valid for low bias conditions and typical temperatures can be generated while at high bias conditions and corresponding high temperatures, the accuracy will be compromised. It is acknowledged again that RTS noise thus generated may not be a completely accurate representation of real RTS noise in MOSFETs. As stated in Sec.3.1, that is not the main aim of this study.

3.4.2 RTS Amplitude

When a trap captures an electron from the channel, the effective drain to source current drops. When the trapped electron is released into the channel, the effective drain to source current increases. The normalised amplitude ($\Delta I_D / I_D$) of current fluctuation between the capture and emission of an electron by the trap is described by Eq.(2.2) in Sec.2.2.2 [40, 41]

Eq.(2.2) assumes RTS noise amplitude does not depend on the position of the trap along the channel region. Evidently, this assumption is not wholly accurate [27, 57]. The RTS amplitude has been shown to peak when the trap is located at the centre of the channel region, and reduces for locations toward the source or drain [27,57]. However, for simplicity, in this study it is assumed that Eq.(2.2) is sufficient to describe RTS amplitude that depends solely on trap depth x_t. It is acknowledged that it is crucial to refine Eq.(2.2) to include the effect of trap position along the channel on noise amplitude in future models. From Eq.(2.2), the deeper the trap location into the oxide from the $Si-SiO_2$ interface, the smaller the amplitude. The maximum amplitude occurs when the trap lies at the interface, $x_t = 0$, where Eq.(2.2) reduces to:

$$\frac{\Delta I_D}{I_D} = \alpha \frac{g_m}{I_D} \cdot \frac{q}{WLC_{ox}}. \tag{3.3}$$

RTS amplitude varies by some 40% in weak inversion compared to 5% for strong inversion [27]. This suggests that RTS noise will depend strongly on the bias point of the MOSFET. In Eq.(2.2), the bias dependence of RTS amplitude is modelled through the channel transconductance g_m.

Transconductance is generally calculated as $g_m = \frac{\delta I_{DS}}{\delta V_{GS}}$. In this study $g_m = \frac{\delta I_{DS}}{\delta V_{GS}}$ is calculated based on the inversion layer approximation [59] in order to cater for weak inversion operation of the MOSFET in which significant RTS amplitude has been found :

g_m can be approximated as [59]:

$$g_m = \frac{I_{DS}}{nkT/q} \tag{3.4}$$

for weak inversion, and

$$g_m = \sqrt{\frac{2\mu C_{ox}}{n} \left(\frac{W}{L}\right) I_{DS} \left(1 + \lambda V_{DS}\right)} \tag{3.5}$$

for strong inversion, where λ is the channel length modulation parameter.

For high drain voltage, g_m is calculated using the velocity saturation approximation [59]:

$$g_m = WC_{ox}v_{sat}. \tag{3.6}$$

The transition between weak and strong inversion is approximated using the transition current I_{DSWS} [59]:

$$I_{DSWS} = \frac{\mu C_{ox}}{2n} \cdot \frac{W}{L}\left(2n\frac{kT}{q}\right)^2, \tag{3.7}$$

and the transition from strong inversion to velocity saturation is approximated by the transition current I_{DSSV} [59]:

$$I_{DSSV} = \frac{8nWLC_{ox}^2 v_{sat}^2}{\mu C_{ox}}. \tag{3.8}$$

3.4.3 RTS mean capture and emission time

The capture and emission of electrons cause fluctuations in the channel conductance, which in turns causes the drain current to fluctuate. $\bar{\tau}_c$ represents the mean time that a trap is empty before capturing an electron and $\bar{\tau}_e$ represents the mean time of a trapped electron is freed. $\bar{\tau}_c$ and $\bar{\tau}_e$ are described by Eq.(2.3) and Eq.(2.4) respectively in Sec.2.2.3.

The trap cross-section pre-factor σ_0, activation energy for capture ΔE_B, and trap binding energy ΔE_{CT} are parameters specific for each trap at a given trap location, bias conditions and temperature [4, 28]. σ_0 and ΔE_{CT} depend strongly and positively on gate voltage, V_{GS}, while ΔE_B does not depend on either V_{GS} or V_{DS} [4, 5, 28]. There has been no reported explicit relation describing σ_0 and ΔE_{CT} dependence to bias and/or temperature. In most cases, their values were extracted from the study of temperature- and bias-dependence of RTS noise from which they were found as fitting parameters [4, 5, 28].

The channel electron concentration, *n* is a further important parameter in Eq.(2.3) and Eq.(2.4). In linear operation (assuming constant electron concentration along the channel), *n* is typically described by [4, 28]:

$$n = \frac{I_{DS}L_{eff}}{q\mu V_{DS}t_{inv}W_{eff}}, \tag{3.9}$$

where μ is electron mobility, t_{inv} is the inversion layer thickness, and W_{eff} and L_{eff} are the effective channel width and length, respectively. In saturation operation, Eq.(3.9) is no longer applicable, as electron concentration can no longer be approximated by a constant channel. A general description of electron concentration at a specific location in the channel is given by [58]:

$$n(x,y) = \frac{n_i^2}{N_a} \exp\left[q\left(\psi(x) - V(y)\right)/kT\right], \tag{3.10}$$

where x is the depth measured from the $Si - SiO_2$ interface, y is the lateral location measured from source, n_i is the intrinsic carrier concentration, $\psi(x)$ is the band bending potential at depth x, $V(y)$ is the quasi-Fermi potential at lateral location y, and N_a is the substrate doping concentration. Eq.(3.10) is too complex to be used here, as it would involve numerical methods to solve for $\psi(x)$ [58]. In this work, the charge sheet approximation was assumed, within which Eq.(3.10) becomes:

$$n(0,y) = \frac{n_i^2}{N_a} \exp\left[q\left(\psi(0) - V(y)\right)/kT\right], \tag{3.11}$$

where $\psi(0) \equiv \psi_S$ is the surface potential. A description of surface potential ψ_S and quasi-Fermi potential $V(y)$ can be found in Appendix A.

Finally, $\bar{\tau}_c$ and $\bar{\tau}_e$ are inversely proportional to thermal velocity, v_{th}, which is described by [4]:

$$v_{th} = \sqrt{\frac{8kT}{\pi m^*}} \tag{3.12}$$

where $m^*$is the effective mass of an electron.

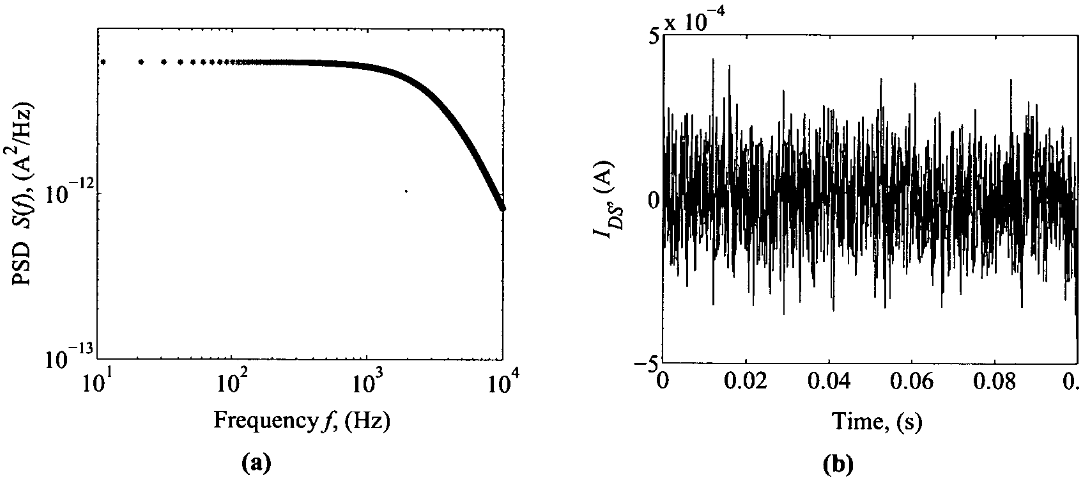


Figure 3.4: *(a) Single trap RTS noise Power Spectral Density, $S(f)$ generated using Eq.(2.5). (b) Time domain RTS noise generated from Power Spectral Density $S(f)$ using Sum-of-Sinusoids techniques.*

3.4.4 Methodology: RTS noise

The method used to generate $1/f$ noise is not applicable for RTS noise, as it cannot generate the discrete noise levels that characterise RTS noise. Fig.3.4(b) shows RTS noise generated from the a single trap RTS noise PSD in Fig.3.4(a) described by Eq.(2.5). The inability of the method described in Sec.3.3.1 to generate the correct time-domain form of RTS noise is clear.

An alternative method was therefore developed, that generates RTS amplitude and mean time statistics using the Monte Carlo simulation method.

The lifetime of a filled trap (lowered I_{DS} current) and an empty trap (high I_{DS} current) obey Poisson statistics [4, 11, 28]. The probability that an empty trap captures an electron is [4]:

$$p_c(t) = \frac{1}{\bar{\tau}_c} \exp\left(-\frac{t}{\bar{\tau}_c}\right), \tag{3.13}$$

and the probability that a full trap emits an electron is:

$$p_e(t) = \frac{1}{\bar{\tau}_e} \exp\left(-\frac{t}{\bar{\tau}_e}\right), \tag{3.14}$$

·where $p_c(t)$ and $p_e(t)$ are the normalised probabilities:

$$\int_0^\infty p_{c/e}(t)dt = 1. \tag{3.15}$$

where $p_{c/e}(t)$ represents either $p_c(t)$ or $p_e(t)$. Based on a Monte Carlo method [60], Eq.(3.15) is generalised to:

$$\int_0^{t_{Tran}} p_{c/e}(t)dt = P(t_{tran}). \tag{3.16}$$

where $P(t_{Tran})$ is the probability of capture or emission by the transition time t_{Tran}, having values from $0(t_{Tran} = 0)$ to $1(t_{Tran} = \infty)$. Performing the integration results in:

$$1 - \exp\left(-\frac{t_{Tran}}{\bar{\tau}_{c/e}}\right) = P(t_{Tran}). \tag{3.17}$$

Eq.(3.17) can be manipulated algebraically to obtain the following

$$\exp\left(-\frac{t_{Tran}}{\bar{\tau}_{c/e}}\right) = 1 - P(t_{Tran}). \tag{3.18}$$

Taking the natural log of both side of Eq.(3.18) and performing another algebraic manipulation gives

$$t_{Tran} = -\bar{\tau}_{c/e} \ln\left(1 - P(t_{Tran})\right). \tag{3.19}$$

If random number generator is used to generate the probability $P(t_{Tran})$, Eq.(3.19) can be used to determined the corresponding transition time t_{Tran}. However, since $P(t_{Tran})$ is evenly distributed between 0 and 1, Eq(3.19) can be re-written as

$$t_{Tran} = -\bar{\tau}_{c/e} \ln\left(P(t_{Tran})\right) \tag{3.20}$$

x_t(nm)	Temp (K)	V_{GS}(V)	σ_0(cm^2)	ΔE_B(eV)	ΔE_{CT}(eV)
1.1	310	3	2.10e-19	0.411	0.076
1.3	320	0.86	8.40e-20	0.186	0.218
1.4	330	1.45	5.80e-15	0.645	0.102
0.9	270	3.67	3.80e-15	0.480	0.078
1	300	1.9	2.04e-15	0.593	0.108
1.2	300	0.86	7.00e-20	0.300	0.750
1.25	300	1.25	8.90e-18	0.320	0.060

Table 3.1: *Fitting parameters σ_0, ΔE_B , ΔE_{CT}, for seven room temperature traps observed in 0.4 μm^2n-channel MOSFETs with corresponding estimated trap depth x_t [4, 5].*

Eq.(3.20) can be used to determine the transition time t_{Tran}. It is assumed that the capture and emission of an electron by a trap are mutually exclusive events. When a trap is empty, the only transition possible is capture and vice versa. Therefore, the generated probability of transition $P(t_{Tran})$ applies to either for electron capture or emission.

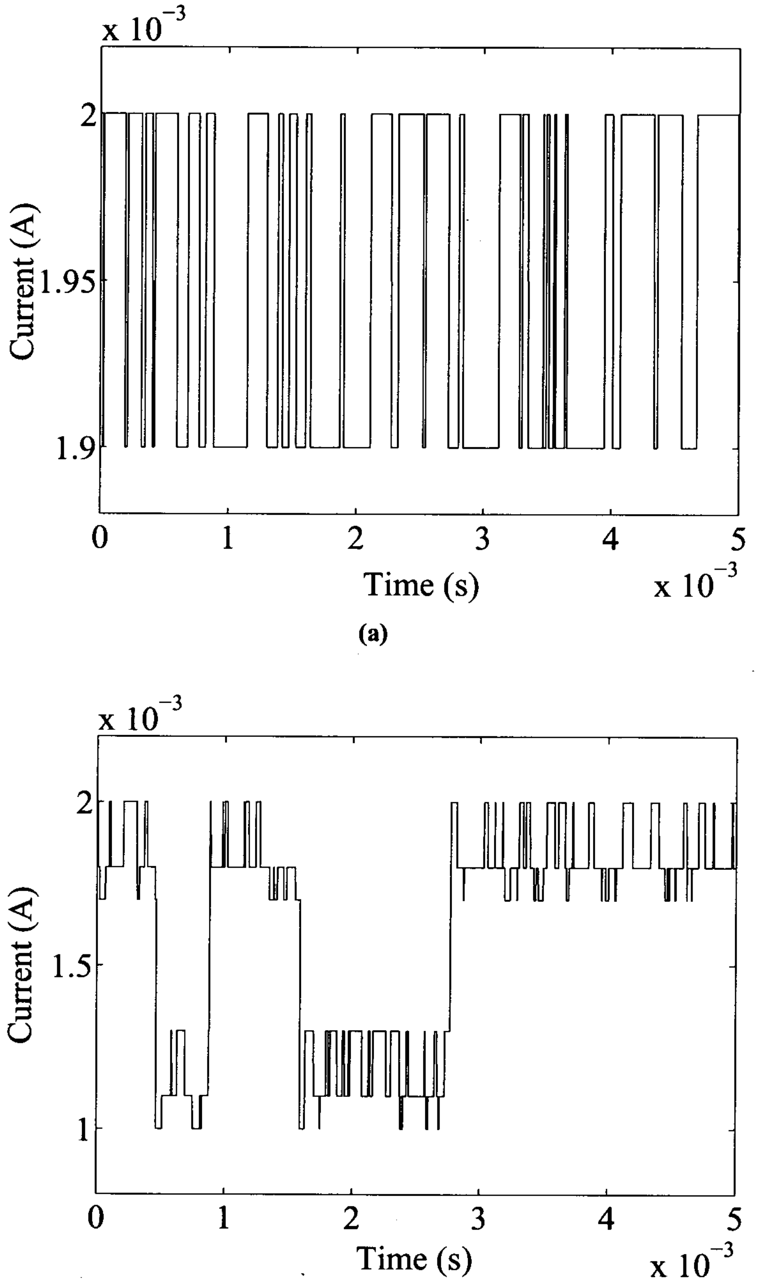
Fig.3.5(a) and Fig.3.5(b) are examples of single trap and multi(3)-trap RTS noise generated using the amplitude given in (2.2), and the transition time generated using Eq.(3.20).

3.4.5 Implementation: RTS noise

In this initial study, the position of the single trap in each noisy MOSFET channel is set arbitrarily to an appropriate lateral position and depth between source and drain. Trap depth x_t is selected from Table 3.1. These are experimental values [4, 5, 28]. In order to use the trap depth x_t for the $35nm$ MOSFET model, the values were scaled down by a factor of 10 to cater for a thin gate oxide ($\sim 7-8$).

Depending on bias conditions, the transconductance g_m was calculated according to Eq.(3.4), Eq.(3.5), or Eq.(3.6). Using the calculated g_m and MOSFET parameters W, L, C_{ox}, and t_{ox}, the RTS noise amplitude ΔI_D for a trap located at (x_t, y_t) was generated based on Eq.(2.2).

At the same time, the mean capture $\bar{\tau}_c$ and mean emission $\bar{\tau}_e$ time were determined according to Eq.(2.3) and Eq.(2.4), respectively. σ_0, ΔE_B, and ΔE_{CT} are selected from Table 3.1 for the appropriate value of the trap depth x_t. It is assumed that ΔE_B is a fixed value for all bias



(a)

(b)

Figure 3.5: *(a) Single trap RTS noise. (b) multi(3)-trap RTS noise.*

conditions. This is a reasonable assumption as ΔE_B has shown no dependence on bias [4]. On the other hand, ΔE_{CT} was reported to show dependence on both gate voltage V_{GS} and drain voltage V_{DS} [4]. This dependence is described by [4]:

$$\delta\left(\Delta E_{CT}\right) = \frac{q\left(\delta V_{GS} - \delta\psi_S\right)x_t}{t_{tox}} + q\delta\psi_S, \tag{3.21}$$

where $\delta(\Delta E_{CT})$ and δV_{GS} are calculated with reference to the values (ΔE_{CT} and V_{GS}) obtained from Table 3.1 for the trap depth x_t selected. The reference V_{GS} from Table 3.1 was scaled ($V_{GS} \times \frac{1.5}{3.5}$) to cater for lower operating voltage for the $35nm$ based implementation. $\delta\psi_S$ is the change in surface potential with reference to initial ψ_S, calculated based on a given trap lateral location y_t and drain voltage V_{DS} .

Similarly, σ_0 depends strongly on the gate voltage V_{GS} but to the best of our knowledge, no explicit mathematical model of the relationship has yet been reported. For simplicity, σ_0 was set to be a fixed parameter for all bias conditions. It is acknowledged that this assumption leads to a less accurate model; however, for the purpose of this work, this is sufficient. The channel electron concentration n is calculated from Eq.(3.11). The quasi-Fermi potential (V_{y_t}) and surface potential ψ_S were calculated for the trap location (x_t, y_t) from equation Eq.(A.1, A.2, A.4) and Eq.(A.6), respectively, in Appendix A.

The average capture and emission times were used to determine the transition from Eq.(3.20). As stated in the assumptions, a trap can either be filled or emptied at a particular time instance t. The initial state (fill or empty) is random. Once the trap state is determined, the transition probability $p(t_{Tran})$ is generated. The probability $p(t_{Tran})$ is used in Eq.(3.20) to determine the transition time t_{Tran}. If the simulation time t coincides with transition time t_{Tran}, a transition will take place. During this transition time, if an empty trap is filled, ΔI_D is deducted from the drain current I_D, and vice versa.

The implementation steps for generating time domain RTS noise are summarised as follows:

1. Select the number of traps.

2. Assign to each trap the depth x_t, the trap lateral y_t, and the corresponding fitting parameters (σ_0, ΔE_B, ΔE_{CT}), based on values in Table 3.1.

3. Determine the current time t:

   - If $t = 0$ (initialisation)

     (a) Probe the noiseless MOSFET drain to source current I_{DS}.

     (b) Randomly generate the trap(s) initial state (empty or full).

   - If $t \neq 0$ (ongoing)

     (a) Determine the trap(s) last state (empty or full).

     (b) Determine the previous effective drain to source current I_{DS}.

4. Calculate transconductance g_m using Eq.(3.4), Eq.(3.5), or Eq.(3.6), depending on the operating regime of the noiseless MOSFET.

5. For each trap:

   (a) Calculate RTS amplitude ΔI_{DS} using transconductance g_m and Eq.(2.2).

   (b) Calculate the electron concentration n using Eq.(3.11).

   (c) Calculate the effective trap binding energy $\Delta E_{CT(eff)}$ by subtracting Eq.(3.21) from the initial ΔE_{CT} (Table 3.1), i.e. $\Delta E_{CT(eff)} = \Delta E_{CT} - \delta\left(\Delta E_{CT}\right)$.

   (d) Calculate the mean capture $\bar{\tau}_c$ and the emission $\bar{\tau}_e$ time using parameters in Table 3.1, and Eq.(2.3) and Eq.(2.4) respectively.

   (e) Calculate the capture $p_c(t)$ and emission $p_e(t)$ probability using the calculated $\bar{\tau}_c$ and $\bar{\tau}_e$, and Eq.(3.13) and Eq.(3.14) respectively.

   (f) Approximate the transition time t_{Tran} using the calculated capture $p_c(t)$ and emission $p_e(t)$ probability and Eq.(??).

6. Calculating the effective I_{DS} at current time t.

   - If current time t = transition time t_{Tran} or 0 (initial)[1] :

     (a) If the trap(s) was(were) empty, subtract ΔI_{DS} of the empty trap from I_{DS} and set the trap(s) to full.

[1] The initial ($t = 0$) RTS amplitude ΔI_{DS} of each trap was calculated based on noiseless I_{DS} (or empty trap). Therefore, the initial noisy I_{DS} is not accurate if the initial state of the trap(s) (generated randomly) was(were) not empty.

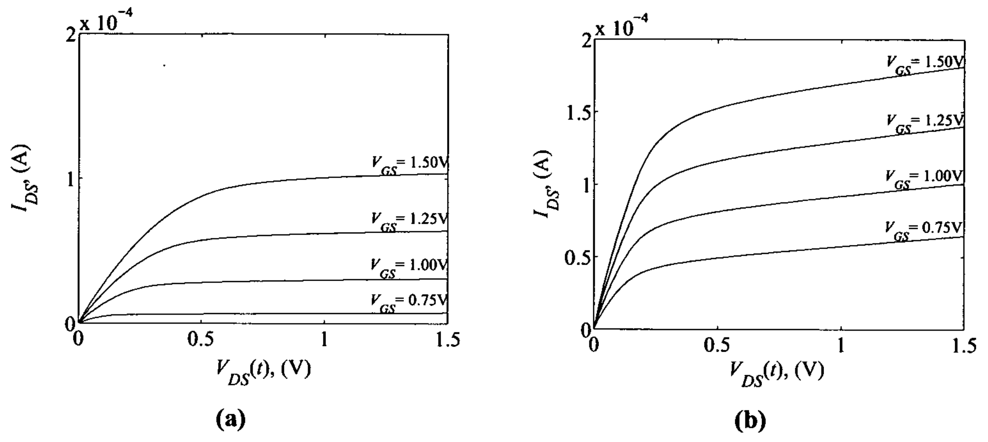


Figure 3.6: *I-V characteristic for (a) 0.35μm (L=$0.35\mu m$, W=$1\mu m$), and (b) 35 nm (L=$35nm$, W=$0.1\mu m$) 'noiseless' MOSFET.*

(b) If the trap(s) was(were) full, add ΔI_{DS} of the full trap to I_{DS} and set the trap(s) to empty.

- Else, *Go to step* 7.

7. Repeat 1-6 until end of simulation time.

3.5 Results and Comparison: $1/f$ and RTS Noise

To explore and visualise the noise signals, the noisy MOSFET was simulated via transient analysis with the gate and drain biased between 0 and 1.5 Volts, while the source and body were fixed at 0 Volts. All analyses were performed at room temperature (27°C).

Initially, the noiseless I-V characteristic was determined by setting the amplitude of the noise source $n(t)$ to zero. The drain voltage V_{DS} was swept from 0 to 1.5 Volts in 1 second for each gate voltage, V_{GS}. Fig.3.6 shows the I-V characteristics for noiseless $0.35\mu m$ and $35nm$ NMOS implemented using an AMS CMOS technology and an *atomistic-based* CMOS technology, respectively. The I-V characteristics of the $0.35\mu m$ NMOS and $35nm$ NMOS clearly have the correct form. The implementation of the noisy MOSFETs has not changed the underlying I-V characteristics of the $0.35\mu m$ and $35nm$ NMOS.

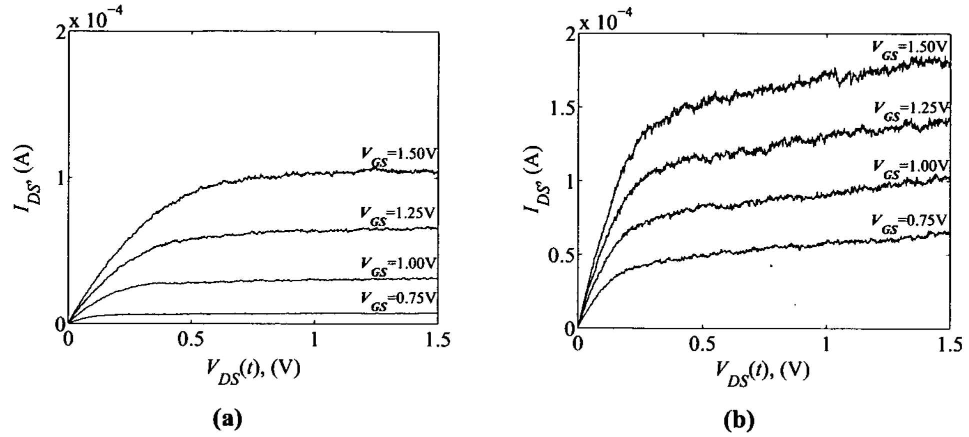
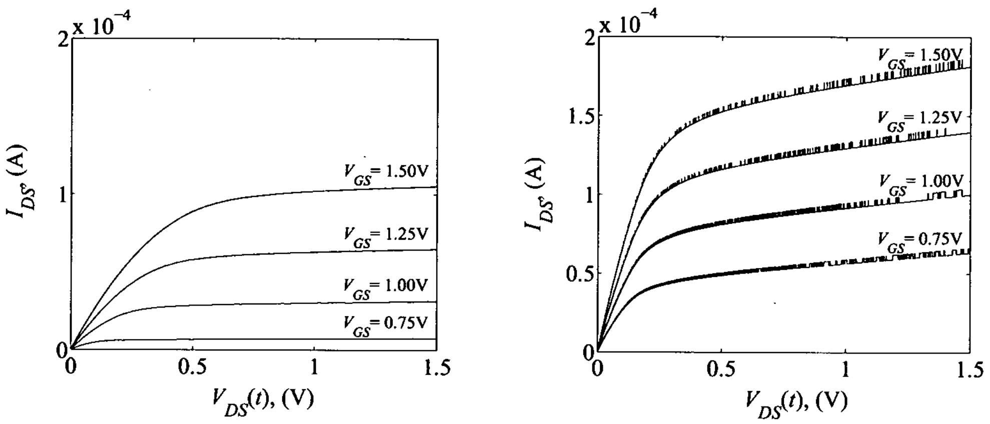


Figure 3.7: *I-V characteristic for $1/f$ based noisy (a) 0.35μm (L=$0.35\mu m$, W=$1\mu m$), and (b) 35 nm (L=$35nm$, W=$0.1\mu m$) NMOS.*

The characteristics of the $1/f$-based and RTS-based noisy $0.35\mu m$ NMOS, and noisy $35nm$ NMOS transistors are shown in Figs.3.7 and 3.8, respectively. The impact of device down-scaling is evident. The noise amplitude generated by noisy $35nm$ NMOS is almost double the noise amplitude generated by noisy $0.35\mu m$ NMOS. This observation becomes prominent in Fig.3.9.

It is vital to confirm that the noise generated has the correct spectral form given in Eq.(2.7) and Eq.(2.5). The noisy MOSFET was therefore simulated for a fixed combination of drain (V_{DS}) and gate (V_{GS}) voltage (i.e. fixing the I_{DS}). Fig.3.9 shows the time domain noise for the $1/f$ based noisy $0.35\mu m$ and $35nm$ MOSFET, while Fig.3.10 shows the RTS noise in the $35nm$ MOSFET. The time-domain noises show the correct form of noise generated. The corresponding PSDs (extracted using periodogram function in Matlab) are shown in Fig.3.11 and Fig.3.12, respectively. The PSDs generated for the time-domain noises show the correct $1/f$ and RTS characteristics match the PSDs predicted by Eq.(2.7) and Eq.(2.5).

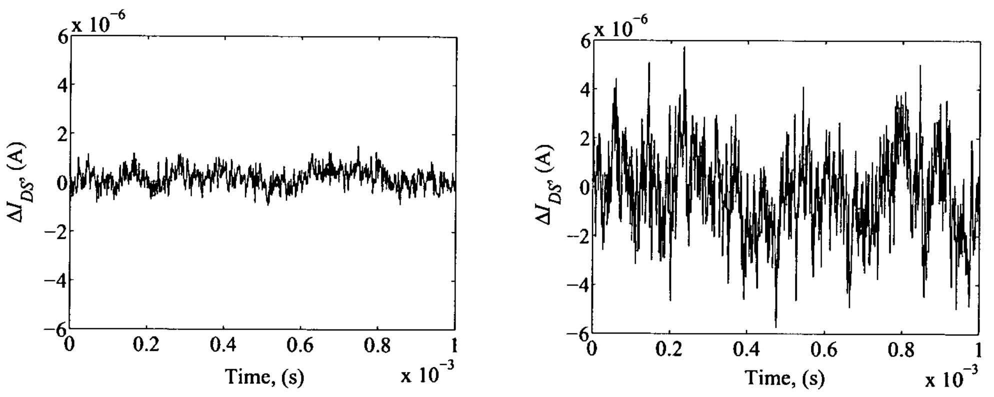
In order to understand how the trap position affects the noise generated in RTS based noisy MOSFETs, a simulation .was setup to further analyse the dependence of RTS amplitude, mean capture $\bar{\tau}_c$, and emission $\bar{\tau}_e$ time on trap location (x_t, y_t). The dependence of RTS amplitude on trap depth and lateral location are shown in Figs.3.13(a) and (b), respectively. Deeper traps produce smaller noise amplitude while lateral position does not influence noise



(a)

(b)

Figure 3.8: *I-V characteristics for RTS based noisy (a) 0.35μm (with L=0.35μm, W=1μm, trap depth x_t = 1.1nm and lateral location y_t = 160nm) and (b) 35 nm (L=35nm, W=0.1μm, trap depth x_t = 0.11nm and lateral location y_t = 16nm) MOSFETs. The values σ_0, ΔE_B, ΔE_{CT} and trap V_{GS} were selected from Table3.1 corresponding to trap depth x_t 1.1nm.*



(a)

Figure 3.9: *Time domain noise generated by (a) 0.35μm and (b) 35 nm noisy MOSFET simulated at static bias conditions (V_{DS} = 1.5V and V_{GS} = 1V) for 1ms.*

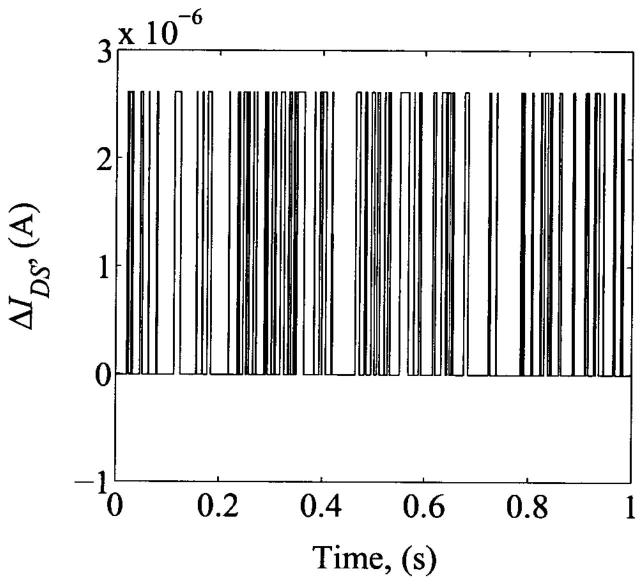


Figure 3.10: *The Drain-Source current noise (ΔI_{DS}) for a $35nm$ (L=$35nm$, W=$0.1\mu m$) single trap RTS, based noisy MOSFET biased at $V_{GS} = 1.0V$ and $V_{DS} = 0.6V$. The drain current with DC value ($I_{DS} = 84\mu A$) removed. (Note that a longer simulation time (1 second) was needed in order to capture more RTS noise). Based on the fixed bias conditions, ΔI_{DS}, $\bar{\tau}_c$ and $\bar{\tau}_e$ in the RTS based noisy MOSFET were generated to be 2.62e-6 A, 2.918e-3 s, and 1.279e-2 s, respectively.*

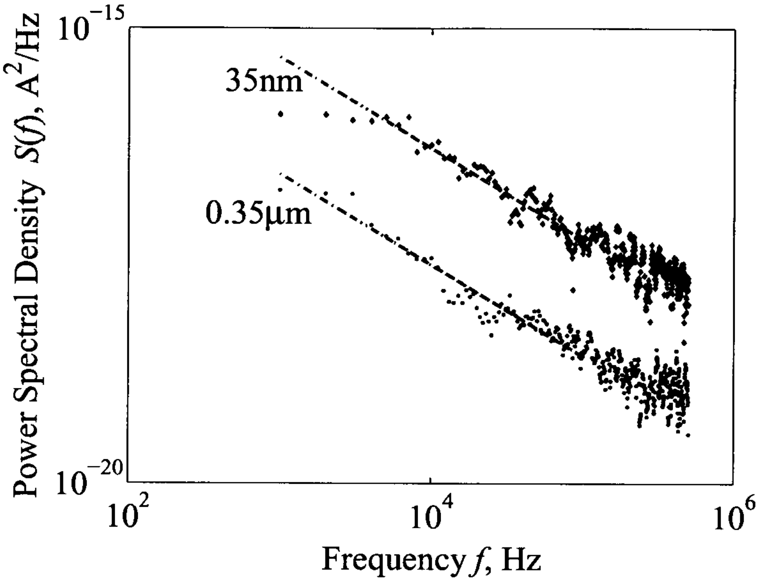


Figure 3.11: *The Power Spectral Density (PSD), $S(f)$ corresponding to the time domain noises shown in Fig.3.9. The dash-dot lines are $1/f$ PSDs generated using Eq.(2.7).*

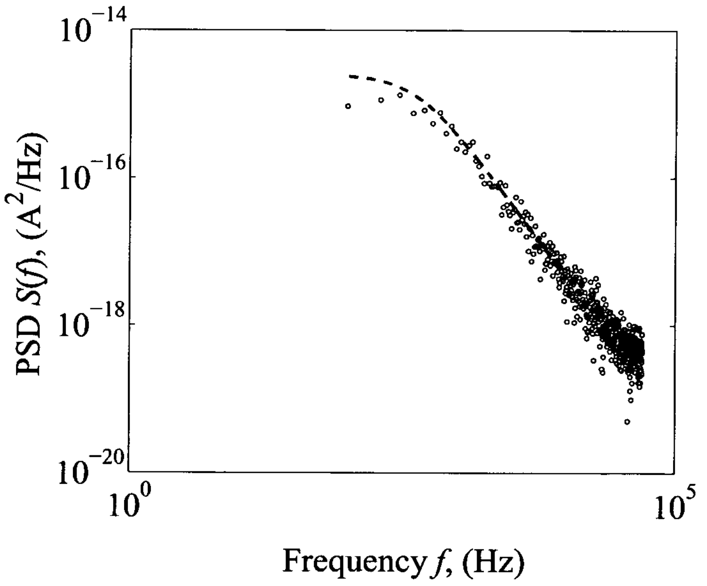


Figure 3.12: *The Power Spectral Density (PSD), $S(f)$ of a single trap RTS plotted in Fig.3.10. The dashed-line is the calculated PSD based on Eq.(2.5).*

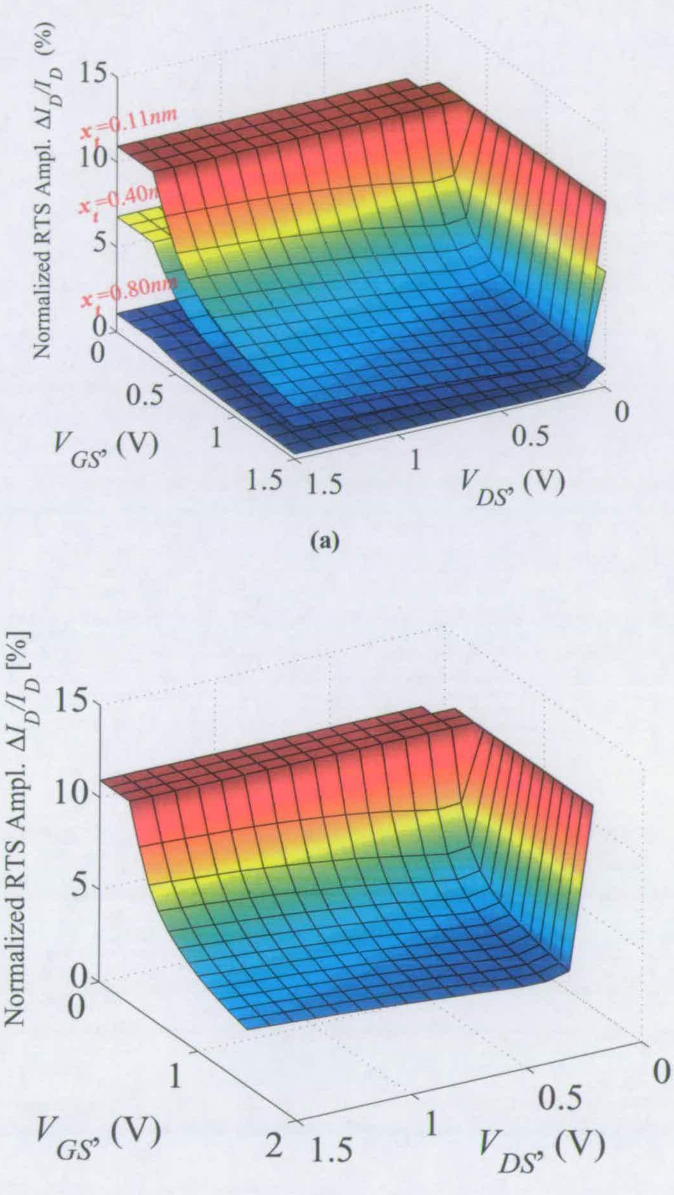
amplitude. RTS noise amplitude shows strong gate voltage (V_{GS}) dependence, where at weak inversion, the amplitude peaks. These results are consistent with the characteristics reported in [27, 41, 57].

From the same simulation, the mean capture and emission times were plotted in Fig.3.14 and Fig.3.15. Trap depth has a significant effect on mean emission time dependence on gate voltage, while mean capture time appears unaffected. At weak inversion (low V_{GS}), deeper traps are unable to retain electrons, however, as the voltage increases, the retention time becomes longer. Trap lateral position influences the dependence of mean capture $\bar{\tau}_c$ and emission $\bar{\tau}_e$ time on drain voltage (V_{DS}). $\bar{\tau}_c$ and $\bar{\tau}_e$ for a trap located near the source appear to be independent of V_{DS} while dependence become very obvious for a trap located near the drain. This effect corresponds to drain bias influence on channel electron concentration described by Eq.(3.11).

Finally, multi-trap RTS noise was studied in the same context. 3-trap RTS noise and 10-trap RTS noise was injected into MOSFETs. The variation between traps was based on implementing different trap depth (x_t) and lateral positioning (y_t). Figs.3.16(a) and (b) show 3-trap RTS noise and 10-trap RTS noise simulated for 1s. The number of noise discrete levels increases with the number of traps. However, due to limited simulation time, the full spread of current level (i.e. for 3-trap RTS, 8 levels are expected) is not seen. Using the Matlab periodogram function, the PSD of the 3-trap and 10-trap RTS noise was extracted and plotted in Fig.3.17. The PSDs approach $1/f$ PSDs as the number of traps increases.

3.6 Discussion: Noise Modeling

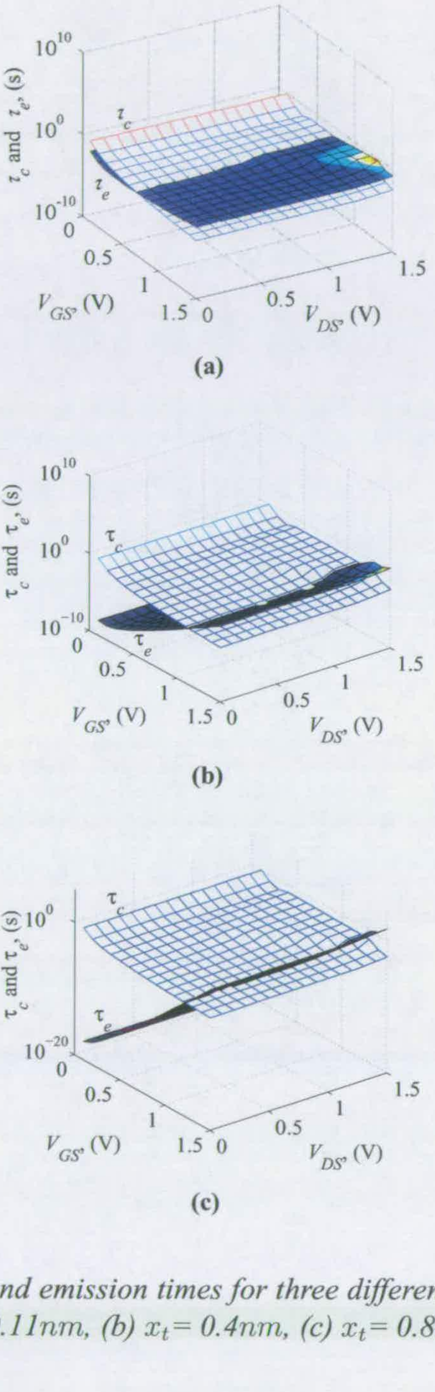
Both $1/f$ and RTS- noisy MOSFET models have been implemented in Verilog-A. The methodology used to generate $1/f$ noise is based on the sum-of-sinusoid approximation [52, 53] while RTS noise is generated based on both noise parameters (amplitude and time statistics) and Monte-Carlo simulation [57]. It was found that while the sum-of-sinusoids technique is simple and easy to implement, extensive computing power is required to generate a better representation of $1/f$ noise. The technique used to generate RTS noise is more intuitive and straightforward to implement. However, as the number of traps increases, the implementation



(a)

(b)

Figure 3.13: *(a) Normalised ($\Delta I_D/I_D$) RTS amplitude for three different trap depths x_t= 0.11nm, 0.4nm, 0.8nm with trap lateral position (measured from source) fixed at y_t= 16nm. (b) Normalised ($\Delta I_D/I_D$) RTS amplitude for three different trap lateral positions (measured from source) y_t= 5nm, 16nm, 30nm with trap depth fixed at x_t= 0.11nm. The simulation was based upon parameters corresponding to $x_t = 1.1nm$ in Table 3.1.*

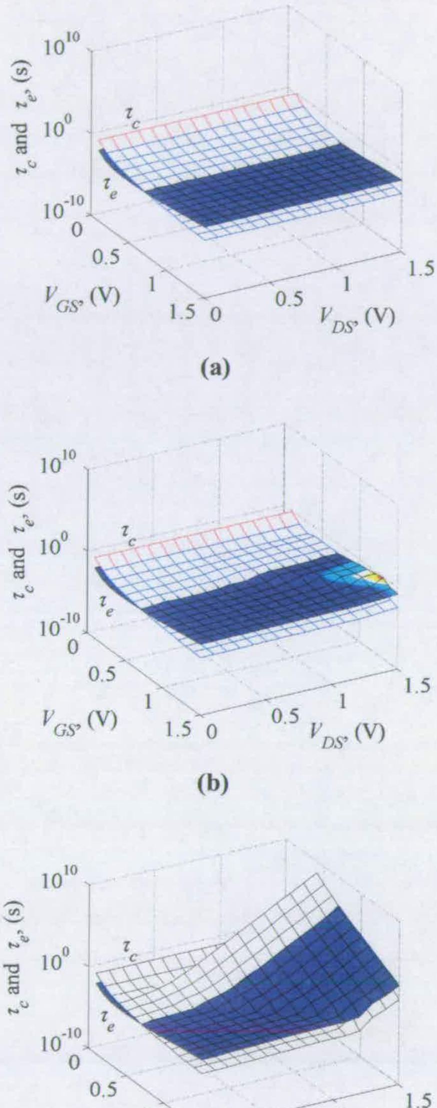


(a)

(b)

(c)

Figure 3.14: *Mean capture and emission times for three different trap depths with fixed y_t= 16nm (a) x_t= 0.11nm, (b) x_t= 0.4nm, (c) x_t= 0.8nm.*



(a)

(b)

(c)

Figure 3.15: *Mean capture and emission times for three different trap lateral locations with fixed x_t= 0.11nm (a) y_t= 5nm, (b) y_t= 16nm, (c) y_t= 30nm.*

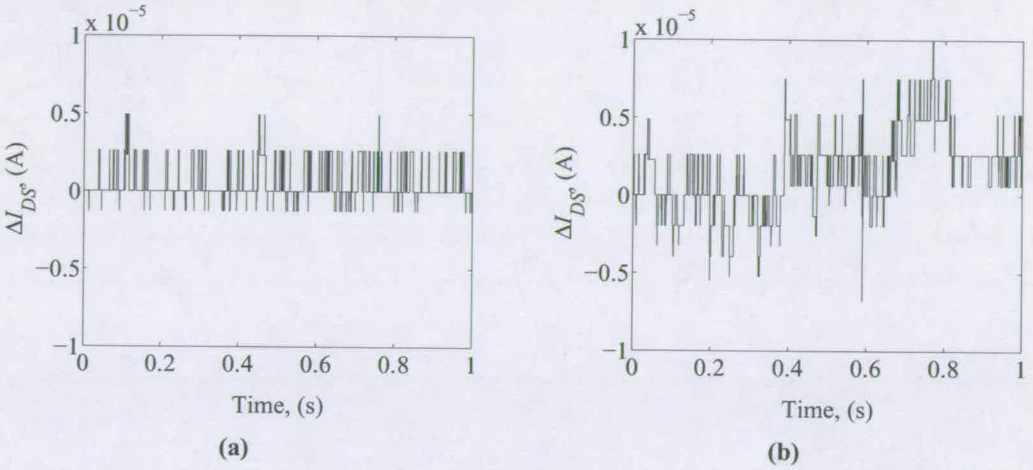


Figure 3.16: *RTS noise generated by noisy MOSFET implemented with (a) 3 Traps and (b) 10 Traps. Trap parameters were based on the value in Table 3.1 corresponding to $x_t = 1.1nm$. V_{DS} and V_{GS} were set to 0.6V and 1V respectively.*

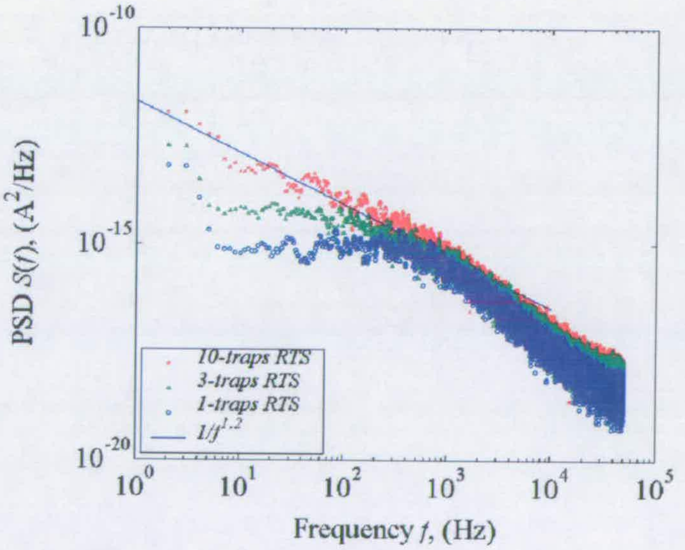


Figure 3.17: *Power Spectral Density of 1-trap, 3-trap, and 10-trap RTS noise. Dashed-line indicates the $1/f$ noise PSD.*

becomes increasingly complex.

It has been shown so far that the techniques used are capable of generating the correct noise form ($1/f$ or RTS). It has been confirmed that PSDs of generated time-domain noises match the predicted PSDs in Eq.(2.7) and Eq.(2.5). It has been noted that superposition of the RTS noise produces $1/f$ noise, in agreement with finding reported by [4, 11, 45]. Multi-trap (3 and 10 traps) RTS noisy MOSFETs were implemented. The time domain plots of these multi-trap noisy MOSFETs (Fig.3.16) show multi-level switching as expected. Fig.3.17 compares the PSDs generated from these time-domain RTS noises. Using power series non-linear least square fitting function (in Matlab), the 10-trap RTS noise fitted well to a theoretical $1/f^{1.2}$ PSD. This suggests that as the number of switching levels increase (i.e. increase number of traps), the generated noise becomes more like the $1/f$.

3.7 Summary

Low frequency noise will have dramatic effects on future nanoscale MOSFETs and circuits. A noisy MOSFET has been modelled to emulate the predicted noisy behaviour of future nanoscale MOSFETs. Both $1/f$ and Random Telegraph Signal (RTS) noise have been studied. A pragmatic approach has been taken to include the effect of this form of noise in MOSFETs, such that the modelled noise can be included in circuit simulations. In effect, a methodology has been put in place that builds a modelling bridge between atomistic models of DSM devices (and the physical devices upon which they were based), through look-up table models of the noise in such devices, to circuit models of noisy DSM devices and circuits that can be simulated in reasonable time and using conventional analogue simulators. The methodology opens up the opportunity to explore and investigate the effect of noise in nanoscale MOSFET circuits.

Chapter 4
Noisy Circuit Implementation

In this chapter, the effects of nanoscale MOSFET noise on circuit performance are explored. For the reason that will be discussed in Chapter 6, the implementation of the noisy circuit focuses on an analogue multiplier as the benchmark architecture. A 2-quadrant analogue multiplier described in [61] will be used in the implementation. With a simple modification [3], a stable 4-quadrant multiplier will be implemented. The noisy 2-quadrant and 4-quadrant analogue multipliers will be implemented by replacing the key MOSFETs with the noisy MOSFET models developed in Chapter 3. The performance of these noisy analogue multipliers will be presented and discussed.

4.1 Noisy 2-Quadrant Multiplier

Analogue multipliers are used extensively in almost all forms of neural architecture, representing, primarily, the effect of synaptic gating. Many forms of multiplier have been used [61–64]; however, simple circuit form and a current-mode output are essential for reducing power- and area- consumption and introducing scaling flexibility in massively-parallel neural architectures [32].

The 2-quadrant multiplier discussed in [61] enjoys a wide input range and simple design [61], without having to have additional biasing circuitry [3]. These are the preferable attributes for neural architecture hardware implementation, and therefore become the basis of our choice in this chapter.

4.1.1 Circuit Description

Fig.4.1 shows a 2-quadrant Chible multiplier configured as a synaptic (weight) multiplier. The weight V_w voltage determines I_1, which is then multiplied by V_{in}, referenced to V_{ref}, to

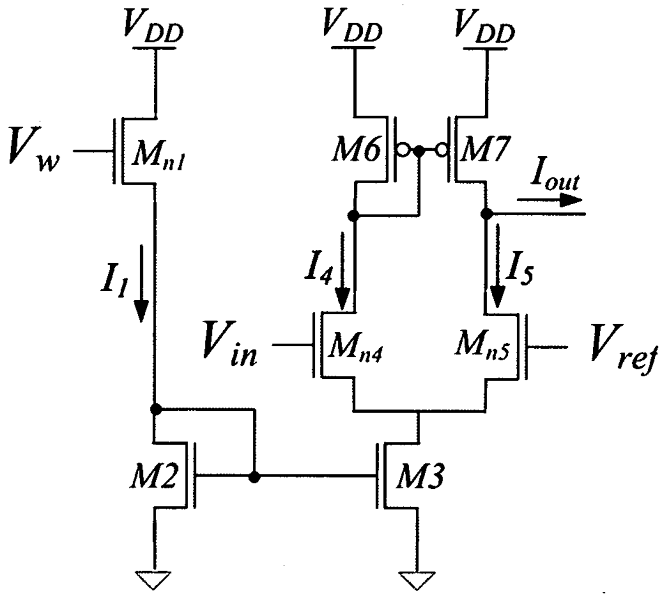


Figure 4.1: *2-quadrant Chible multiplier.*

produce I_{out}.

All MOSFETs are in strong inversion and in saturation, so the output current I_{out} is given by (channel length modulation is neglected) [61]:

$$I_{out} = \sqrt{\frac{\beta_n \beta_{4,5}}{n^2}} \left(V_w - nV_{TH2} - V_{TH1}\right)\left(V_{in} - V_{ref}\right), \tag{4.1}$$

where $\beta_{4,5}$ refers to the transfer parameters for M_{n4} and M_{n5}, respectively and β_n is given by:

$$\frac{1}{\sqrt{\beta_n}} = \frac{1}{\sqrt{\beta_1}} + n \times \frac{1}{\sqrt{\beta_2}}. \tag{4.2}$$

β_1 and β_1 are the transfer parameters for M_{n1} and *M2* respectively. The transfer parameters are defined as $\beta_1 = \mu_1 \times C_{ox}\left(\frac{W}{L}\right)_1$ and $\beta_2 = \mu_2 \times C_{ox}\left(\frac{W}{L}\right)_2$. V_{TH1} and V_{TH2} are the threshold voltage for M_{n1} and *M2*, respectively, and n is the slope factor usually smaller than 2 which tends to 1 for very large values of gate voltage [61]. $V_w > V_{TH1} + V_{TH1}$ must be satisfied.

4.1.2 Circuit Implementation

The 2-quadrant multiplier was implemented in an artificially-modelled $35nm$ CMOS technology. MOSFETs M_{n1}, M_{n4}, and M_{n5} are modelled as noisy MOSFETs, to introduce noise to the signal path without compromising the performance of the multiplier. The current mirrors remain noise-free. This caveat is reasonable, as *M2*, *M3*, *M6*, and *M7* are large, to provide good transfer characteristics.

4.1.3 Simulation Results and Discussion

The 2-quadrant multiplier shown in Fig.4.1 has been simulated using the SPECTRE simulator with BSIM3v3 version 3.1 models. V_{DD} was set to $1.5V$ and V_{ref} to $0.75V$. Using transient analysis, the circuit was simulated by sweeping the *weight* voltage V_w from $0V$ to $1.5V$ for each *input* voltage V_{in} which is also varied between $0.5V$ and $0.9V$ with a $0.025V$ step size. The output current for no injected noise I_{out} are shown in Fig.4.2. The output current I_{out} shown in Fig.4.2 matches the behaviour predicted by Eq.(4.1). However, a slight deviation can be observed for V_{in}=V_{ref}, where $I_{out} \neq 0$ as predicted by Eq.(4.1). This observation is attributed to the effect of channel length modulation [49, 59] in *M6-M7*. A remedy to this problem is to use a cascode current mirror in place of *M6-M7* and increase the gate length of the current mirror MOSFETs.

In Fig.4.3(a) shows the output of a $1/f$ noisy multiplier and Fig.4.3(b) shows that of an RTS noisy multiplier implemented using $1/f$ and single-trap RTS noisy MOSFET models respectively. In Fig.4.3(a) and Fig.4.3(b), the multiplier output noises depict dependence on both V_w and V_{in} with significant noise amplitude observed for $V_w = 1.5V$ and $V_{in} = 0.7V$.

In order to analyse the output current noise characteristic, the noisy 2-quadrant multiplier was simulated using fixed bias transient analysis. Fig.4.4(a) and Fig.4.4(b) show the noisy output current, I_{out}, for implementation using $1/f$-based noisy MOSFET models and single-trap RTS-based noisy MOSFET models respectively. There is, however, no correlation between the amplitude produced by the $1/f$- and RTS-based implementations as the parameters used to generate $1/f$ noise were artificially scaled to produce statistically significant noise amplitudes. The output current noises shown in Fig.4.4(a) and Fig.4.4(b) exhibit the time domain

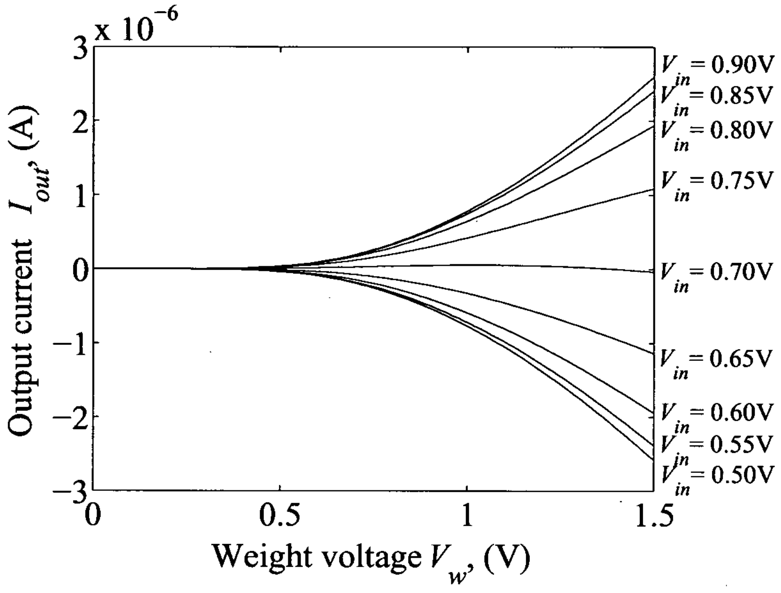


Figure 4.2: *35nm CMOS technology 2-quadrant multiplier output current I_{out}.*

noise characteristics expected from each implementation which are confirmed by their PSD plots as shown in Fig.4.5.

For RTS, the output noise produces 6-7 discrete levels, corresponding to 3-traps activity injected into the multiplier through M_{n1}, M_{n4}, and M_{n5}. Ideally, 8 discrete levels are expected, however for the given simulation time, it might not be possible to see them all. The PSDs (Fig.4.5) show that the output current noise inherits the characteristic noise behaviour of the noisy MOSFETs used.

4.2 Noisy 4-Quadrant Multiplier

Chible [61] suggested that a 4-quadrant multiplier can be implemented by augmenting the 2-quadrant multiplier discussed in Sec.4.1 . However, a major drawback of this suggestion is that the circuit's reference zero is dependent on the threshold-voltage, and is therefore process-dependent [3]. This is clearly evident in Eq.(4.1). The lack of a unique reference zero discourages the precise mapping of parameter values between the hardware implementation and the behavioural model in the Matlab simulation. To cater for this problem, a modified

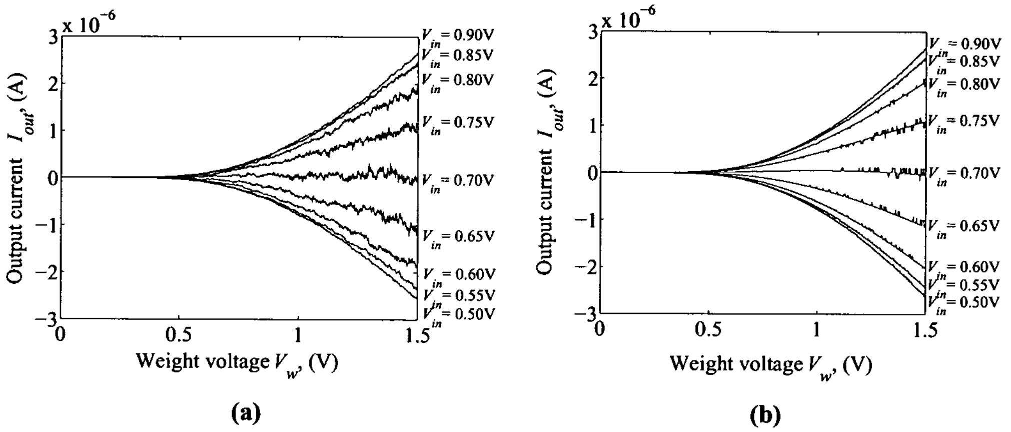


Figure 4.3: *35nm CMOS technology 2-quadrant multiplier output current with M_{n1}, M_{n4}, and M_{n5} replaced with (a) $1/f$ and (b) single trap RTS based noisy n-MOSFETs.*

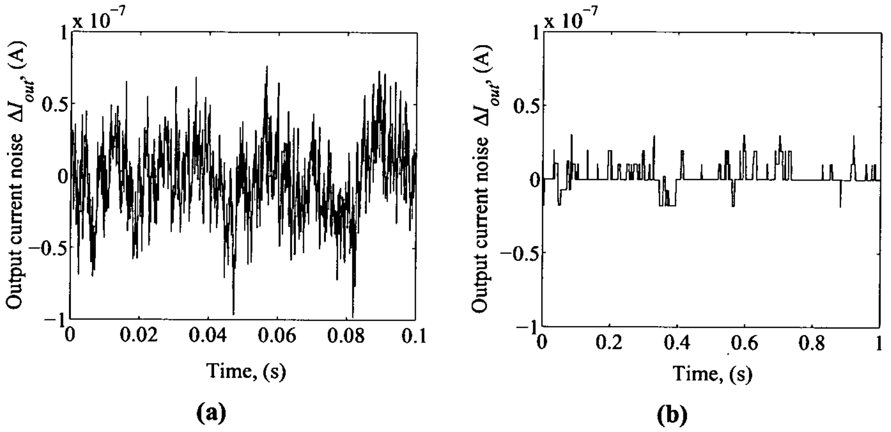


Figure 4.4: *(a) $1/f$ and (b) single trap RTS based noisy MOSFET transient plot with V_{DD}, V_w, V_{in}, V_{ref}were set to 1.5V, 1V, 0.6V, and 0.75V, respectively. Note that longer simulation time was necessary for RTS noise in order to capture more noise data.*

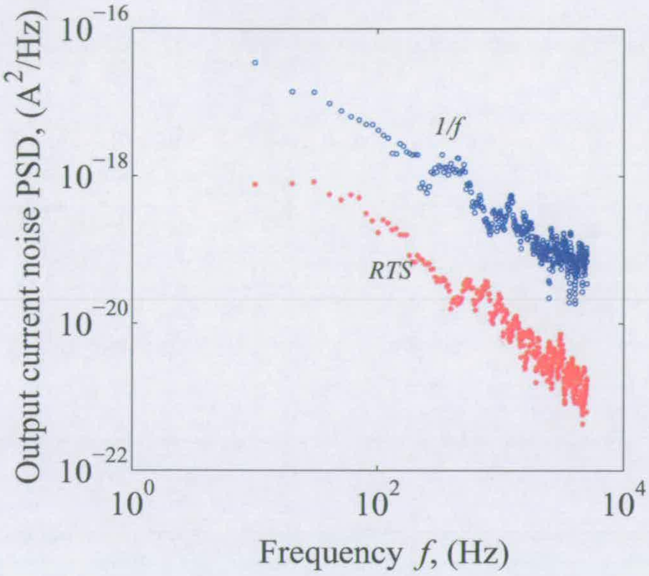


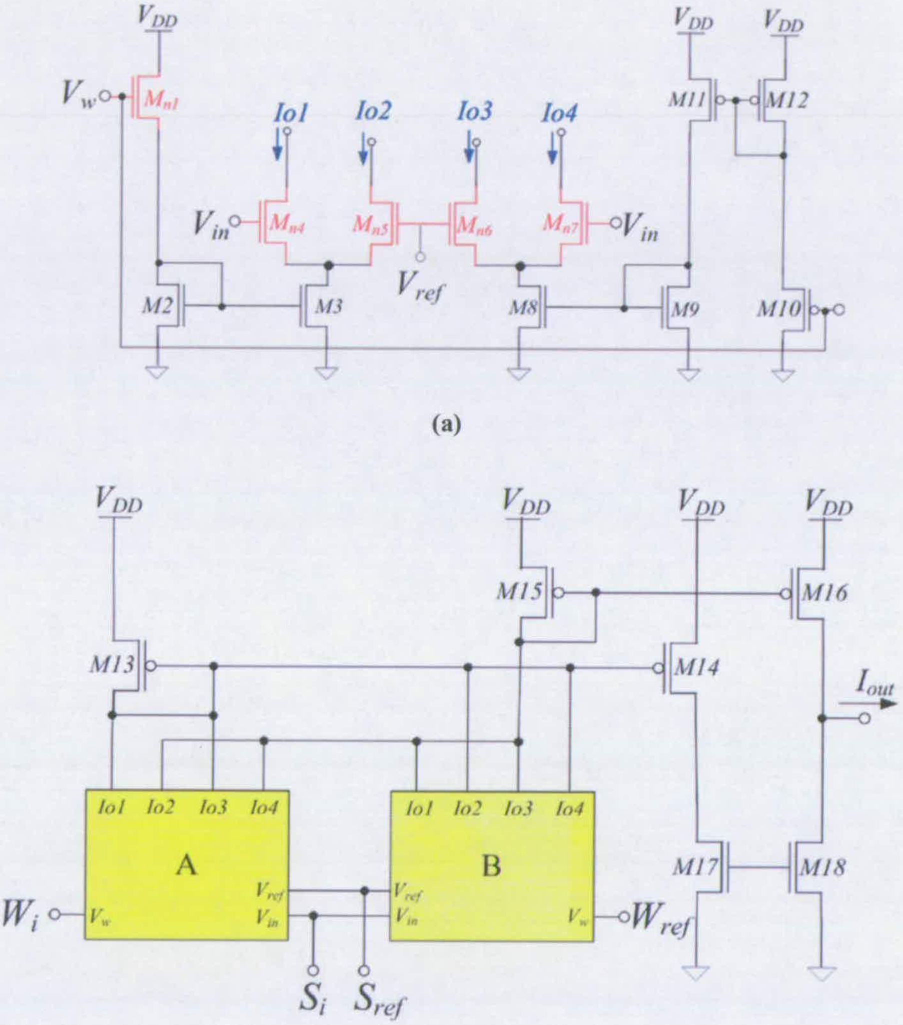
Figure 4.5: *Power spectral density generated from time domain noise data in Fig.4.4.*

Chible multiplier has been implemented in [3] with two identical computing cells, each of which corresponds to the Chible multiplier proposed in [61]. In this implementation, the reference zero can be externally set by the external inputs, W_{ref} and S_{ref}. Due to its simple architecture and reliable performance, the modified Chible multiplier (Fig.4.6(b)) is adopted for this project. However, the modified Chible multiplier's computing cell architecture is changed slightly here to cater for the NMOS-based implementation, as the noisy MOSFET developed in Chapter 3 is NMOS. The computing cell is shown in Fig.4.6(a).

4.2.1 Circuit Description

A detailed description of the modified Chible multiplier circuit can be found in [3, 32]. The change made in this project on the computing cell's architecture does not change the overall multiplier behaviour. From [3, 32], the output current I_{out} is described as:

$$I_{out} = \begin{cases} K_N\left(W_i - W_{ref}\right) \cdot \left(S_i - S_{ref}\right), & \text{if } W_i > W_{ref} \\ \\ K_P\left(W_i - W_{ref}\right) \cdot \left(S_i - S_{ref}\right), & \text{if } W_i < W_{ref} \end{cases} \tag{4.3}$$



(a)

(b)

Figure 4.6: *The modified Chible 4-quadrant multiplier adopted from [3]. (a) one- computing cell of the modified Chible multiplier (b) the full 4-quadrant multiplier circuit composed of two computing cells.*

where K_N and K_P are constants that depend on the size of differential pair $M_{n4} - M_{n5}$ and $M_{n6} - M_{n7}$, respectively.

4.2.2 Circuit Implementation

The 4-quadrant multiplier was implemented in an artificially-modelled $35nm$ CMOS technology. M_{n1}, M_{n4}, M_{n5}, M_{n6}, and M_{n7} in both cell A and cell B are modelled as noisy MOSFETs, to introduce noise to the signal path without compromising the performance of the multiplier, while the current mirrors remain noise-free. The effect of different numbers of traps was explored by implementing a 4-quadrant multiplier based on 1-trap noisy MOSFETs, and 10-trap noisy MOSFETs, making the total number of active traps in each implementation 10 and 100 respectively. In addition, a 4-trap, 4-quadrant multiplier (i.e. only 4 active traps in the full 4-quadrant multiplier) was also implemented to explore the effect of a small number of traps on output noise.

4.2.3 Simulation Results and Discussion

The 4-quadrant multiplier has been simulated using the SPECTRE simulator with BSIM3v3 version 3.1 models. V_{DD} was set to $1.5V$, W_{ref} to $0.75V$ and S_{ref} to $0.75V$. Using transient analysis, the circuit was simulated by sweeping *weight* voltage W_i from $0V$ to $1.5V$ for each *input* voltage S_i which was also varied between $0.55V$ and $0.95V$ with a $0.05V$ step size. The output current for no injected noise, I_{out}, is shown in Fig.4.7. The output current I_{out} shown in Fig.4.7 matches the behaviour predicted by Eq.(4.3).

To investigate the time domain noise characteristic, the noisy 4-quadrant multiplier was simulated using fixed-bias transient analysis. For this analysis, W_i is set to values within the range [0,1.5] (V) with a 0.15V step size, while $S_i$is set to value within the range [0.55,0.95] (V) with a 0.05V step size. For each combination of W_i and S_i, the multiplier was simulated for $250ms$ with a $1\mu s$ time step. In order to observe significant trap activities (capture and emission) within this 'short' simulation time, the trap cross-section pre-factor σ_0 values in the mean capture Eq.(2.3) and emission Eq.(2.4) time models described in Chapter 3 were

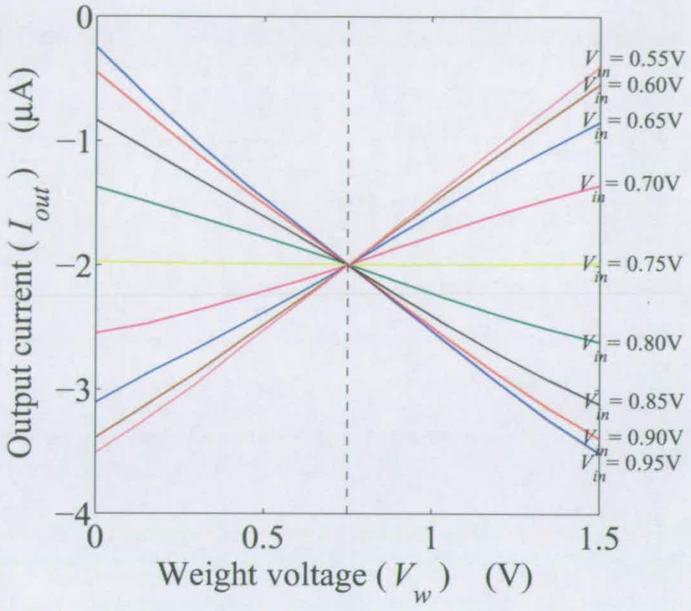
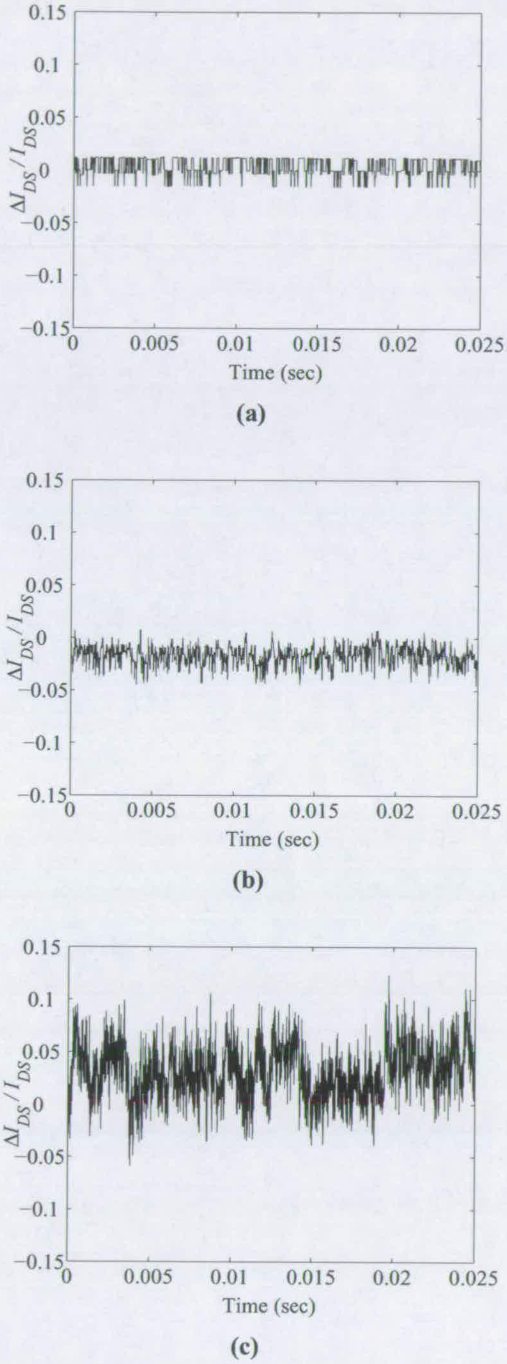


Figure 4.7: *35nm CMOS technology 4-quadrant multiplier output current I_{out}.*

increased arbitrarily. An example of time domain noisy 4-quadrant multiplier output noise is shown in Fig.4.8. The 4-trap implementation (Fig.4.8(a)) produces distinct levels of noise amplitudes, while 10-trap and 100-trap implementations (Fig.4.8(b) and Fig.4.8(c) respectively) produce a 'continuous' level of noise amplitudes; the expected results based on the discussion in Sect.2.2.4. Their corresponding PSD plots shown in Fig.4.9 confirm to the earlier findings that a large number of traps produces a noise closer to the $1/f$ characteristic. The peculiar (up-turn) characteristics observed at high frequency for 100-implementation is attributed to aliasing effect during PSD generation.

Figs.4.10(a-c) show the output noise amplitude from three different combinations of the 4-quadrant multiplier input implemented based on 4-traps. These results are typical examples of output noise amplitude variation caused by changing bias conditions.

The linear combination of normalised noise amplitude data for all input combinations generates distribution as shown in Fig.4.11(a). To perceive how noise amplitudes behave in comparison to Gaussian noise, a histogram-fit curve of Gaussian noise with $\sigma = 0.1$ generated using Matlab is also included. Similar plots are done for 10-trap and 100-trap based imple-



(a)

(b)

(c)

Figure 4.8: *The output current I_{out} for (a) 4-trap, (b) 10-trap, and (c) 100-trap 4-quadrant noisy multiplier simulated for 250ms with 1μs time step.*

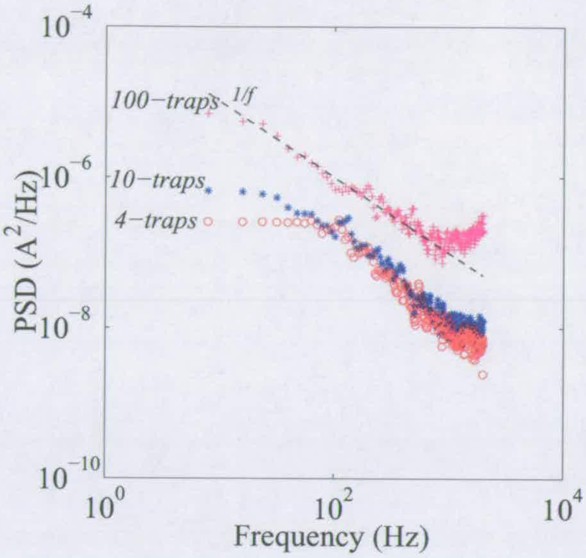


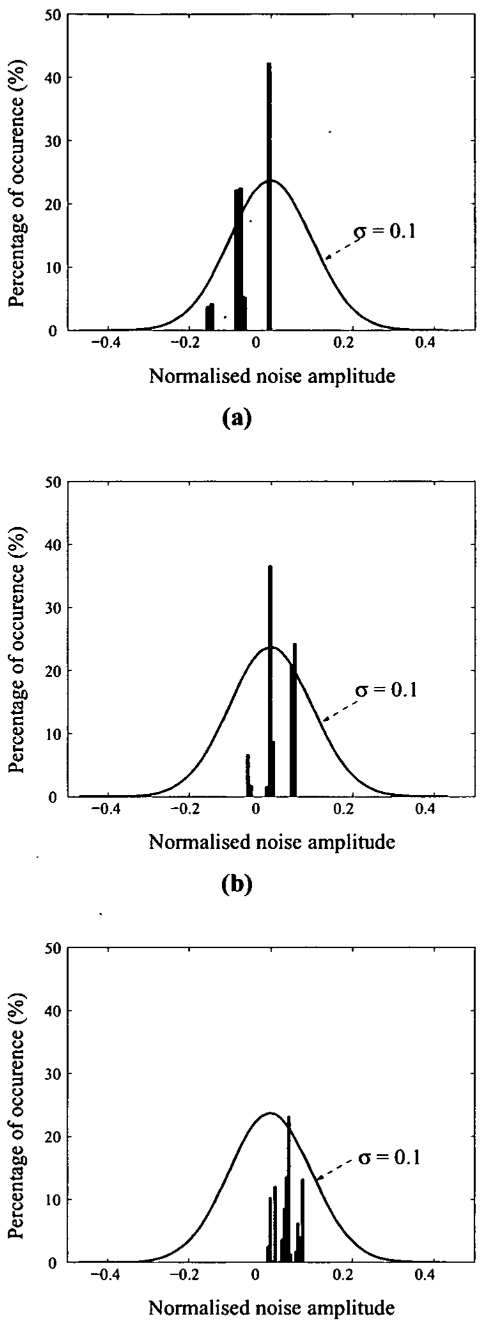
Figure 4.9: *The power spectral densities (PSDs) of the time domain noise data shown in Fig.4.8, generated using periodogram function in Matlab.*

mentation (Fig.4.11(b) and (c)). It is important to note that noise amplitudes are grouped as 16 levels only[1] for all RTS implementations (4, 10, and 100 traps) to visualise and compare the different implementation output noise as a histogram. The general noise amplitude distribution follows a Gaussian distribution and as expected; the 100-trap output noise amplitude distribution is almost Gaussian. This is consistent with the central limit theorem of mathematical statistics, that suggests the superposition of many independent random phenomena produces a Gaussian distribution [65]. Based on discussion in Sect.2.2.4, the large number of traps (i.e. 100 traps) give rise to $1/f$ noise.

4.3 Summary

Noisy 2-quadrant and 4-quadrant multipliers have been implemented and simulated using noisy MOSFET models placed at the strategic locations to introduce noisy products without degrading the original performance of the multiplier itself. Simulation results indicate that

[1] A 10-trap implementation can produce 1024 (2^{10}) possible levels, whilst a 100-trap implementation can produce 1.27×10^{30} (2^{100}) possible levels.

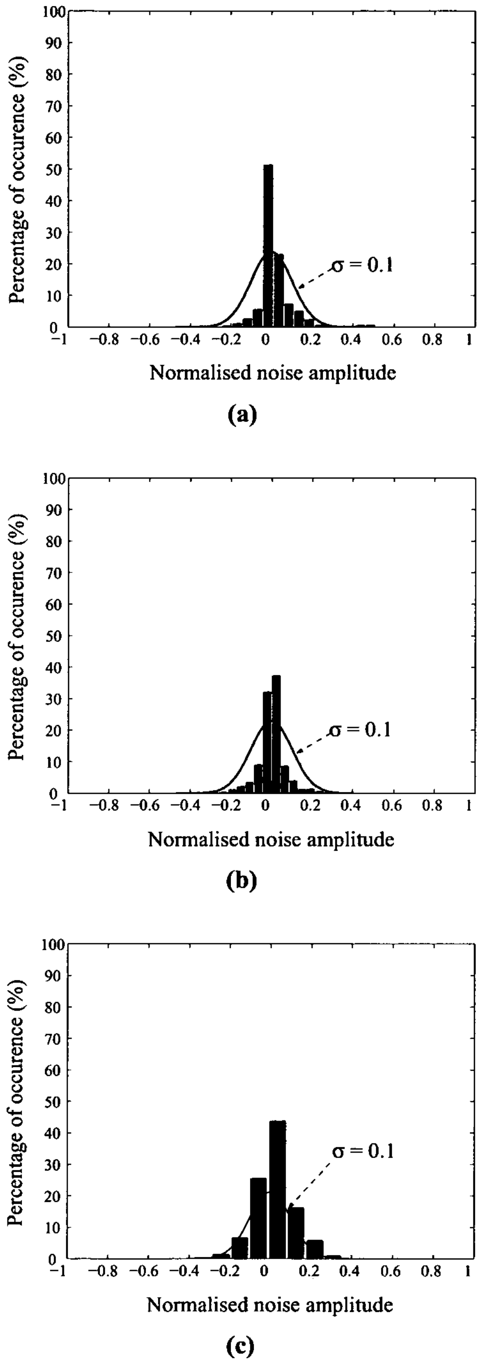


(a)

(b)

(c)

Figure 4.10: *4-trap noisy synaptic multiplier output noise (20000 data points) for (a)* $V_w = 0.3V$ *&* $V_{in} = 0.65V$ *(b)* $V_w = 0.6V$ *&* $V_{in} = 0.7$ *(c)* $V_w = 1.05V$ *&* $V_{in} = 0.8V$.



(a)

(b)

(c)

Figure 4.11: *Cumulative normalised noisy synaptic multiplier output noise amplitude implemented with (a) 4-trap (b) 10-trap (c) 100-trap.*

the multipliers retain the original performance, although the output becomes noisy. Small numbers of traps produce distinct levels of output noise amplitude while, as the number of traps become large, the levels become ‘continuous’, approximating a $1/f$ characteristic.

Chapter 5
Probabilistic Neural Computation

5.1 Introduction

Probabilistic (Stochastic) neural computation uses stochasticity to extract and classify important features in real-world data, in applications such as sensory fusion and classification [21–23, 31, 32].

The fundamental element of probabilistic neural computation is the stochastic neuron, which sums its inputs to decide the probability of the output state of the neurons. This probabilistic relationship gives a probabilistic neural system the ability to model the natural variability of real data. In addition, the probabilistic relationship enhances the system's fault tolerance [3, 24, 32, 66]; an important capability for the implementation of robust and reliable systems using future nanoscale MOSFETs.

The ability to both adapt to and tolerate noise makes probabilistic neural computation attractive for VLSI implementation. However, few probabilistic neural models are hardware-amenable, and even fewer are capable of modelling continuous-valued (analogue) data. The Diffusion Network for example has been shown to be able to model analogue data [67, 68], but a hardware implementation of this model is impractical because of its plethora of recurrent connections [3]. On the other hand, the PoE/RBM[1] hardware [3, 69] has limited ability to model continuous-valued data [3]. The Continuous Restricted Boltzmann Machine, with a simple and unsupervised training algorithm, has been shown to have the ability to model continuous-value data [21–24] and is amenable in VLSI [3, 23, 31, 32]. For the purpose of this study, therefore, the CRBM is chosen to serve as a well-developed and well-understood experimental platform for the investigation of probabilistic neural computation using nanoscale MOSFET noise. The results have, however, wider implications and the methodology developed is generic.

[1] POE/RBM is an abbreviation for *Product of Experts in the Restricted Boltzmann Machine form.*

5.2 Continuous Restricted Boltzmann Machine (CRBM)

The Continuous Restricted Boltzmann Machine (CRBM) is a probabilistic neural architecture capable of modelling analogue data and adapted ('trained') according to a simple, unsupervised training algorithm based on minimising contrastive divergence [3,23,24,32]. The CRBM is based on Hinton's Product of Experts [70], in Restricted Boltzmann Machine form (POE/RBM), comprising continuous stochastic neurons analogous to those of Diffusion Networks, with limited (Restricted) interconnectivity [24, 70]. The CRBM has been shown to be amenable to VLSI implementation and to be potentially useful as both a robust classifier and as a "novelty detector" [3,30,31]..

The probabilistic behaviour of the CRBM is introduced by the continuous stochastic neurons. The stochastic behaviour of the CRBM's neuron is driven by noise injection to its input. The noise inputs cause neurons to have continuous-valued, probabilistic outputs. Neurons with noise-induced stochasticity provide the ability to develop diverse stochastic behaviour (binary, continuous, deterministic) in the CRBM which leads to a modelling flexibility that is advantageous with real data. Experiments with both artificially-generated and real- biomedical [24] and chemical data [21,22] show that CRBM can model continuous data successfully with a simple, reliable training algorithm.

5.2.1 General architecture

The CRBM has one visible and one hidden layer with only interlayer connections. Fig.5.1 shows an example of CRBM with 3 visible neurons and 4 hidden neurons. The dark-grey circles (v_0 and h_0) represent two bias (permanently 'ON') neurons whose outputs are 1. The connection between bias neurons to a neuron is thus the threshold of the neuron. The interlayer connections (between visible and hidden neurons) are bidirectional and symmetric ($w_{ij} = w_{ji}$). The vector $w^{(i)}$ denotes the weight vector of hidden neuron i.

The restricted architecture enhances the distinctive functions of neurons in the visible and hidden layers [71]. Visible neurons pass and receive data to and from the 'outside world' (interface). Normally the number of visible neurons corresponds to the number of dimensions

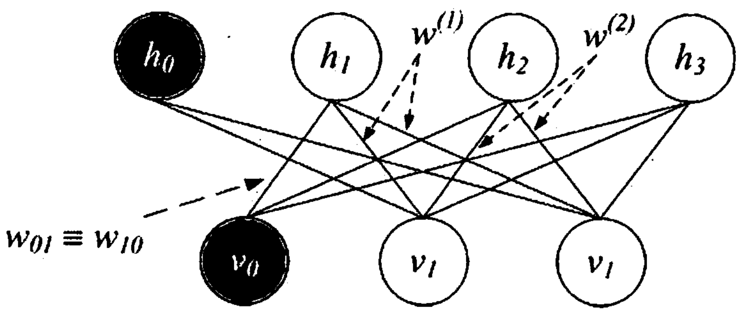


Figure 5.1: *CRBM network with 3 visible neurons and 4 hidden neurons.*

of data the CRBM must model. The function of hidden neurons, however, differs from that of visible neurons. Each hidden neuron represents an 'expert' whose weight vector encodes a particular feature of the input data. The number of hidden neurons therefore depends on the complexity of the features of the input data that the CRBM needs to model. A small number of hidden neurons may result in a poor modelling capability. However, increasing the number of hidden neurons (at the expense of bigger computation and network size) may not necessarily give a better capability [3, 24]. For each given case, the number of hidden neurons is chosen empirically to maximise modelling ability while minimising the number of free parameters [24].

5.2.2 A continuous stochastic neuron

The CRBM employs continuous valued stochastic neurons (Fig.5.2). Let s_i be the output of neuron i, with inputs from neurons with states $\{s_j\}$ connected by a weight matrix $\{w_{ij}\}$. The behaviour of neuron i is [32]:

$$s_i = \varphi_i \left(\sum_j w_{ij} s_j + n_i \right) \tag{5.1}$$

$$\text{with} \quad \varphi_i(x_i) = \theta_L + (\theta_H - \theta_L) \cdot \frac{1}{1 + \exp(-a_i x_i)} \tag{5.2}$$

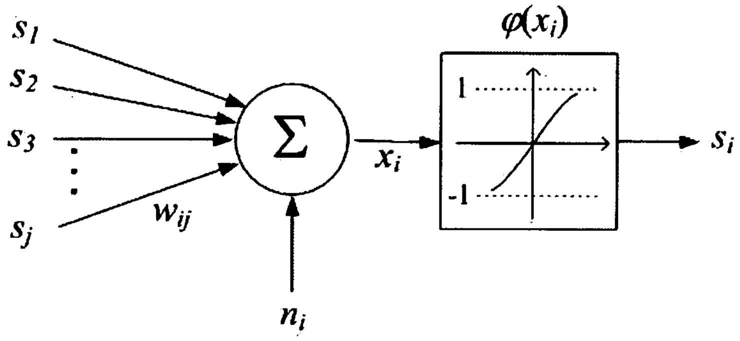


Figure 5.2: *Continuous valued CRBM neuron.*

where $\varphi(\cdot)$ is a sigmoid function with asymptotes at θ_H and θ_L, where a_i controls the slope of $\varphi(\cdot)$, and thus the nature of the neuron's stochastic behaviour [32]. If a_i is high, the neuron is essentially a binary-stochastic 'decision-make', while for a_i low, the neuron is more or less deterministic (i.e. comparable to a Multi-Layer Perceptron unit). Between those extremes, the neuron is able to model the noise and variability that is present in all real data. In the 'perfect CRBM' [3,23,24,32], $n_i = \sigma \cdot N_i(0,1)$ represents a noise input component according to the probability distribution

$$p(n_i) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-n_i^2}{2\sigma^2}\right). \tag{5.3}$$

σ represents a 'noise'-scaling constant and $N_i(0,1)$ represents a Gaussian variable ('noise') with zero mean and unit variance.

5.2.3 CRBM training

The CRBM is trained by adapting both the weight $\{w_{ij}\}$ and noise-control $\{a_i\}$ parameters by minimising "contrastive divergence" (MCD) between training data and the one-step Gibbs sampled data [72]. During training, the visible neurons are clamped with training data to produce $\{v_i\}$. Then, the hidden neurons states $\{h_j\}$ are sampled according to Eq.(5.1). One-step Gibbs sampled data are derived by repeating these procedure so that visible and hidden neurons are sampled once more to produce $\{\hat{v}_i\}$ and $\left\{\hat{h}_j\right\}$. Fig.5.3 illustrates the one-step

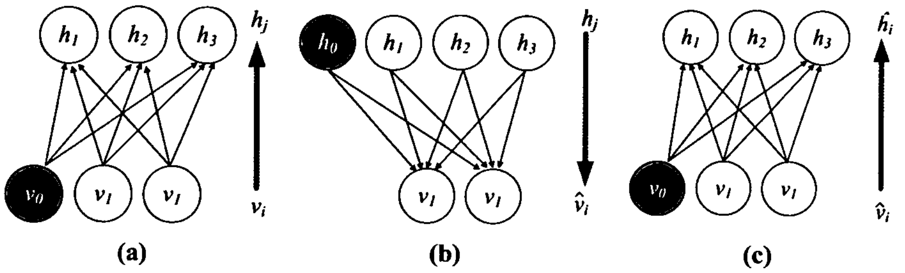


Figure 5.3: *One-step Gibb sampling.*

Gibbs sampling of CRBM with two visible and three hidden neurons. The weight $\{w_{ij}\}$ and noise-control parameters $\{a_i\}$ of CRBM are updated according to the following simplified[2] MCD training rules [3, 32]:

$$\Delta w_{ij} = \eta_w \left(\langle v_i h_j \rangle - \left\langle \hat{v}_i \hat{h}_j \right\rangle \right), \tag{5.4}$$

and

$$\Delta a_i = \frac{\eta_a}{a_i^2} \left(\langle s_i^2 \rangle - \langle \hat{s}_i^2 \rangle \right), \tag{5.5}$$

where v_i and h_j refer to the state of visible neuron i and hidden neuron j respectively, and s_i represents both v_i and h_j. η_w and η_a are constants defining the learning rates of w_{ij} and a_i respectively, and the brackets $\langle \cdot \rangle$ in Eq.(5.4) and Eq.(5.5) denote the expectation value over all training data.

The values of learning rate for visible η_v, hidden η_h and weight η_w are determined empirically depending of the complexity of the training data. Typical learning rate for visible η_v is set larger than ($\geq$10 times) that of hidden η_h and weight η_w. This setting encourages faster adaption of the $\{a_i\}$ for the visible layers, compared to the weight w_{ij}, and hidden noise control parameters $\{a_i\}$, to model the detail of the training data distribution [3, 21, 22].

[2] Training algorithms are simplified to enable a CRBM implementation with only multiplication and addition/subtraction [23, 30].

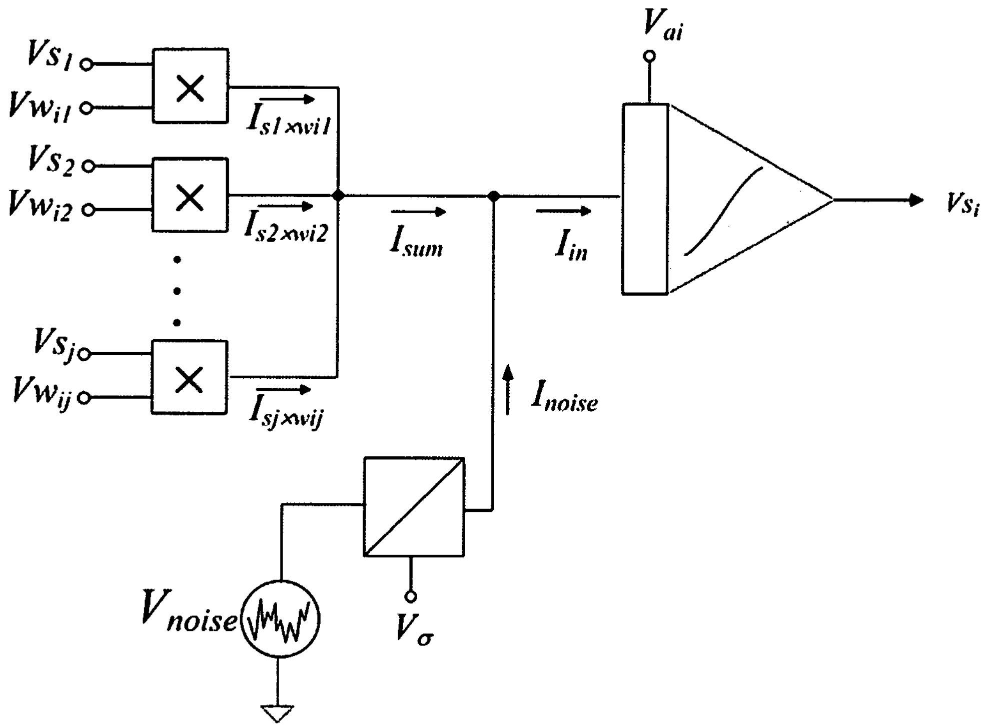


Figure 5.4: *Typical CRBM neuron circuit implementation.*

5.2.4 CRBM in VLSI

A full CRBM system implemented in VLSI, can reconstruct a variety of continuous data distributions [3]. The major improvement over previous probabilistic neural hardware implementations [66, 73, 74] is the use and training of the continuous-stochastic neuron. As this element of the CRBM also incorporates the noise inputs, it deserves some detailed description.

The CRBM's neuron circuit ([3]) is shown in Fig.5.4. The outputs of the four-quadrant analogue multipliers are summed into a current I_{sum} representing $\left(\sum_j w_{ij}s_j\right)$, which is subsequently threshold and transformed by the sigmoid-function block to produce output voltage Vs_i. The probabilistic behaviour of the CRBM's neuron is driven by injecting noise I_{noise} to I_{sum} to produce current I_{noisy} with stochastic behaviour controlled by the input voltage V_a, as depicted in Fig.5.4.

In the VLSI CRBM, noise is generated on-chip and injected directly into the CRBM neuron. However, [3] found that, unsurprisingly, noise generators can interfere with the analogue references, introducing extra computational errors. In addition, in large CRBM networks, the implementation of multiple, uncorrelated noise sources on- or off-chip becomes impractical, as unwanted correlations, reliability, area, and power problems increase. One solution is to localise fluctuations—introducing fluctuations only to the neurons but not to the deterministic training circuit. This points to the use of intrinsic MOSFET noise to replace externally generated noise.

5.3 Summary

CRBM has demonstrated a promising modelling ability and hardware amenability suitable for realising intelligent embedded systems. The modelling ability is attributed to the noise-induced, continuous-valued probabilistic behaviour of the CRBM neuron while hardware-amenability owes to the simple training algorithm employed by the CRBM.

The incorporation of artificially-generated noise is the distinctive feature of CRBM. Noise is added to the deterministic signal in the CRBM neuron to produce probabilistic output which is used to develop diverse stochastic behaviour in the CRBM.

While a full CRBM system implemented in VLSI can reconstruct a variety of continuous data distributions, the problems associated with on- or off-chip noise generation limit the robustness and flexibility of the CRBM VLSI system [3]. This leads to the idea of using intrinsic MOSFET noise to introduce noise localisation into each neuron, which would potentially alleviate the problems.

Chapter 6
Noise in the CRBM

6.1 Introduction

The 'Perfect CRBM' neurons use zero-mean Gaussian noise, injected from an external source to the pre-sigmoid sum of synaptic products, as shown in Fig.5.2. This artificially-generated noise (Eq.(5.3)) causes the neurons to have continuous-valued, probabilistic outputs. The injected noise variance (and hence, the effective maximum noise magnitude) is controlled by the global noise-scaling constant, σ. Small σ generates over-fitting and the CRBM system becomes near-deterministic, while large σ results to complete loss of the modelling capability—attributable to domination of the injected noise [32]. The optimal σ value depends upon the data distribution to be modelled. A useful rule-of-thumb, drawn from several CRBM-modelling projects, is to set σ for visible neurons close to the standard deviation of the data to be modelled, while σ for hidden neurons is set to intermediate values between 0.4 and 0.6 [21, 22, 24].

In VLSI implementation, the noise generator circuit must deliver uncorrelated noise, otherwise the correlation will be 'detected' by the training rule, and subsequently will introduce training errors [3, 66]. An efficient on-chip implementation of a multiple uncorrelated noise generator circuit has been reported in [3, 66], but this becomes unfeasible as the network size increases.

The externally generated Gaussian noise could, in principle, be replaced by intrinsic nanoscale MOSFET noise. Intrinsic nanoscale MOSFET noise is unlikely to be Gaussian with a mean of zero. It is therefore crucial to investigate and to understand how the CRBM's performance is affected by non-ideal noise characteristics. Additionally, the possible locations in the neuron architecture for the MOSFET noise source(s) have been investigated.

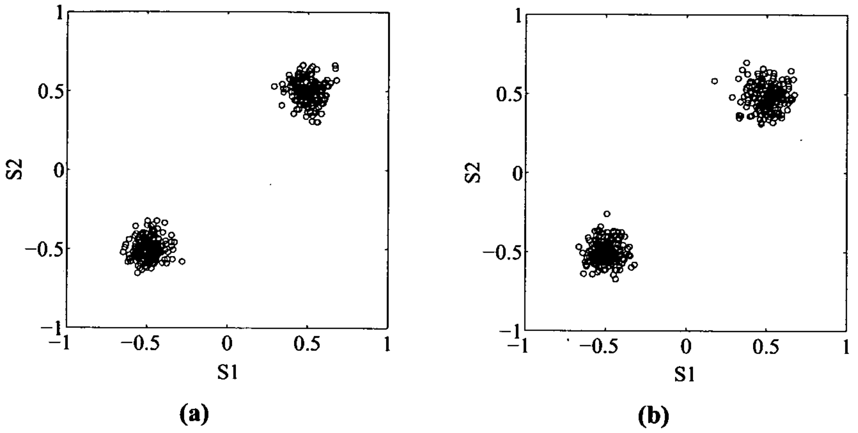


Figure 6.1: *(a) Training data. (b) 20-step reconstruction of the CRBM with zero mean Gaussian noise.*

6.2 Zero-mean Gaussian noise in CRBM

For a benchmark performance measure, a (Matlab) CRBM network (Fig.5.1) with zero-mean Gaussian noise is used to model two well-separated clusters of 200 data points, shown in Fig.6.1(a). Artificially-generated zero-mean Gaussian noise is injected to the pre-sigmoid input of each neuron. The CRBM is trained with $\eta_w = 0.3$, $\eta_{av} = 10$ for visible neurons, $\eta_{ah} = 1$ for hidden neurons, and $\sigma = 0.1$ for 5000 epochs. Fig.6.1(b) shows the reconstruction of the trained CRBM by Gibbs sampling from 200 random initial data for 20-steps[1]. The hidden neurons' weight vectors are projected into the visible neurons' state space to reveal the contribution of each neuron to the distribution of the reconstructed data. This is done by passing $\{w^{(i)}\}$ through the sigmoid function $\varphi(\cdot)$ in Eq.(5.1), i.e. $\mathrm{r}^{(i)} = \varphi\left(w^{(i)}\right)$ shown in Fig.6.2. The projected weight vectors are treated as continuous-valued outputs of visible neurons which reveals the effects of hidden neurons on the distribution of reconstructed data.

Fig.6.2 shows that the hidden neuron bias unit *h0*, hidden neuron *h1,* and hidden neuron *h2,* do not contribute to the reconstruction. On the other hand, the weight vector $\mathrm{r}^{(3)}$ suggests that hidden neuron *h3* encodes the training data clusters' (symmetric) positions.

[1] The reconstruction of the trained CRBM network by Gibbs sampling from any given number of random initial data for 20-steps will be referred to as 20-step reconstruction, unless clearly stated otherwise.

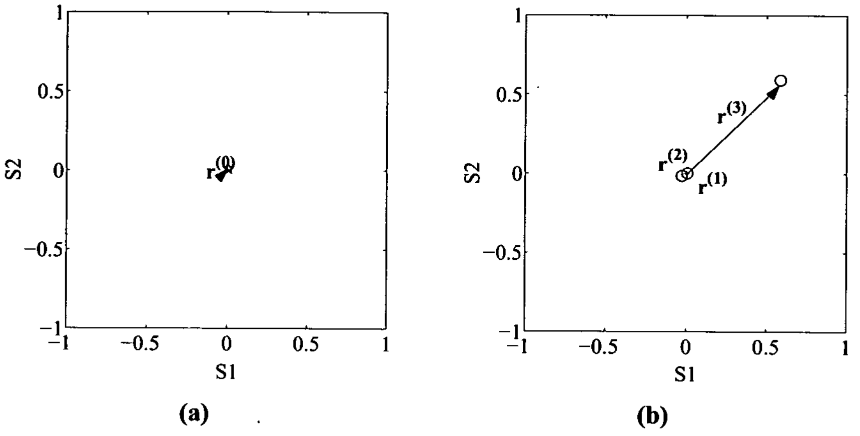


Figure 6.2: *(a) Weight vector for bias neuron (b) Weight vector for hidden neurons.*

In this model, the 'scatter' in the data is entirely attributable to the injected noise; not to the contribution of hidden neurons *h1* and *h2*.

Fig.6.3(a) and Fig.6.3(b) show the evolution of $\{a_i\}$ for visible neurons and hidden neurons during training, respectively. Fig.6.3(a) displays $\{a_i\}$ evolving in a form of "autonomous annealing" [24, 32]; gradually reducing the noise level in the visible neurons, thus making the neurons near-deterministic. Fig.6.3(b) depicts similar $\{a_i\}$ evolution behaviour for the hidden neuron *h1* and *h2* while hidden neuron *h3* clearly performs a different function. The large $\{a_i\}$ of hidden neuron *h3* suggests that the neuron is behaving near-binary, and allowing two clusters to be modelled. For this simple training data, hidden neurons *h0-h2* can be removed without affecting the reconstruction quality.

6.3 Non-zero mean Gaussian noise in CRBM

Intrinsic nanoscale MOSFET noise may not be Gaussian with a mean of zero [4, 39]. It is therefore crucial to investigate and to understand how the CRBM's performance is affected by this form of non-ideal noise characteristic.

Before injecting MOSFET noise, the CRBM's ability to 'cope' with non-Gaussian noise in a pure software model will be demonstrated. With non-zero mean Gaussian noise, Eq.(5.1)

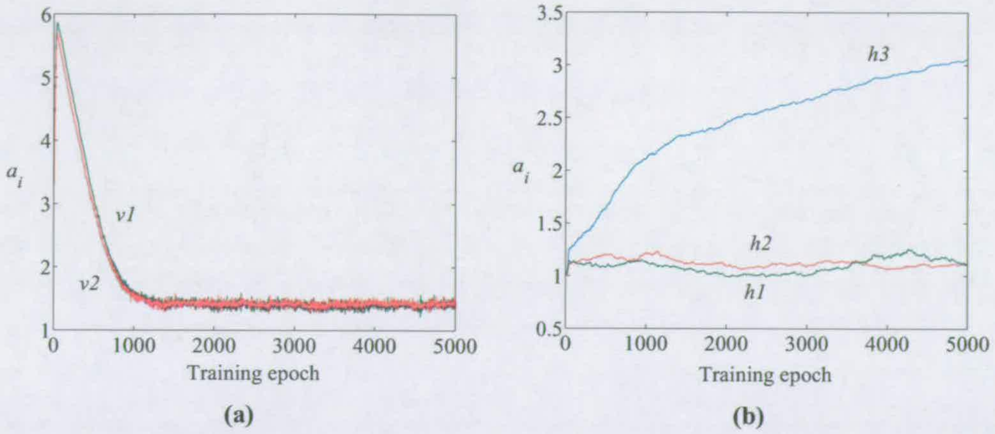


Figure 6.3: *(a) Visible neurons and (b) Hidden neurons noise control parameters evolution with training epoch.*

becomes:

$$s_i = \varphi_i \left(\sum_j w_{ij} s_j + n_{x(i)} \right), \tag{6.1}$$

where $n_{x(i)}$ represents a non-zero mean Gaussian noise source with distribution described as

$$p(n_{x(i)}) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-\left(n_i - \bar{n}_i\right)^2}{2\sigma^2} \right). \tag{6.2}$$

A CRBM (Fig.5.1) with non-zero mean noise was trained to model the data shown in Fig.6.1(a). The CRBM is trained with $\eta_w = 0.3$, $\eta_{av} = 10$ for visible neurons, $\eta_{ah} = 1$ for hidden neurons, $\sigma = 0.1$, and mean $\bar{n}_i = 0.8$. After 5000 training epochs, the CRBM reconstructed data points by Gibbs sampling from 200 random initial data for 20 steps, is shown in Fig.6.4.

The simulation is repeated with $\bar{n}_i = 2$. After 100000 epochs, the CRBM achieves a good 20-step reconstruction. These results indicate that the CRBM's modelling capability remains good, although longer training times are required as the mean of the noise increases.

To investigate how the CRBM responds to the injection of non-zero mean noise, the CRBM's 20-step reconstructions after 500, 30000, 40000 training epochs are shown in Figs.6.5(a-c),

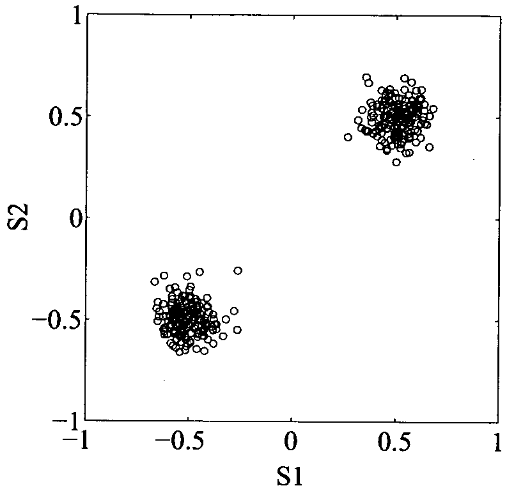
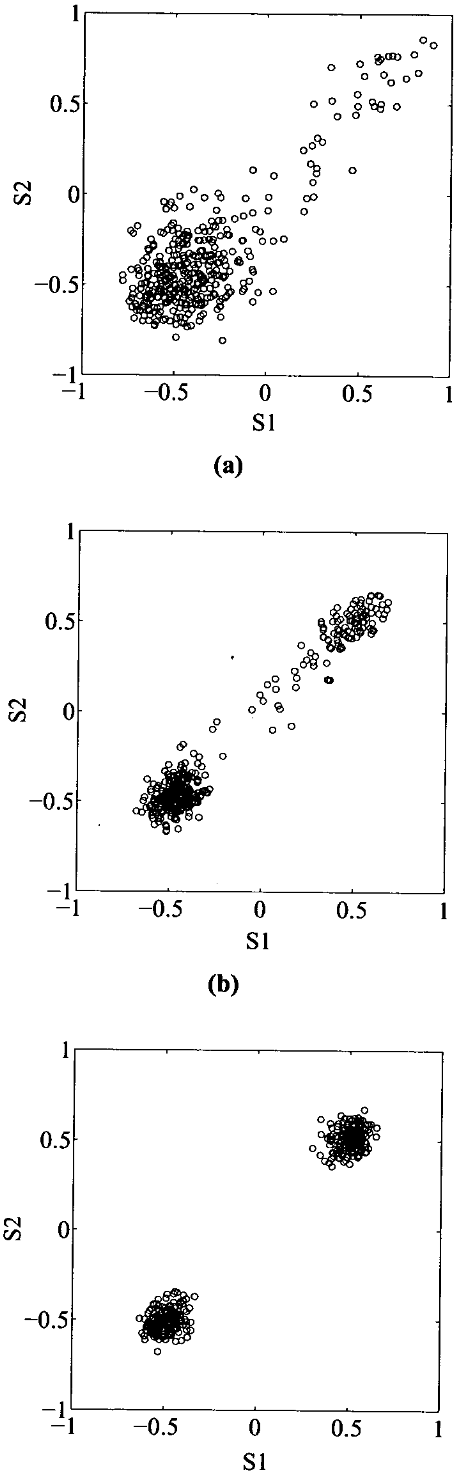


Figure 6.4: *20-step reconstruction of the CRBM injected with non-zero mean* $\bar{n}_i = 0.8$, *trained for 5000 epochs.*

respectively. The evolution during training of the $\{a_i\}$ of visible neurons and hidden neurons and the hidden-visible weights are shown in Fig.6.6(a-c). At the initial stage of training, the CRBM performs a crude approximation to the training data as shown in Fig6.5(a). Fig.6.6(a) shows large $\{a_i\}$ values for the visible neurons up to 2000 epochs, which inhibit the ability of the weights $\{w_{ij}\}$ to model the correct distribution [24]. As training continues, the $\{a_i\}$ values for the visible neurons are reduced, allowing weight adaptation (Fig.6.6(a)) to model the training data. Fig.6.5(b) shows that the CRBM has learnt the correct cluster separation but is still biased to the bottom-left corner. After 40000 epochs, the CRBM has modelled the training data well (Fig.6.5(c)). The weight evolution plot (Fig.6.6(c)) demonstrates that weights $\{w_{ij}\}$ are adapted to model the training data. The hidden to visible neuron bias weights (w_{10} and w_{20}) adapt to compensate for the large non-zero mean.

As training progresses, the hidden neuron *h1* becomes near-binary (Fig.6.6(b)), allowing the two clusters to be modelled clearly. The other hidden neurons encode the variance of the clusters in different directions. These results suggest that the CRBM's ultimate performance is not affected by the large mean. It simply takes longer to converge.

Finally, the CRBM with non-zero mean noise was used to model the non-symmetric training data shown in Fig.6.7(a). This training data is particularly interesting as it tests both the

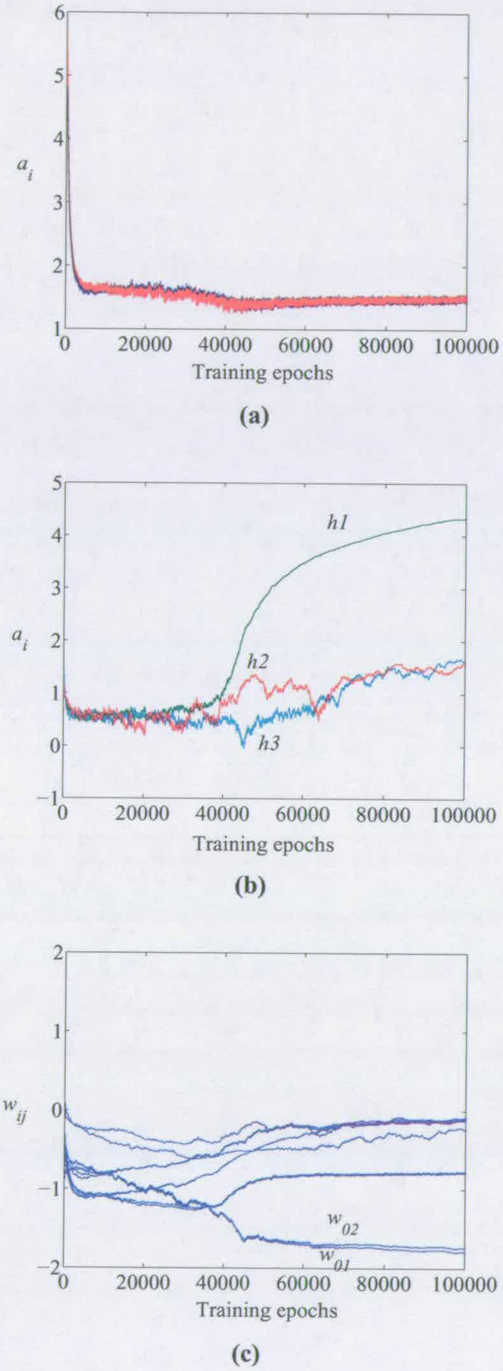


(a)

(b)

(c)

Figure 6.5: *20-step reconstruction by the CRBM injected with non-zero mean ($\bar{n}_i = 2$) Gaussian noise (a) after 500 training epochs (b) after 30000 epochs (c) after 40000 epochs.*



(a)

(b)

(c)

Figure 6.6: $\{a_i\}$ *for (a) visible, (b) hidden neurons, and (c) hidden-visible weight evolutions during training. w01 and w02 in (c) indicate the biased hidden neurons weight to visibles.*

CRBM's ability to regenerate a non-symmetric distribution and also to model the probability distribution of the training data correctly. For this experiment, to accelerate training, both η_{ah} and η_{av} were set to 7, $\eta_w = 0.3$ and $\bar{n}_i = 2$. After 10000 training epochs, the CRBM reconstructed data points by Gibbs sampled from 200 random initial data for 20 steps as shown in Fig.6.7(b). The results shown in Fig.6.7(b) indicate that the CRBM is able to adapt the non-zero mean noise to generate the correct data in position and probability. With 50 non-symmetric training data with similar distribution shown in Fig.6.7(a), the CRBM with the non-zero mean noise was still able to generate the correct data position and distribution.

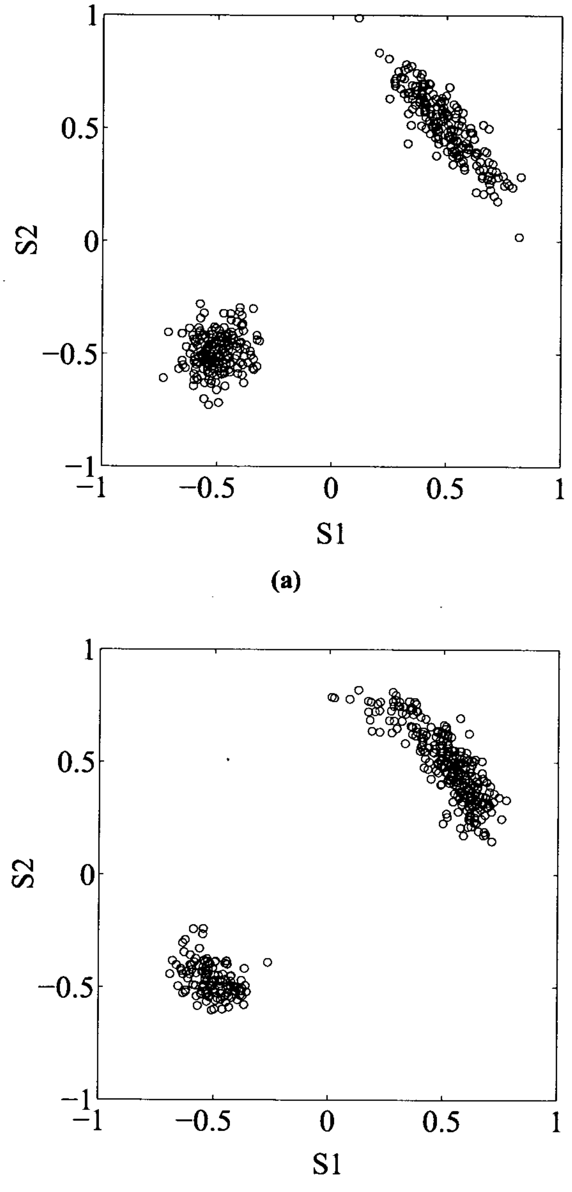
6.4 Non-Gaussian 'Pseudo-RTS' noise in the CRBM

The CRBM system has shown a good modelling ability by adapting its 'internal' noise (Gaussian distribution) to model an input data distribution.

Fig.6.8 shows the amplitude histogram of an artificial non-Gaussian noise source generated using Matlab to mimic RTS noise. The artificially-generated non-Gaussian noise is characterised by the separation x and distribution variance σ. This is a very crude representation of real nanoscale MOSFET noise, but it can be used to give a rough indication as to whether nanoscale MOSFET noise would work in a CRBM. Chapter 7 discusses a CRBM implementation with real nanoscale MOSFET noise.

The CRBM injected with a non-Gaussian ($x = 2$ and $\sigma = 0.1$) noise is trained to model two sets of data shown in Fig.6.1(a) and Fig.6.7(a) separately. Based on the non-zero mean Gaussian CRBM results, the separation x is expected to cause slow convergence. Therefore, in this experiment, the learning rates are set to $\eta_w = 0.3$, $\eta_{av} = 7$, and $\eta_{ah} = 7$. The CRBM is trained for 30000 epochs and Figs.6.9(a) and (b) depict the 20-step reconstruction results of the trained CRBM. In general, the CRBM has been able to model the data cluster separation and distribution correctly. However, the CRBM generates paired-clusters distribution shown in Figs.6.9(a) and (b). The occurrence of these peculiar reconstruction results is attributed to the non-Gaussian noise.

A simple experimentation found that by reducing x to zero, the additional clusters disappear.



(a)

(b)

Figure 6.7: *(a) Non-symmetric distribution training data. (b) 20-step reconstruction of the CRBM injected with non-zero mean ($\bar{n}_i = 2$) Gaussian noise after 10000 epochs.*

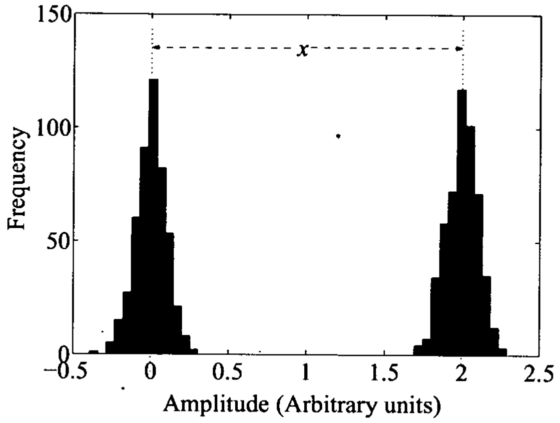


Figure 6.8: *Artificially generated non-Gaussian noise.*

The clusters become prominent as x increases. These analyses confirm the notion that the non-Gaussian noise distribution causes the extra clusters and that they are therefore unavoidable.

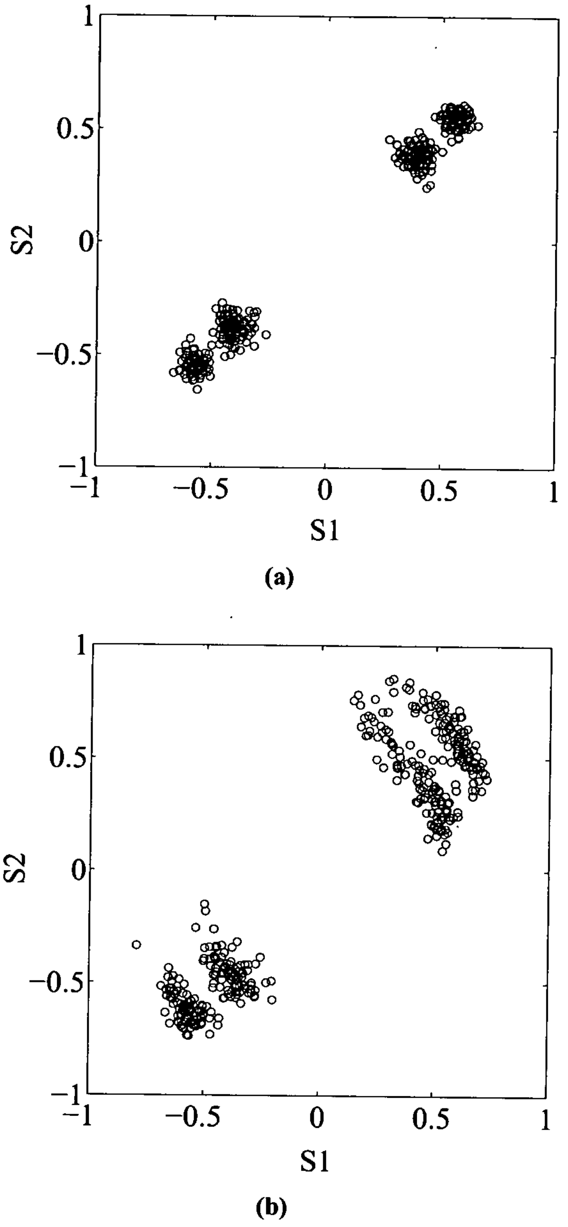
6.5 CRBM with noise in Multiplier

It was decided to localise noise in the synaptic multipliers as shown in Fig.5.2, as the multiplier is the most-repeated circuit in most architectures of this form and is thus the most likely candidate for the use of extremely small (nanoscale) devices. This analysis will be extended to other elements of this and alternative neural architectures in future work.

Eq.(5.1) is re-arranged to form Eq.(6.3), as addition and multiplication are both linear operations.

$$s_i = \varphi_i \left(\sum_j (w_{ij} s_j + n_i) \right) \tag{6.3}$$

To demonstrate that the CRBM is not affected by this change, artificially-generated zero-mean Gaussian noise is injected into the neurons of a CRBM network with two visible and three hidden units, as shown in Fig.6.10. Using the training data shown in Fig.6.1(a), the



(a)

(b)

Figure 6.9: *20-step reconstruction of CRBM injected with non-Gaussian noise after 10000 epochs for (a) symmetric distribution (b) non-symmetric distribution training data.*

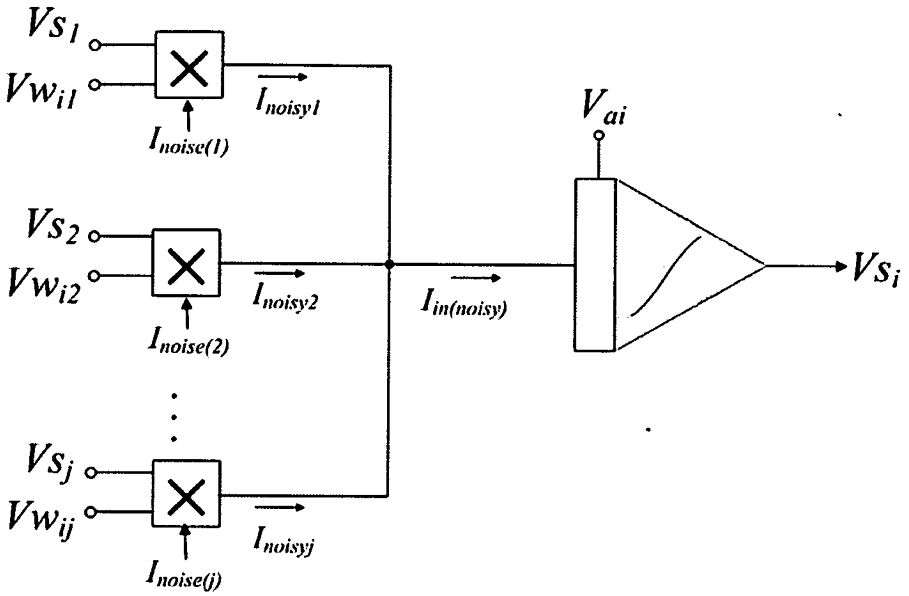
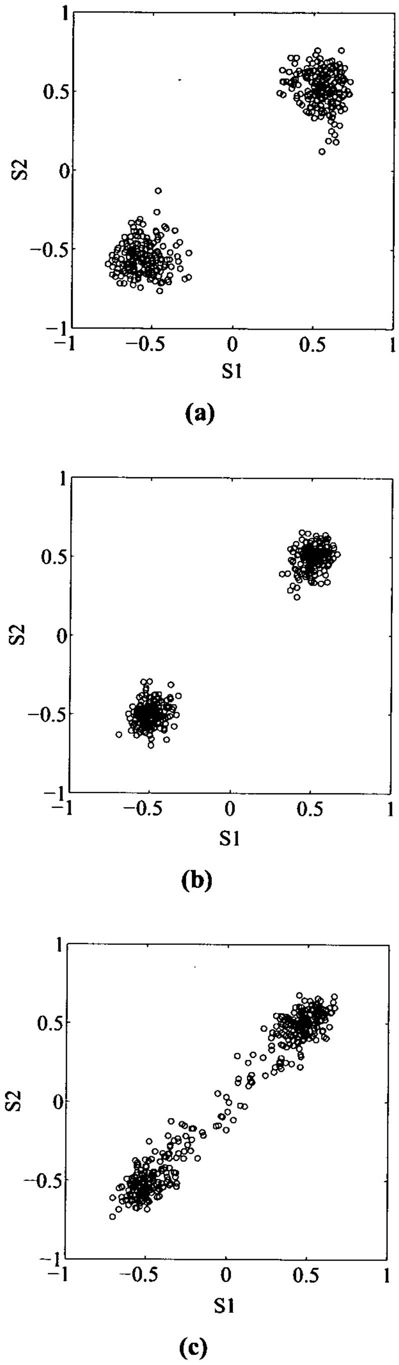


Figure 6.10: *CRBM neurons with localised noise in synaptic multipliers.*

CRBM is trained with $\eta_w = 0.3$, $\eta_{av} = 10$ for visible neurons, $\eta_{ah} = 1$ for hidden neurons, and $\sigma = 0.1$. Fig.6.11(a) shows 20-step reconstructions after 5000 training epochs. This is a rather dispersed distribution, indicating that overall noise levels are too high. The experiment is repeated by setting smaller σ, i.e 0.05 and 0.01, and the results are shown in Fig.6.11(b) and Fig.6.11(c), respectively. While $\sigma = 0.05$ gives the best reconstruction, the CRBM with $\sigma = 0.01$ shows a rough reconstruction of the training data. When subjected to further training (10000 epochs), the CRBM with $\sigma = 0.01$ produces similar performance to the CRBM with $\sigma = 0.05$. For a perfect CRBM, the noise with $\sigma = 0.01$ is unable to produce the correct reconstruction. These results indicate that injecting noise into the synaptic multiplier enables smaller noise magnitudes to be used, owing to the superposition effect as many synapses add their inputs to the receiving neuron's activity.

6.6 Summary

The ultimate goal of this study is to show that intrinsic low frequency nanoscale MOSFET noise can be used to implement a CRBM system. As the noise may not be Gaussian with



(a)

(b)

(c)

Figure 6.11: *20-step reconstruction with zero-mean Gaussian noise injected into every synaptic multiplier with noise variance set to (a) 0.1 (b) 0.05 (c) 0.01.*

mean of zero, it is important to investigate how the CRBM's performance is affected by the non-ideal noise characteristics. As the first step towards achieving the goal, the CRBM system performance has been evaluated with artificially generated non-Gaussian noise.

The results in Sec.6.4 show that the CRBM can display degraded, but usable, modelling performance when non-Gaussian noise sources with non-zero means are injected into the CRBM synaptic multipliers. Although the reconstruction results do not match the (essentially Gaussian) training data as faithfully as those with Gaussian noise, the CRBM is still able to capture the correct separation and distribution. These results suggest that nanoscale MOSFET noise has the potential to be used in CRBM implementation as long as the noise magnitude is relatively well matched to the distribution of the data.

The following chapter presents and discusses the implementation of a CRBM system with intrinsic low frequency nanoscale MOSFET noise.

Chapter 7
CRBM with Nanoscale MOSFET Noise

The experimental results discussed in Chapter 6 suggest that intrinsic MOSFET noise may indeed be able to replace externally-generated Gaussian noise. It is therefore the aim of this chapter to explore the potential of intrinsic nanoscale MOSFET noise to produce useful probabilistic behaviour of CRBM neurons.

The chapter first describes the methodology used to inject the noise data generated by the noisy 4-quadrant multiplier in Chapter 4 to the synaptic multiplication in the CRBM neurons. The CRBM system was then trained to model continuous data sampled from a non-symmetric distribution. The CRBM system's performance in regenerating the continuous data distribution, through its noise-induced probabilistic behaviour will be explored.

This chapter provides the necessary linkage between nanoscale device physics and probabilistic neural computation from which a preliminary conclusion can be drawn as to whether intrinsic nanoscale MOSFET noise can be used in probabilistic computation.

7.1 Methodology

The temporal current fluctuations associated with nanoscale MOSFET noise are incorporated in the CRBM through the implementation of the noisy synaptic multiplications shown in Fig.7.1. Full hardware implementation of a nanoscale-based CRBM system is not yet possible. Rather, the simulated temporal fluctuation of a noisy synaptic multiplier output, drawing on values stored in a look-up table (LUT) that represents both the multiplier's functionality and the noise associated with its DSM MOSFETs is incorporated. The synaptic multiplications are implemented using the noisy multipliers discussed in Chapter 4.

The general method for incorporating noise data into CRBM neurons is described by Fig.7.2. The product of $\{w_{ij}\}$ and $\{s_j\}$ is added with the corresponding noise datum extracted from

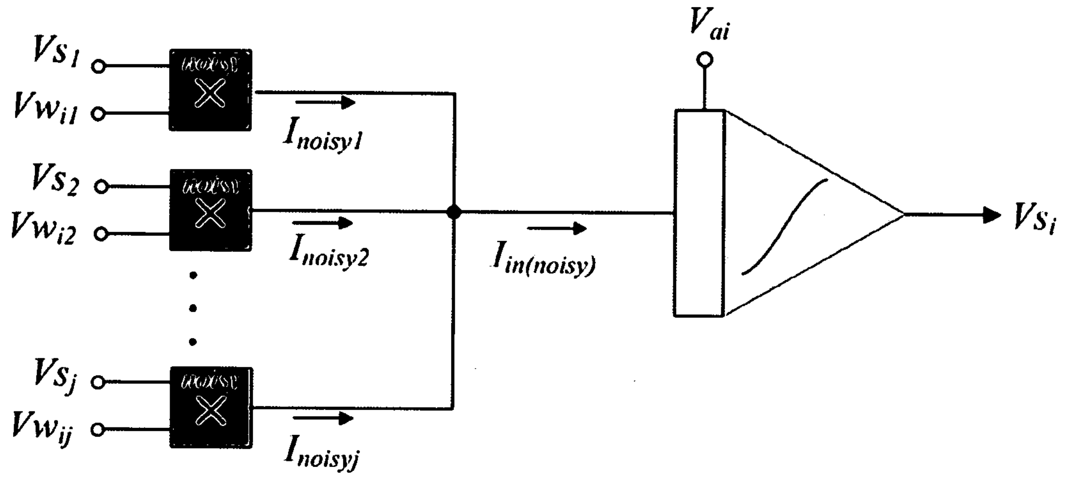


Figure 7.1: *CRBM neuron with noisy synaptic multiplication.*

the LUT that stores time domain noise data of a noisy synaptic multiplier output. Ideally, one LUT represents output noise data unique for a given multiplier. For a 3 visible and 4 hidden network, 3 multipliers are required to perform synaptic multiplication for each hidden neuron and 4 multipliers for each visible neuron. The total number of multipliers needed for the ideal case is 12, and thus 12 LUTs of time domain noise data are needed. It is not practical to implement this number of LUTs in Matlab, as this requires very large memory usage with a consequently unacceptably long simulation time. For the purpose of this study, only one LUT is used for noisy synaptic multiplication of CRBM neurons. In order to differentiate between multipliers, the start of the time domain noise data is shifted at random for each multiplier. It is acknowledged that by doing this, the noise data injected in each synaptic multiplication can be correlated. However, for the purpose of this study, this is acceptable, as the main investigation is on the effect of DSM noise form on CRBM performance. This limitation of the study is acknowledged, although it is not believed to cast any doubt on the thesis conclusions.

7.1.1 Noise data in look-up table

The noisy multiplier was designed based on 35*nm* atomistic-based CMOS technology that operates with a supply voltage of 1.5V (see Chapter 4). The multiplier inputs, i.e. the weight

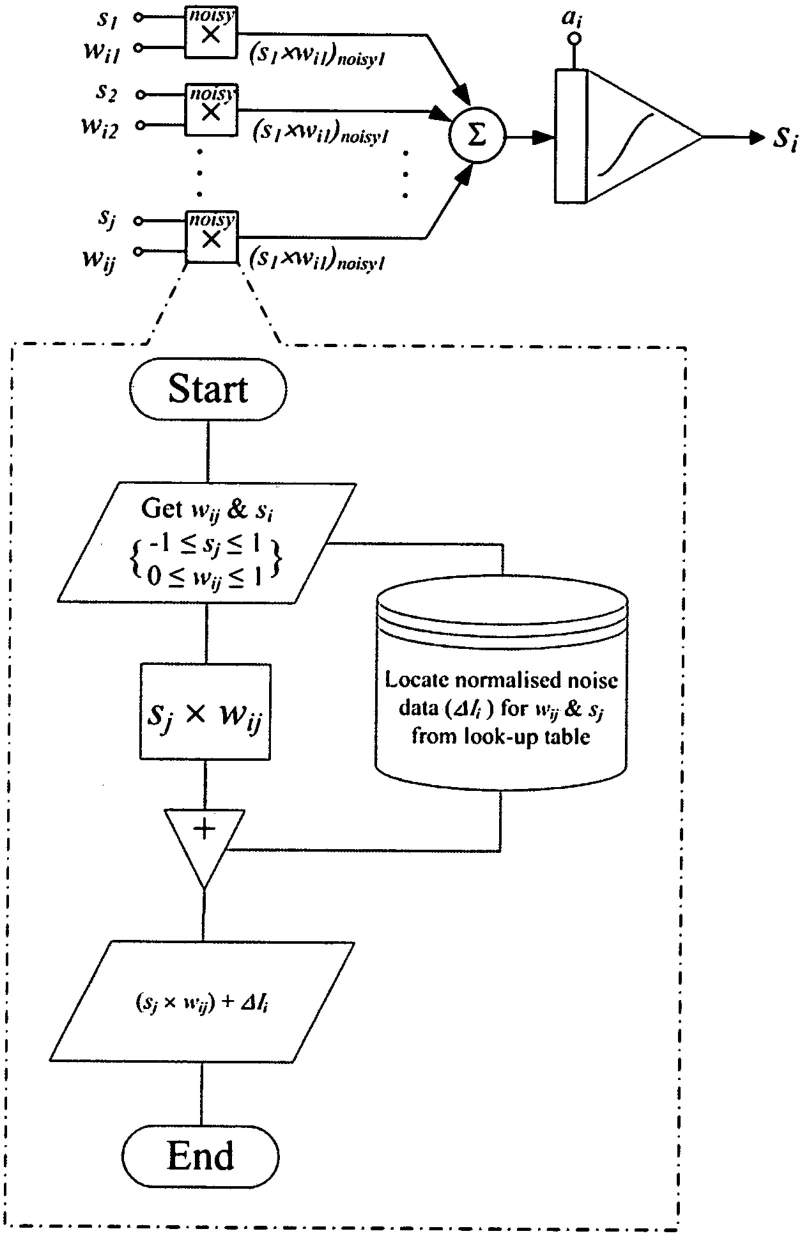

s_1
w_{i1}
noisy
×
$(s_1 \times w_{i1})_{noisy1}$
s_2
w_{i2}
noisy
×
$(s_1 \times w_{i1})_{noisy1}$
s_j
w_{ij}
noisy
×
$(s_1 \times w_{i1})_{noisy1}$
Σ
a_i
s_i
Start
Get w_{ij} & s_i
$\left\{ \begin{matrix} -1 \leq s_j \leq 1 \\ 0 \leq w_{ij} \leq 1 \end{matrix} \right\}$
Locate normalised noise data (ΔI_i) for w_{ij} & s_j from look-up table
$s_j \times w_{ij}$
+
$(s_j \times w_{ij}) + \Delta I_i$
End

Figure 7.2: *An illustration of how noisy synaptic multiplication is implemented in Matlab.*

	VLSI	Software
s_i	[0.55,0.95] (V)	[-1,1]
w_{ij}	[0.0,1.5] (V)	[-1.5,1.5]
noise amplitude	[-1,1] (μA)	[-1,1]

Table 7.1: *Mapping noisy multiplier input bias voltage (hardware) to Matlab (software).*

V_w and the state input data $V_{in}$are confined to [0,1.5] (V) and [0.55,0.95] (V) respectively. The noisy multiplier output current fluctuates in the time domain with noise characteristics inherited from the noisy MOSFETs used in key locations. For the purpose of this work, these time domain noise data are stored in a LUT, indexed according to V_w and V_{in}. Due to limited data storage and computation bottlenecks, only a limited number of $V_w \times V_{in}$ time domain noise data can be stored in the LUT. A compromise must be made between accuracy and computation time in representing output noise in Matlab. For this study, multiplier output noise data corresponding to V_w=[0:0.15:1.5] (V) and V_{in}=[0.55:0.05:0.95] (V) are stored in a LUT. For values that fall between the values stored, the closest-neighbour approximation is adopted.

This study incorporates synaptic multiplier output noise generated in SPECTRE into the Matlab implementation. The LUT method is adopted where the stored noise data are 'searched' and 'extracted' according to index parameters based on weight V_w and state V_{in} input voltages. In order to simplify the implementation in Matlab, the values for LUT are normalised. The weight V_w and state V_{in} input voltage are normalised to $\{w_{ji}\}$ and $\{s_i\}$ parameters in Matlab, respectively. The corresponding noisy multiplier output current noise which ranges between $-1\mu A$ and $1\mu A$ is normalised to value -1 to 1 in the LUT. Table 7.1 summarises the mapping of parameter values between the noisy multiplier hardware implementation and the software representation.

7.1.2 Data look-up

The flowchart in Fig.7.2 describes how noise data from a LUT is incorporated into synaptic multiplication $\{w_{ij} \times s_j\}$ to produce a probabilistic synaptic product. The flow is described as follow:-

1. Determine whether it is the start of a session, i.e. training (epoch = 1) or reconstruction (step=1):

   Condition 1: If epoch = 1 or step = 1 (start of a session), randomly generate the start point $t_{ij} = 1$ corresponding to the shaded row in the LUT shown in Fig.7.3, where index i and j identify a synaptic multiplier.

   Condition 2: If epoch $\neq$ 1 or step $\neq$ 1 (ongoing session), get the current epoch/step which corresponds to the row number in the LUT, i.e. $t_{ij} = n$ where *n* is the current epoch or step.

2. Get the current $\{w_{ij}\}$ and $\{s_j\}$ values.

3. Locate the corresponding noise datum for $\{w_{ij} \times s_j\}$ from the selected row (the 'black-shaded' location in the selected row in Fig.7.3)

4. Add the noise datum to $\{w_{ij} \times s_j\}$ to produce the noisy synaptic product.

5. Repeat 1 if it is not the end of simulation.

Condition 1 provides a degree of 'randomness' in the noise data to be linked to each multiplier which subsequently minimises correlation between noise data 'generated' for each synaptic multiplication. Another shortcoming is that the methodology assumes an instantaneous change in noise amplitude (due to instantaneous change in trap-occupancy) with changing bias conditions.

7.2 Modelling data with non-symmetric distribution

The modelling capability of a CRBM system with nanoscale MOSFET noise is explored with a network of 3 visible and 4 hidden neurons. The CRBM is trained to model two well-separated clusters of 200 data points with the non-symmetric distribution (one circular-Gaussian and one elliptic-Gaussian) shown in Fig.7.4(a). This training data is used as it tests the CRBM's capability to regenerate the non-symmetric cluster with the correct probability

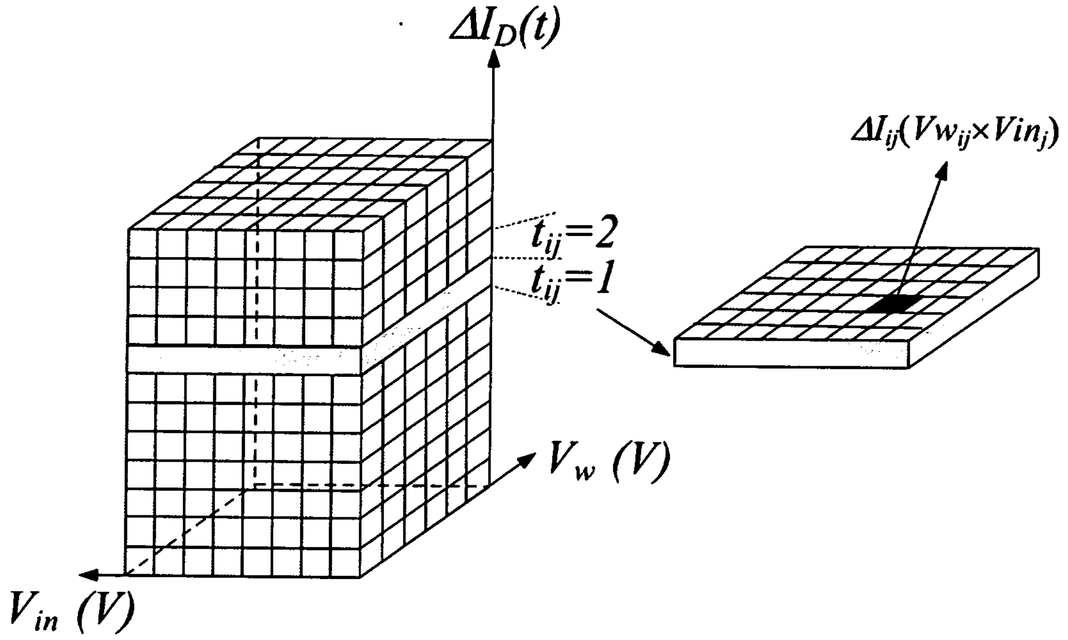


Figure 7.3: *Example of data extracted from the LUT. The value for V_w and V_{in} are mapped to w_{ij} and s_j respectively.*

distribution. However, the total number of training data is reduced to 50 to render simulation times tolerable[1]. The simulation results in Sec.6.3 show that reducing the training data to 50 would not affect the ability of the CRBM to generate the correct data position and distribution.

7.2.1 4-traps

The CRBM with 4-trap RTS noise data has been trained with $\eta_w = 0.3$, $\eta_{av} = 10$ for visible neurons, and $\eta_{ah} = 1$ for the hidden neurons, for 30000 epochs. After training, the CRBM generated the 20-step reconstruction shown in Fig.7.4(b), Gibbs sampling from 200 random initial data. Although this CRBM is unable to reconstruct an exact match to the training data, the general reconstruction characteristics are encouraging.

[1] On a Pentium 4 with 512MB cache PC workstation, it takes twenty hours to train CRBM with 200 training data point for 10000 epochs. On the other hand, 50 training data takes about 4 hours to complete 10000 epochs.

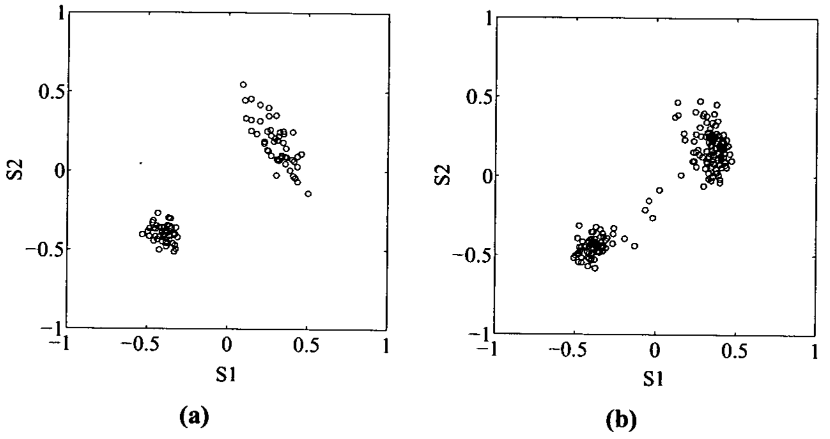
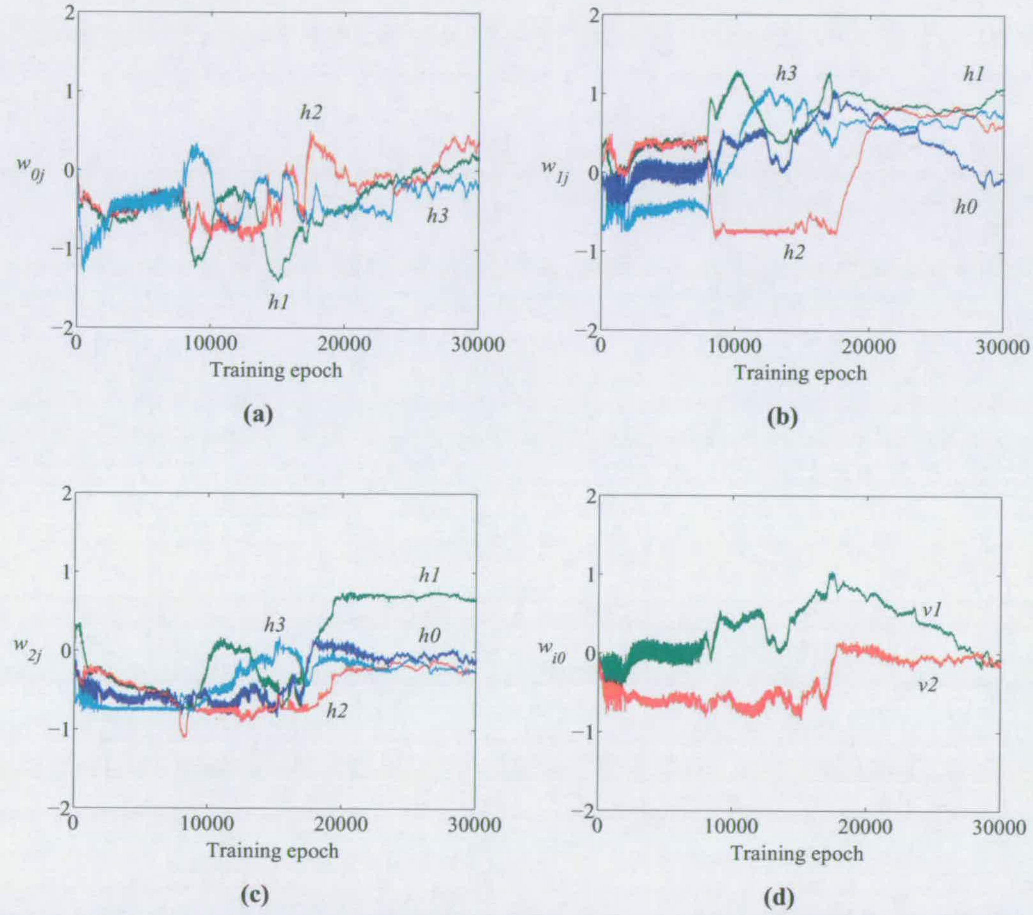


Figure 7.4: *(a) Non-symmetric distribution training data (b) 20-step reconstruction after 30000 epochs.*

Fig.7.5 shows the evolution of weight $\{w_{ji}\}$ during training. During the initial 20000 epochs, the weights fluctuate significantly, indicating that the system is performing a noise-mediated, crude approximation to the training data. In comparison, the Gaussian CRBM weight evolution (Fig.6.6(c)) follows a less noisy trend. Fig.7.5 shows that most $\{w_{ji}\}$ settle after 20000 training epochs, indicating that the training process reaches equilibrium after 20000 epochs. Fig.7.5 demonstrates the CRBM system's adaptability and attempt to minimise contrastive divergence, despite significant noise variation during training. Fig.7.6(a) shows the final (learnt) weight vectors of the hidden neurons projected onto the state space of the visible neurons after 30000 epochs, highlighting the contribution of each hidden neuron to the data reconstruction. The large noise control parameter a_i ($\rightarrow$ 3.5) of hidden neuron *h3* (Fig.7.6(b)) indicates the neuron's 'near-binary' behaviour. The 'near-binary' hidden neuron *h3* models cluster separation while other hidden neurons with smaller $\{a_i\}$ encode the variance (distribution) of the cluster. In other words, the hidden neuron *h3* decides which cluster each reconstruction point should belong to—the elliptic cluster or the circular cluster .

7.2.2 10-trap and 100-trap RTS in a CRBM

A CRBM has been implemented with more 'continuous' RTS noise based on 10-trap and 100-trap RTS models. The term 'continuous' refers to the less obviously quantised noise



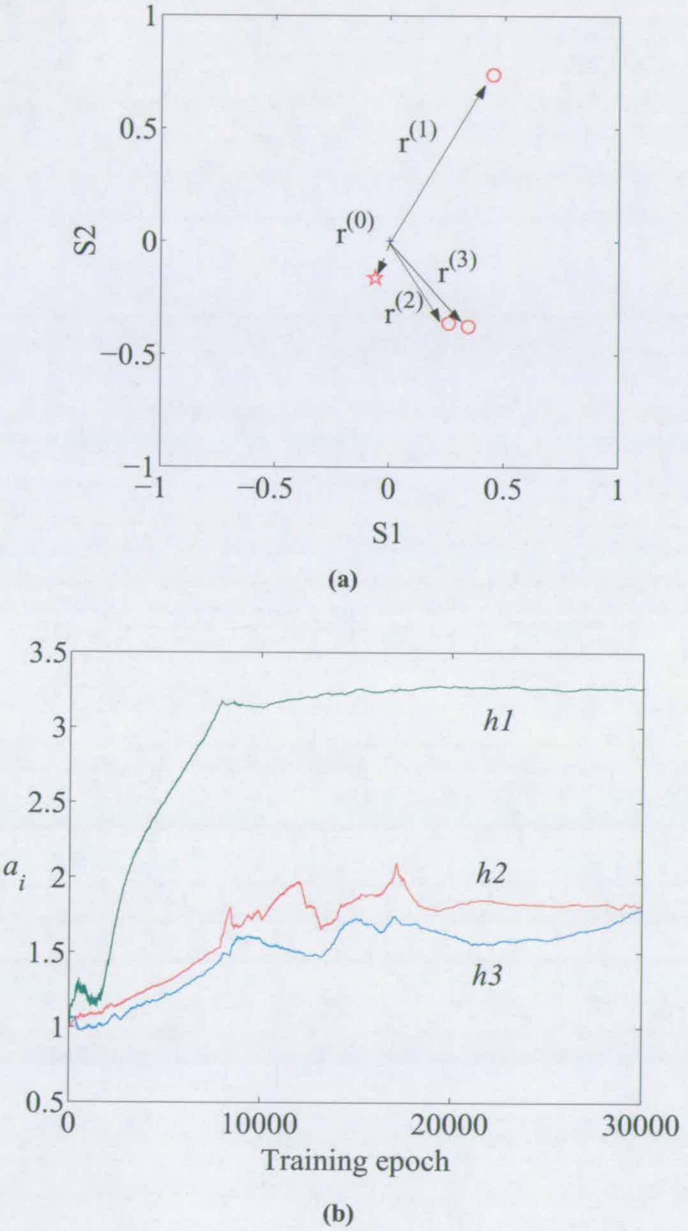
(a)

(b)

(c)

(d)

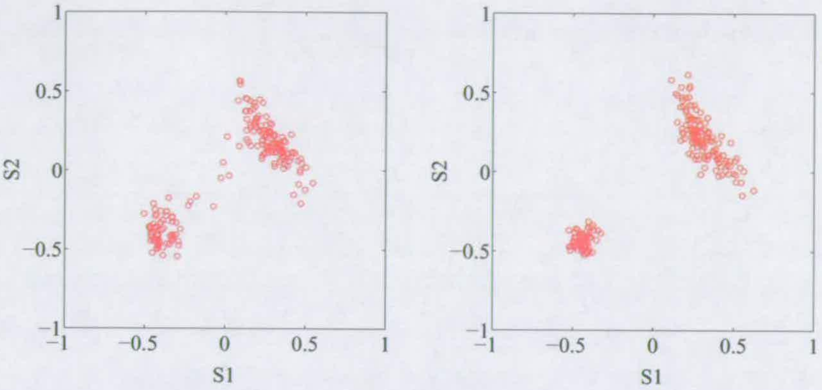
Figure 7.5: *CRBM with 4-trap implementation weights $\{w_{ji}\}$ evolution for 30000 epochs where visible neuron i and hidden neuron j and index 0 represents bias neurons during training.*



(a)

(b)

Figure 7.6: *(a) Learnt weight vector after 30000 epochs and (b) evolution of noise control parameters $\{a_i\}$ for hidden neurons, during training.*



(a)

(b)

Figure 7.7: *20-step reconstruction of CRBM implemented with (a) 10-trap and (b) 100-trap noisy synaptic multiplier output noise after 30000 epoch.*

amplitudes, in the presence of a larger number of traps [2].

In this experiment, 10-trap and 100-trap synaptic multiplier output noise data were used to implement a CRBM. The CRBM was trained with $\eta_w = 0.3$, $\eta_{av} = 10$ for visible neurons, $\eta_{ah} = 1$ for hidden neurons for 30000 epochs. After training, the CRBM generated the 20-step reconstruction shown in Fig.7.7(a)&(b). Both CRBM implementation (10-trap based and 100-trap based) have modelled the data well and reconstructed continuous-valued data points with results comparable to those of a Gaussian CRBM. These promising results indicate that intrinsic nanoscale MOSFET noise can, indeed, be used to implement a CRBM—only with a slower convergence time during learning.

7.3 Summary

The objective of this work was to explore whether relatively accurately-modelled intrinsic nanoscale MOSFET noise can, in principle, be used to underpin useful probabilistic behaviour in a stochastic computing architecture—in this study, the CRBM. This was done, as a first step, by localising fluctuation to the synaptic multipliers, which are the dominant element in many neural architectures. The main challenge of this study is linking the synaptic multiplier output noise generated in the SPECTRE circuit simulator to the CRBM model developed and simulated in Matlab. The linking is done through a Look-Up-Table that stores a synaptic multiplier's time domain output noise.

The results in this section demonstrate that the intrinsic nanoscale MOSFET noise can, indeed, be used to implement probabilistic computation in the CRBM, whilst retaining modelling ability comparable to that of a 'perfect' Gaussian CRBM. RTS noise takes the form of distinct discrete levels (i.e. 16 levels for 4-trap RTS implementation), but the CRBM Sec.6.4 would seem to suggest that discretised reconstructions would occur, owing to the noise level's quantised probability. This would be the case if noise was taken from fixed MOSFET bias conditions. However, in reality, bias conditions change as a natural feature of multiplier activity. Thus, dynamic bias conditions enable a CRBM to reconstruct a continuous data distribution from discretised noise. In a software CRBM, the noise level in visible neurons and hidden neurons can be tuned to optimum level to enhance learning [23, 24]. However, intrinsic nanoscale MOSFET noise cannot be tuned, and it is not, therefore, surprising that a CRBM with intrinsic nanoscale MOSFET noise performance is more difficult to train than a Gaussian software implementation.

As evident in [2], the large number of traps produces noise amplitudes closer to a Gaussian noise distribution, which is in turn closer to the noise in a 'perfect' CRBM. It is therefore expected that 10-trap and 100-trap based CRBM implementation will perform better than a 4-trap implementation, owing to the more 'continuous' distribution of noise amplitude. Experiments confirm this expectation.

Chapter 8
Summary and Conclusion

8.1 Summary

In future, RTS noise will become the dominant source of uncertainty in nanoscale MOSFETs. It will cause significant drain current fluctuation that limits the performance and functionality of a circuit. Several alternative architectural paradigms, such that the unreliable performance of these nanoscale MOSFETs could be tolerated (fault tolerance) or useful (adaptive), have been proposed [15–20]. This thesis investigates the use of DSM noise in an architecture that actually requires it. A 'demonstrator' architecture has been identified—the Continuous Restricted Boltzmann Machine (CRBM).

In the original, 'pure' CRBM, stochastic behaviour is introduced through injection of Gaussian noise from an external source [3, 23, 24, 32]. In hardware implementation, this requires additional circuitry and thus more silicon area [3]. In addition, the noise from external sources was found to introduce additional computational error to the system, making it less robust [3]. In this thesis, the noise source is localised in the synaptic multiplications of each stochastic neuron. An initial analysis with artificially generated 'pseudo-RTS' noise injected into a CRBM's synaptic multiplier shows that the CRBM retains a good modelling capability. This result encourages the potential of using nanoscale MOSFET noise in a CRBM system. The temporal current fluctuations associated with nanoscale MOSFET noise were incorporated in the CRBM through the implementation of noisy synaptic multiplications.

Unfortunately, statistically significant intrinsic MOSFET noise is not yet available. Therefore, a noisy MOSFET model has been developed to emulate the predicted noisy behaviour of future nanoscale MOSFET noise. Both $1/f$ and RTS noise have been studied, and a pragmatic approach has been taken to include the effect of this form of noise in MOSFETs, such that modelled noise can be included in circuit simulations. At this concept-proving stage, a computationally-simple model is more important than detailed accuracy. The noisy MOSFET model has been shown to produce the correct form of noise behaviour, valid in all operating regimes, and implementable in a circuit simulator.

A noisy analogue multiplier circuit has been implemented by replacing the key MOSFETs in the circuit with the noisy MOSFET models. This introduces noise to the signal path, producing a noisy output without compromising the underlying performance of the multiplier. Simulation results indicate that the multiplier retains the original performance, although the output become noisy.

The noisy multiplications are used to replace the normal synaptic multiplication process of the stochastic CRBM neurons. A systematic approach that embeds the noisy synpatic multiplication in the CRBM has been implemented, drawing on values stored in a look-up table (LUT) that represents both the multiplier's functionality and the noise associated with the nanoscale MOSFETs. This approach provides the necessary linkage between nanoscale device physics and probabilistic computation. Subsequently, the nanoscale-based CRBM system has demonstrated the ability to model data with simple, yet non-trivial distributions.

8.2 Conclusion

This thesis examined the suggestion that :-

"Low frequency drain current noise in future nanoscale MOSFETs can underpin useful probabilistic computation."

In this research, a novel methodology has been developed that builds a modelling bridge between atomistic models of nanoscale devices (and the physical devices upon which they were based), through look-up table models of the noise in such devices, to circuit models of the noisy nanoscale devices and circuits that can be simulated in reasonable time and using a conventional analogue simulator has been presented. Although we have chosen to study a probabilistic architecture, the methodology is perfectly generic and could be used to explore and investigate the temporal effects of nanoscale device noise on conventional computational paradigms, significantly in advance of the availability of integrated circuits (ICs) with true nanoscale devices. Understanding how future noisy nanoscale devices may effect circuit performance may pave the way to new design paradigms, some of which may exploit the noise for reliable system implementation—as explored in this thesis.

From the results presented, this research has demonstrated that intrinsic MOSFET noise can potentially be useful for probabilistic computation in future hardware implementation. Clearly, these results, taken from a single probabilistic architecture (the CRBM) with accurate, but limited models of nanoscale RTS noise, are insufficient for claiming that the proposed approach will solve all conventional computing problems. However, even with this simple caveat, these results indicate that a principled method can be developed to study the effects of nanoscale MOSFET noise in computational architectures and that early indications are that architectures can be developed that can 'make use of' such noise in their operation.

While the outcomes of this study are encouraging, there are several limitations which must be acknowledged.

1. Simplified nanoscale MOSFET noise models: the models were developed based on extensive assumptions listed in Sec.3.4.1 and fitting parameters (Table 3.1) adopted from large MOSFETs. Although it has been shown that the simplified models are able to produce the predicted noise characteristics in future nanoscale MOSFETs, detailed accuracy in the generated noise cannot be guaranteed.

2. Representation inaccuracy from using LUT: since the full hardware implement of CRBM with nanoscale MOSFETs is not yet possible, a LUT was used to link noise data to the CRBM software implementation. Due to limited storage space and computation resources, only limited data can be represented in the LUT. An approximation is done for any value not stored in the LUT and thus may introduce inaccuracies. In addition, since only one LUT was used to represent noise data from several noise sources (multipliers), it is possible that correlation may be introduced.

3. Only one probabilistic architecture was studied: the CRBM was used as a 'demonstrator' architecture and has shown encouraging results. Unfortunately, it is not possible to generalise the findings of this thesis to other architectures. In fact, the study on the CRBM system has not been done exhaustively with other variation of data. Therefore,

there is not enough evidence to comment about performance integrity of the CRBM with nanoscale MOSFET noise.

8.3 Future work

The work presented in this thesis is based on a simplified nanoscale MOSFET noise model. While the noise generated conformed to the predicted noise behaviour in future MOSFETs, real nanoscale MOSFET noise mechanisms will be more complex [4]. Once nanoscale MOSFET with significant noise become available, it will be important to repeat the work presented in this research with real noise data.

The further development of time domain noise analysis based on the nanoscale MOSFET model as a standard capability in a commercially available simulator is recommended for future nanoscale circuit analysis. This capability will allow better understanding of circuit performance in the presence of significant temporal fluctuation of device performance. Once this capability becomes available, it would be useful to implement a full hardware simulation of a CRBM system. This eliminates the approximation errors introduced when mapping noise data into the software implementation, giving a closer representation of a real nanoscale hardware implementation.

A promising modelling ability has been demonstrated, it is recommended to further explore the capability of nanoscale-based CRBM system to model more complex and realistic data, e.g. bio-medical data. This is important to prove that a nanoscale-based CRBM system can be exploited in all contexts and with all data for a real world applications.

In this research, the CRBM system is a 'demonstrator' architecture for studying the feasibility of using nanoscale MOSFET noise for probabilistic computation. The CRBM was chosen mainly due to its intrinsically-interesting probabilistic behaviour, which has been proven suitable for hardware implementation [3]. Furthermore, the CRBM system was developed locally, and thus expertises on this system was readily available. In order to generalise the findings of this research, it is encouraged to repeat the work with a different probabilistic architecture such as the Diffusion Network [67, 68].

Appendix A
MOSFET Channel Voltage Approximation

The mean capture $\bar{\tau}_c$ and emission $\bar{\tau}_e$ time are described by Eq.(2.3) and Eq.(2.4), respectively. The bias dependency $\bar{\tau}_c$ and $\bar{\tau}_e$ is reflected through the approximation of electron concentration n at the given trap location using Eq.(3.11). Using Eq.(3.11), effective electron concentration n at a specific location in a channel is determined by approximating the effective surface potential ψ_S and quasi-Fermi potential $V(y)$ at that location.

A.1 Approximating the Quasi-Fermi Potential, $V(y)$

The Quasi-Fermi potential $V(y)$ describes the effective voltage approximated for a given lateral position along the channel. It is very dependent on bias conditions.

In linear operation, the quasi-Fermi potential $V(y)$can be described by [58]:

$$V(y) = G_y - \sqrt{(G_y)^2 - 2\frac{y}{L}\left(G_y\right)V_{DS} + \frac{y}{L}V_{DS}^2}, \tag{A.1}$$

where G_y represents $\frac{V_G - V_{th}}{m}$, y is the lateral location measured from the source, L is the channel length, and m is the body effect coefficient defined by $\frac{\sqrt{2q\varepsilon_{si}N_{sub}}}{C_{ox}}$.

In order to describe quasi-Fermi potential $V(y)$ in saturation operation ($V_{DS} > V_{GS} - V_{th}$), the MOSFET channel has to be divided into two sections defined by L and S, as seen in Fig.A.1.

The section between source end and pinch-off point is denoted by the y^L axis. Along this axis, $V(y^L)$ varies linearly and can be described by slight modification (A.1) to be:

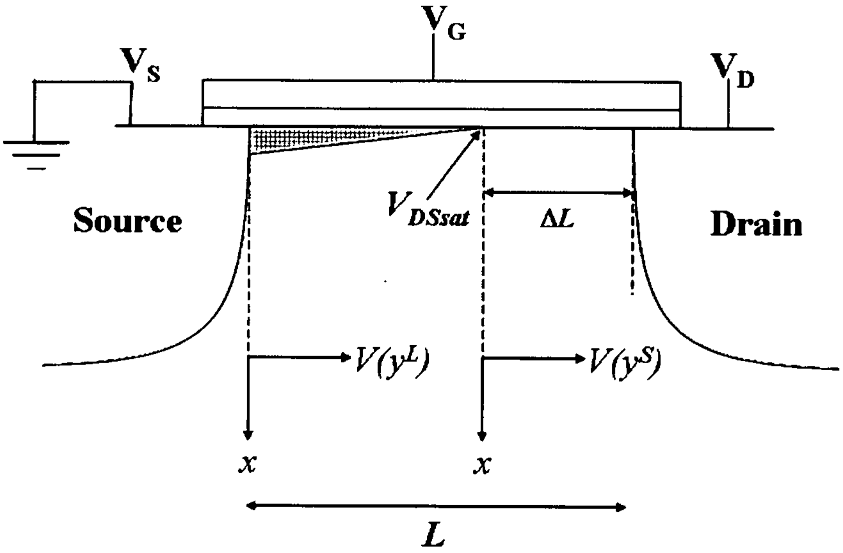


Figure A.1: *MOSFET cross-section for saturation operation with channel length modulation. The pinch-off point is defined when $V_{DS} = V_{GS} - V_{th}$ and denoted by V_{DSsat}.*

$$V(y) = G_y - \sqrt{(G_y)^2 - 2\frac{y \times G_y \times V_{DSsat}}{L - \Delta L} + \frac{y \times V_{DSsat}^2}{L - \Delta L}}, \tag{A.2}$$

where ΔL refers to the amount of channel length modulation by the drain voltage and defined by:

$$\Delta L = l \ln\left(\frac{V_{DS} - V_{DSsat}}{lE_{sat}} + \sqrt{\left(\frac{V_{DS} - V_{DSsat}}{lE_{sat}}\right)^2 + 1}\right), \tag{A.3}$$

where y is measured between 0 to $L - \Delta L$. l is a characteristic length defined by $\sqrt{\frac{\varepsilon_{si}}{\varepsilon_{ox}} T_{ox} x_j}$ and E_{sat} is the lateral field at saturation point, and is defined by $2\frac{v_{sat}}{\mu_{eff}}$. x_j, v_{sat}, and μ_{eff} are junction depth, velocity saturation , and effective mobility, respectively and their values can be extracted from the technology process file.

The section between pinch-off point and the drain, denoted by the y^S axis exhibits the effect of channel length modulation, where the pinch-off point moves towards the source end, exposing a region (ΔL) depleted of inversion charge. In this region, the carriers reach saturation velocity (v_{sat}). The quasi-Fermi potential $V(y^S)$ in this region is defined by:

$$V(y) = V_{DSsat} + lE_{sat}\sinh\left(\frac{y}{l}\right), \tag{A.4}$$

where y is measured between $L - \Delta L$ and L.

A.2 Approximating Surface Potential, ψ_S

In BSIM, MOSFET characteristics model are developed based on a threshold-voltage based model where surface potential ψ_S is approximated as a fixed quantity, described by [48]:

$$\psi_S = \frac{2kT}{q}\cdot\ln\left(\frac{N_{ch}}{n_i}\right), \tag{A.5}$$

where N_{ch} is channel doping concentration, a process dependent parameter. While this approximation is acceptable for 'long' devices—with continuous down scaling of MOSFET towards nano-scale dimension—a more accurate approximation of surface potential ψ_S has been suggested in [75] which explicitly includes the effect of bias conditions applied to the MOSFET. In [75], surface potential is approximated as:

$$\psi_S = f + u_t \times \ln\left(X_S^2 - \frac{f}{u_t} + 1\right), \tag{A.6}$$

where X_S is given by

$$X_S = \frac{V_{GB} - V_{FB} - f}{m \times \sqrt{u_t}} - \frac{(\psi_{Swi} - f)}{\left(m \times \sqrt{u_t}\right) \times \sqrt{1 + \left[\frac{\psi_{Swi} - f}{4u_t}\right]^2}}. \tag{A.7}$$

V_{FB} is the flat-band voltage, u_t is the thermal voltage defined by $u_t = \frac{kT}{q}$, ψ_F is the bulk Fermi potential defined by $u_t \times \ln\left(\frac{N_{sub}}{n_i}\right)$, and m is the body factor defined earlier. ψ_{Swi} is the surface potential in weak inversion defined by $\left(\sqrt{V_{GB} - V_{FB} + \frac{m^2}{4}} - \frac{m}{2}\right)^2$. The smooth transition of ψ_S from weak inversion to strong inversion is approximated by f, and is defined as:

$$\frac{2\psi_F + V(y)}{2} + \frac{\psi_{Swi}}{2} - \frac{1}{2} \times \sqrt{(\psi_{Swi} - 2\psi_F - V(y))^2 - 4\varepsilon^2}, \tag{A.8}$$

where $\varepsilon$is a smoothing factor conveniently fixed at 2×10^{-2}.

Appendix B
Publication list

This appendix lists the papers produced as the contributions of this PhD research:

N. H. Hamid, A. F. Murray, D. Laurenson, S. Roy, and B. Cheng, "Probabilistic computing with future deep submicrometer devices: a modelling approach," in *Proc. IEEE ISCAS '05*, Kobe, Japan, May 23-26, 2005, In Press.

N. H. Hamid, A. F. Murray, and S. Roy, "Noisy mosfets for time domain circuit implementation: A modelling approach," submitted to *IEEE Trans. on Circuit and System I.*

N. H. Hamid, A. F. Murray, and T. B. Tang, "Probabilistic neural computing with future nanoscale MOSFETs," submitted to *IEEE Trans. Neural Network.*

References

[1] R. F. Pierret, ed., *Field Effect Devices*, vol. IV of *Modular series on solid state devices 2nd Edition.* New York, USA: Addison-Wesley Publishing Company, 1990.

[2] N. H. Hamid, A. F. Murray, and S. Roy, “Noisy mosfets for time domain circuit implementation: A modelling approach,” *submitted to IEEE Trans. on Circuit and System 1.*

[3] H. Chen, *Continuous-value probabilistic neural computation in VLSI.* Phd thesis, University of Edinburgh.

[4] M. J. Kirton and M. J. Uren, “Noise in solid-state microstructures: a new perspective on individual defects, interface states and LF (1/f) noise?,” *Adv. Phys.*, vol. 38, p. 367, 1989.

[5] N. V. Amarasinghe and Z. C. Bulter, “Extraction of oxide trap properties using temperature dependence of random telegraph signals in submicron metal-oxide-semiconductor field effect transistors,” *J. Appl. Phys.*, vol. 89, pp. 5526–5532, 2001.

[6] “International technology roadmap for semiconductors, 2004 edition,” 2004.

[7] H. S. P. Wong, D. J. Frank, P. L. Solomon, C. H. J. Wann, and J. J. Welser, “Nanoscale cmos,” in *Proceedings of The IEEE*, pp. 537–570, , 1999.

[8] K. Likharev, “Electronics below 10nm,” in *Nano and Giga Challenges in Microelectronics* (J. G. et. al., ed.), (Amsterdam), pp. 27–68, Elsevier, 2003.

[9] M. H. Tsai and T. P. Ma, “The impact of device scaling on the current fluctuations in MOSFET’s,” *IEEE Trans. Elect. Devices*, vol. 41, pp. 2061–2068, Nov. 1994.

[10] M. Valenza, A. Hoffmann, D. Sodini, A. Laigle, F. Martinez, and D. Rigaud, “Overview of the impact of downscaling technology on 1/f nosie in p-mosfets to 90nm,” *IEE Proc. Circuits, Devices, and Systems*, vol. 151, pp. 102–110, April 2004.

[11] G. Ghibaudo and T. Boutchacha, “Electrical noise and RTS fluctuation in advanced CMOS devices,” *Microelectronic Reliability*, vol. 42, pp. 573–582, 2002.

[12] H. Wong, “Low-frequency noise study in electron devices: review and update,” *Microelectronic Reliability*, vol. 43, pp. 585–599, 2003.

[13] H. M. Bu, Y. Shi, X. L. Yuan, Y. D. Zheng, S. H. Gu, H. Majima, H. Ishicuro, and T. Hiramoto, “Impact of the device scaling on low frequency noise in n-MOSFETs,” *Appl. Phys. A*, vol. 71, pp. 133–136, 2000.

[14] C. Hu, G. P. Li, E. Worley, and J. White, “Consideration of low-frequency noise in MOSFET’s for analog performance,” *IEEE Elect. Devics Lett.*, vol. 17, p. 552, Dec. 1996.

[15] A. DeHon, “Array-based architecture for fet-based, nanoscale electronics,” *IEEE Trans. on Nanotechnology*, vol. 2, pp. 23–32, , 2003.

[16] S. C. Goldstein and M. Budiu, “Nanofabrics: Spatial computing using molecular electronics,” in *The 28th Annual Int. Symp. on Computer Architecture.*

[17] J. Han and P. Jonker, “A system architecture solution for unreliable nanoelectronics devices,” *IEEE Trans. on Nanotechnology*, vol. 1, pp. 201–208, Dec., 2002.

[18] S. Folling, O. Turel, and K. Likharv, “Single-electron latching switches as nanoscale synapses,” in *Proc. of Int. Joint Conf. on Neural Network*, pp. 216–221, , 2001.

[19] R. I. Bahar, J. Mundy, and J. Chen, “A probabilistic-based design methodology for nanoscale computation,” in *Int. Conf. on CAD*, (San Jose, CA), Nov., 2003.

[20] K. Nepal, R. I. Bahar, J. Mundy, W. R. Patterson, and A. Zaslavsky, “Designing logic circuit for probabilistic computation in the presence of noise,” in *Design Automation Conference*, , 2005.

[21] T. B. Tang, H. Chen, and A. F. Murray, “Adaptive stochastic classifier for noisy ph-isfet measurements,” in *Proceedings of the International Conference on Artificial Neural Networks (ICANN2003)*, pp. 638–645, 2003.

[22] T. B. Tang, H. Chen, and A. F. Murray, “Adaptive, integrated sensor processing to compensate for drift and uncertainty: a stochastic ’neural’ approach,” in *IEE Proc. on Nanobiotechnology*, vol. 151, 2004.

[23] H. Chen and A. F. Murray, “A continuous restricted boltzmann machine with a hardware amenable learning algorithm,” in *Proceedings of the International Conference on Artificial Neural Networks (ICANN2002)*, pp. 358–363, 2002.

[24] H. Chen and A. F. Murray, “A continuous restricted boltzmann machine with an implementable training algorithm,” in *IEE Proc. of Vision, Image, and Signal Processing*, vol. 150, 2003.

[25] S. Roy, B. Cheng, G. Roy, and A. Asenov, “A methodology for introducing ’atomistic’ parameter fluctuations into compact device models for circuit simulation,” *J. Comp. Elec.*, vol. 2, pp. 427–431, 2003.

[26] A. Asenov, A. R. Brown, J. H. Davies, and S. Saini, “Hierarchical approach to ’atomistic’ 3-d mosfet simulation,” *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 18, pp. 1558 – 1565, Nov., 1999.

[27] A. Asenov, R. Balasubramaniam, A. R. Brown, and J. H. Davies, "RTS amplitudes in decananometer MOSFETs:3-D simulation study," *IEEE Trans. Elect. Devices*, vol. 50, pp. 839–845, March 2003.

[28] N. V. Amarasinghe, Z. C. Butler, A. Zlotnicka, and F. Wang, "Model for Random Telegraph Signals in sub-micron MOSFETs," *Solid State Electronics*, vol. 47, pp. 1443–1449, 2003.

[29] A. P. van der Wel, E. A. M. Klumperink, L. K. J. Vandamme, and B. Nauta, "Modeling Random Telegrah Noise under switched bias conditions using Cyclostationary RTS Noise," *IEEE Trans. Elect. Devices*, vol. 50, pp. 1378–1384, May 2003.

[30] H. Chen, P. Fleury, and A. F. Murray, "Minimizing contrastive divergence in noisy, mixed-mode vlsi neurons," in *Advances in Neural Information Processing System (NIPS2003)*, vol. 17, (Vancouver, Canada), 2003.

[31] H. Chen, P. Fleury, T. B. Tang, and A. F. Murray, "Adaptive noisy neural computation in mixed-mode vlsi," in *Proceedings of the Seventh International Conference on Cognitive and Neural Systems*, p. 68, May, 2003.

[32] H. Chen, P. Fleury, and A. F. Murray, "Unsupervised probabilistic neural computation in mixed-mode vlsi," in *Smart Adaptive System on Silicon (M. Valle ed.)*, Springer, Oct., 2004.

[33] B. J. Cheng, S. Roy, G. Roy, and A. Asenov, "Integrating Atomistic, intrinsic parameter fluctuations into compact model circuit analysis," in *33rd Conference of European Solid-State Device Research (ESSDERC03*, pp. 437–440, , 2003.

[34] M. Schulz, "Electrical characterization of the sio2-si system," *Microelectronic Engineering*, vol. 40, pp. 113–130, 1998.

[35] M. J. Uren, D. J. Day, and M. J. Kirton, "1/f and random telegraph noise in silicon metal-oxide-semiconductor field-effect transistor," *Appl. Phys. Lett.*, vol. 47, pp. 1195–1197, Dec 1985.

[36] E. Simoen and C. Claeys, "Random Telegraph Signal: a local probe for single point defect studies in solid-state devices," *Material Science and Engineering*, vol. B91-92, pp. 136–143, 2002.

[37] Z. Shi, J. P. Miĕville, and M. Dutoit, "Random Telegraph Signals in Deep Submicron n-MOSFET's," *IEEE Trans. Elect. Devices*, vol. 41, pp. 1161–1168, July 1994.

[38] N. V. Amarasinghe and Z. C. Bulter, "Complex Random Telegraph Signals in 0.06 μm^2 MDD n-MOSFETs," *Solid State Electronics*, vol. 44, pp. 1013–1019, 2000.

[39] E. Simoen, B. Dierickx, C. L. Claeys, and G. J. Declerck, "Explaining the amplitude of RTS noise in submicrometer MOSFET's," *IEEE Trans Elect. Devices*, vol. 39, pp. 423–429, Feb. 1992.

[40] E. Simoen, B. Dierickx, and C. Claeys, "Hot-carrier degradation of the Random Telegraph Signal amplitude in submicrometer Si MOSTs," *Appl. Phys. A: Solids and Surfaces*, vol. 57, pp. 283–289, 1993.

[41] O. R. D. Buisson, G. Ghibaudo, and J. Brini, "Model for drain current RTS amplitude in small-area MOS transistors," *Solid-State Electronics*, vol. 35, pp. 1273–1276, 1992.

[42] E. Simoen and C. Claeys, "Substrate bias effect on the random telegraph signal parameters in submicrometer silicon p-metal-oxide-semiconductor transistors," *Appl. Phys.*, vol. 77, pp. 910–914, Jan 1995.

[43] N. V. Amarasinghe, Z. C. Butler, and P. Vasina, "Characterization of oxide traps in 0.15 μm^2 MOSFETs using Random Telegrah Signal," *Microelectronics Reliability*, vol. 40, pp. 1875–1881, 2000.

[44] J. H. Scofield, N. Borland, and D. M. Fleetwood, "Reconciliation of different gate-voltage dependencies of *1/f*noise in n-MOS and p-MOS transistors," *IEEE Trans. Elect. Devices*, vol. 41, pp. 1946–1952, Nov 1994.

[45] K. K. Hung, P. K. Ko, C. Hu, and Y. C. Cheng, "A unified model for the flicker noise in metal-oxide-semiconductor field effect transistors," *IEEE Trans Elect. Devices*, vol. 37, p. 654, 1990.

[46] J. Chang, A. A. Abidi, and C. R. Viswanathan, "Flicker noise in CMOS transistor form subthreshold to strong inversion at various temperatures," *IEEE Trans. Elect. Devices*, vol. 41, pp. 1965–1971, Nov 1994.

[47] Y. Nemirovsky, I. Brouk, and C. G. Jakobson, "1/f noise in CMOS transistor for analog applications," *IEEE Trans. Elect. Devices*, vol. 48, pp. 921–927, 2001.

[48] "Bsim3v3.2.2 MOSFET Model: User's Manual," 2004.

[49] P. E. Allen and D. R. Holberg, eds., *CMOS Analog Circuit Design*. 2nd Edition, Oxford, UK: Oxford University Press, 2002.

[50] D. Xie, M. Cheng, and L. Forbes, "SPICE models for flicker noise in n-MOSFETs from subthreshold to strong inversion," *IEEE Trans. Computer-Aided Design of Integ. Circuit and Syst.*, vol. 19, pp. 1293–1303, Nov 2000.

[51] D. E. Hocevar, P. Yang, T. N. Trick, and B. D. Epler, "Transient sensitivity computation for MOSFET circuits," *IEEE Trans. on Computer-Aided Design*, vol. CAD-4, pp. 609–620, Oct. 1985.

[52] P. Bolcato and R. Poujois, "A new approach for noise simulation in transient analysis," in *Proc. IEEE Int. Symp. Circuits Syst.*, pp. 887–890, May, 1992.

[53] Mentor Graphic, *ELDO (Demo Version) User's Manual*, 2003.

[54] M. J. Uren, M. J. Kirton, and S. Collins, "Anomalous telegraph noise in small-are silicon metal-oxide-semiconductor field-effect transistors," *Physical Review B*, vol. 37, pp. 37–41, May, 1988.

[55] K. K. Hung, P. K. Ko, C. Hu, and Y. C. Cheng, "Random Telegraph Noise of Deep-Submicrometer MOSFET's," *IEEE Trans. Elect. Devices*, vol. 11, pp. 90–92, Feb. 1990.

[56] S. T. Martin, G. P. Li, E. Worley, and J. White, "The gate bias and geometry dependence of random telegraph signal amplitudes," *IEEE Elect. Devics Lett.*, vol. 18, pp. 444–446, Sep 1997.

[57] A. Lee, A. R. Brown, A. Asenov, and S. Roy, "Random telegraph signal noise simulation of decanano mosfets subject to atomic scale structure variation," *Superlattices and Microstructure*, vol. 34, pp. 293–300, 2003.

[58] Y. Taur and T. H. Ning, eds., *Fundametals of Modern VLSI Devices*. Cambridge, UK: Cambridge University Press, 1998.

[59] K. R. Laker and W. M. C. Sansen, eds., *Design of analog integrated circuits and systems*. International Edition, Singapore: McGraw-Hill Inc., 1994.

[60] C. Jacoboni and L. Reggiani, "The Monte Carlo method for the solution of charge transport in semiconductors with applications to covalent materials," *Review of Modern Physics*, vol. 55, pp. 645–705, July 1983.

[61] H. Chible, "Analog circuit for synapse neural networks VLSI implementation," in *The 7th IEEE Int. Conf. on Electronics, Circuits and Systems (ICECS 2000)*, vol. 2, pp. 1004–1007, May, 2000.

[62] G. Colli and F. Montecchi, "Low voltage low power CMOS four-quadrant analog multiplier," *IEEE Int. Symp. on Circuits and Systems*, vol. 1, pp. 496–499, 1996.

[63] D. Coue and G. Wilson, "A four-quadrant subthreshold mode multiplier for analog neural-network application," *IEEE Trans. on Neural Network*, vol. 7, pp. 1212–1219, 1996.

[64] N. Saxena and J. Clark, "A four-quadrant cmos analog multiplier for analog neural-network," *IEEE Journal of Solid-State Circuits*, vol. 29, pp. 746–749, 1994.

[65] A. Ambrozy, ed., *Electronic Noise*. New York: McGraw-Hill Inc., 1982.

[66] A. Astaras, *Pulse-stream binary stochastic hardware for neural computation: The Helmholtz machine*. Phd thesis, University of Edinburgh.

[67] J. R. Movellan and J. L. McClelland, "Learning continuous probability distributions with symmetric diffusion network," *Cognitive Science*, vol. 17, pp. 463–496, 1993.

[68] J. R. Movellan, "A learning theorem for networks at detailed stochastic equilibrium," *Neural Computation*, vol. 10, pp. 1157–1178, 1998.

[69] P. Fleury, H. Chen, and A. Murray, "On-chip contrastive divergence learning in analogue vlsi," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN'04)*, (Budapest, Hungary), 25-29, 2004.

[70] G. E. Hinton, "Product of experts," in *Proceedings of the Ninth International Conference on Artificial Neural Networks (ICANN'99)*, (Edinburgh, Scotland), 1-6, 1999.

[71] P. Smolensky, "Information processing in dynamical systems: Foundation of harmany theory," in *Parallel Distributed Processing: Exploration in the Microstructure of Cognition*, vol. 1.

[72] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Computation*, vol. 14, pp. 1771–1800, 2002.

[73] J. Alspector, B. Gupta, and R. B. Allen, "Performance of a stochastic learning microchip," in *Advances in Neural Information Processing Systems (NIPS88)*, vol. 1.

[74] A. Jayakumar and J. Alspector, "A cascadable neural network chip set with on-chip learning using noise and gain annealing," in *Proceedings of IEEE Custom Integrated Circuits Conference*, pp. 19.5.1–19.5.4.

[75] R. van Langevelde and F. M. Klaassen, "An explicit surface-potential-based MOSFET model for circuit simulation," *Solid-State Elect.*, vol. 44, pp. 409–418, 2000.