UNIVERSITY OF LIVERPOOL

SYNAPTIC WEIGHT MODIFICATION AND STORAGE IN HARDWARE NEURAL NETWORKS

Thesis submitted in accordance with the requirements of the University of Liverpool for the degree of Doctor in Philosophy by

Andrew Smith

November 2012

Abstract

In 2011 the International Technology Roadmap for Semiconductors, ITRS 2011, outlined how the semiconductor industry should proceed to pursue Moore's Law past the 18nm generation. It envisioned a concept of 'More than Moore', in which existing semiconductor technologies can be exploited to enable the fabrication of diverse systems and in particular systems which integrate non-digital and biologically based functionality. A rapid expansion and growing interest in the fields of microbiology, electrophysiology, and computational neuroscience occurred. This activity has provided significant understanding and insight into the function and structure of the human brain leading to the creation of systems which mimic the operation of the biological nervous system. As the systems expand a need for small area, low power devices which replicate the important biological features of neural networks has been established to implement large scale networks.

In this thesis work is presented which focuses on the modification and storage of synaptic weights in hardware neural networks. Test devices were incorporated on 3 chip runs; each chip was fabricated in a 0.35μm process from Austria MicroSystems (AMS) and used for parameter extraction, in accordance with the theoretical analysis presented.

A compact circuit is presented which can implement STDP, and has advantages over current implementations in that the critical timing window for synaptic modification is implemented within the circuit. The duration of the critical timing window is set by the subthreshold current controlled by the voltage, V_{leak}, applied to transistor M_{leak} in the circuit. A physical model to predict the time window for plasticity to occur is formulated and the effects of process variations on the window is analysed. The STDP circuit is implemented using two dedicated circuit blocks, one for potentiation and one for depression where each block consists of 4 transistors and a polysilicon capacitor, and an area of 980μm^2. SpectreS simulations of the back-annotated layout of the circuit and experimental results indicate that STDP with biologically plausible critical timing windows over the range 10μs to 100ms can be implemented. Theoretical analysis using parameters extracted from MOS test devices is used to describe the operation of each device and circuit presented. Simulation results and results obtained from fabricated devices confirm the validity of these designs and approaches. Both the WP and WD circuits have a power consumption of approximately

2.4mW, during a weight update. If no weight update occurs the resting currents within the device are in the nA range, thus each circuit has a power consumption of approximately 1µW.

A floating gate, FG, device fabricated using a standard CMOS process is presented. This device is to be integrated with both the WP and WD STDP circuits. The FG device is designed to store negative charge on a FG to represent the synaptic weight of the associated synapse. Charge is added or removed from the FG via Fowler-Nordheim tunnelling. This thesis outlines the design criteria and theoretical operation of this device. A model of the charge storage characteristics is presented and verified using HFCV and PCV experimental results.

Limited precision weights, LPW, and its potential use in hardware neural networks is also considered. LPW offers a potential solution in the quest to design a compact FG device for use with CTS. The algorithms presented in this thesis show that LPW allows for a reduction in the synaptic weight storage device while permitting the network to function as intended.

Acknowledgments

I would first like to thank my supervisor, Prof. Stephen Hall, for his assistance, guidance and knowledge which have greatly benefited me in both terms of my research and in preparation of this thesis.

Secondly I wish to thank my fiancée, Joanne, for all her support, advice and guidance during my PhD.

In addition I would like to thank Dr. L. McDaid, Dr. J. Marsland, Dr. T. Dowrick, Dr. S. Huang and Dr. A. Ghani for their collaboration on this work. Their support, advice and comments have been greatly appreciated.

I would also like to thank my colleagues at the University of Liverpool who have assisted me in various ways throughout my PhD: Dr D. Donaghy, Dr. R. Myers, Dr. L. Tan, Dr. I. Mitrovic, Dr. N. Sedghi, Dr. W. Davey, Mr. S. Boyes, Miss. S. Afzal, Mr. A. Edwards and Miss G. Carradice.

Finally I would like to thank all of my family and friends who have supported me throughout the duration of my PhD studies.

List of Symbols

Symbol	Significance	Unit
ϕ_B	Barrier height	eV
ϕ_S	Surface potential	V
μ	Mobility	$m^2V^{-1}s^{-1}$
A	Area	m^2
A	FN constant A	AV^{-2}
A_{FG}	Area of floating gate	μm^2
A_{poly}	Area of control gate	μm^2
A_{total}	Total area	μm^2
A_{tun}	Tunnelling area	μm^2
B	FN constant B	Vcm^{-1}
C'_{poly}	Interpoly oxide capacitance	Fm^{-2}
C_d	Depletion layer capacitance	F
C_{fox}	Field oxide capacitance	F
CG	Control Gate	-
C_{GS}	MOSFET gate-source capacitance	F
C_{max}	Maximum MOS capacitance	F
C_{min}	Minimum MOS capacitance	F
C_o	Oxide capacitance	Fm^{-2}
C_{ox}	Oxide capacitance	F
C_p	Interpoly oxide capacitance	Fm^{-2}
C_{poly}	Interpoly oxide capacitance	F
C_T	Total capacitance	F
C_w	Leakage capacitance	F
E_C	Conductance band edge	eV
E_F	Fermi energy level	eV
E_g	Energy gap	eV
E_i	Intrinsic Fermi level	eV

E_{ox}	Oxide Electric field	$MVcm^{-1}$
E_V	Valence band edge	eV
FG	Floating Gate	-
I_D	Drain current	A
I_{leak}	Subthreshold current	A
J_{FN}	Fowler-Nordheim Current Density	A/cm^2
k	Boltzmann's constant ($1.28x10^{-23}$)	JK^{-1}
L_{Cox}	Length of gate oxide	m
L_{poly}	Length of control gate	m
m	Gate-channel coupling coefficient	-
m_o	Electron Mass At Rest	kg
m_{ox}	Effective Mass of an Electron In The Insulator	kg
N_A	Acceptor doping concentration	m^{-3}
N_{elec}	Number of electrons	-
N_f	Total oxide charge density	Cm^{-2}
N_{FG}	Number density of charge on FG	cm^{-2}
n_i	Intrinsic doping concentration ($1.6x10^{16}$)	m^{-3}
N_{inj}	Number density of Q_{inj}	cm^{-2}
N_{injT}	Total Number density of Q_{inj}	-
P_{2CS}	Poly2 Layer Contact Spacing	m
P_{2CW}	Poly2 Layer Contact Width	m
q	Electron charge ($1.6x10^{-19}$)	C
Q_d	Depletion region charge	C
Q_f	Fixed oxide charge density	Cm^{-2}
Q_{FG}	Charge seen on FG per unit area	Ccm^{-2}
Q_g	MOS gate charge	C
Q_{init}	Initial charge on floating gate and in oxide	C
Q_{inj}	Charge injected	Ccm^{-2}
Q_m	Mobile oxide charge density	Cm^{-2}

Q_{sc}	Inversion layer charge	C
Q_t	Trapped oxide charge density	Cm^{-2}
Q_w	Total charge on FG	C
S	Subthreshold slope	mV/decade
T	Temperature (300K)	K
t_{cw}	Duration of critical timing window	µs
t_{fox}	Field Oxide Thickness	m
t_{inj}	Injection time	µs
t_{ox}	Oxide thickness	cm
t_{poly}	Interpoly Oxide Thickness	m
t_{post}	Time of post synaptic spike	µs
t_{pre}	Time of pre synaptic spike	µs
ΔV_w	Change in potential of Q_{inj}	V
V_A	Potential of Q_{FG}	V
V_b	Overall potential of FG	V
V_{buf}	Thresholding voltage of output buffers	V
V_C	Voltage on C	V
V_{CG}	Voltage applied to control gate	V
V_{DD}	Positive supply voltage	V
V_{DS}	Drain-source voltage	V
V_{FB}	Flat Band Voltage	V
V_{FG}	Capacitively coupled FG voltage	V
V_G	Gate voltage	V
V_{GS}	Gate-source voltage	V
V_{leak}	Voltage applied to subthreshold transistor M_{leak}	V
V_{mg}	Mid-gap voltage	V
V_{ox}	Semiconductor oxide voltage drop	V
V_{post}	Voltage of Postsynaptic Spike	V
V_{pre}	Voltage of Presynaptic Spike	V
V_{Qinit}	Potential of initial charge	V
V_{SS}	Negative supply voltage	V
V_{sub}	MOSFET substrate bias	V

V_t	Threshold voltage	V
V_{to}	AMS ideal threshold voltage	V
V_w	Potential of stored Q_w	V
V_{wi}	Input node to output buffer	V
W_{Cox}	Width of gate oxide	m
W_d	Depletion layer width	m
W_{df}	Equilibrium depletion width	m
W_{do}	Depletion layer width in deep depletion	m
W_{poly}	Width of control gate	m
α	Coupling coefficient	-
β	MOSFET gain factor	A/V
γ	Body effect factor	$V^{1/2}$
δ	Channel depth	m
Δt	Time step	µs
Δt_s	Time difference between pre- and post-synaptic spikes (was delta t)	µs
ε_0	Permittivity of Free Space	Fm^{-1}
ε_{ox}	Relative Permittivity of Silicon Dioxide	-
ε_{si}	Relative Permittivity of Silicon	-
λ	Channel length modulation factor	V^{-1}
τ_{CG}	Pulse width of V_{CG}	µs
Φ_b	Bulk Potential	eV
Φ_m	Metal Work Function	eV
Φ_{ms}	Work Function difference	eV
Φ_s	Semiconductor Work Function	eV
χ	Electron Affinity	eV

Contents

Chapter 1 – Introduction

1.1 Background

Over 50 years ago Bell laboratories developed the first working transistor based upon the work conducted by Julius Edgar Lilienfeld [1]. Two decades later the first silicon MOSFET, (metal-oxide-semiconductor field-effect transistor) was made commercially available. In 1965 Gordon Moore outlined the future of the semiconductor industry, stating that the number of transistors on a chip will double approximately every 18 months [2]. This has driven the semiconductor industry for over 40 years, and is envisioned to do so for at least another decade. Moore's Law has pushed manufacturers to fabricate transistors which are more efficient in both performance and power requirements with continuing reduction in dimensions. Semiconductor manufacturers, such as Intel, can now fabricate a MOSFET with minimum feature size of 32nm allowing in excess of 2 billion transistors to be placed in a dual-core microprocessor [3]. In 2011 the International Technology Roadmap for Semiconductors, ITRS 2011 [4], outlined how the semiconductor industry should proceed to pursue Moore's Law past the 18nm generation. It envisioned a concept of 'More than Moore', in which existing semiconductor technologies can be exploited to enable the fabrication of diverse systems and in particular systems which integrate non-digital and biologically based functionality.

Concurrently to the rapid progress in the semiconductor industry, a rapid expansion and an increase in research in the fields of microbiology, electrophysiology, and computational neuroscience, has also occurred. This activity has provided significant understanding and insight into the function and structure of the human brain [5]. Specifically the molecular components which are responsible for growth and modification of neural cells as well as the electrical characteristics of individual neurons have been identified. While the microprocessor has the ability to perform mathematical calculations and algorithms which are faster and more complex than that possible in the brain, there are several areas, such as visual and auditory processing, pattern recognition and classification, where the brain is superior [6]. This is due to the way that the brain processes information. The brain is a massive network of interconnected neurons and synapses which connect and operate in parallel. This allows for parallel processing of information rather than the serial approach taken by microprocessors. This has caused the fields of neural networks and neuromorphic engineering to emerge. A neural network is a paradigm which is inspired by the way in which biological systems operate, to solve problems which cannot be rapidly tackled by conventional techniques. Neuromorphic engineering is concerned with the design of electronic circuits that mimic the structure, processes and functions of biological systems. These circuits can then be used in the construction of artificial neural networks (ANNs) which closely resemble biological neural networks.

1.2 Neural Networks

The structure, operation and important features of biological neural networks are now presented in order to discuss the implementation of artificial neural networks (ANNs) and spiking neural networks (SNNs) in software and hardware. The main fundamental rules for synaptic plasticity, in particular spike timing dependent plasticity (STDP), are also presented in this section.

1.2.1 Biological Neural Networks

In the average human brain there are in excess of 10^{14} interconnected neurons, with each having up to 10^{3} synaptic connections [7]. A neuron is an elementary process block which typically operates slower than silicon logic gates. However due to the massive parallelism of the human brain, it can compensate for the slower operating speed. Fig. 1.1 shows the typical structure of a biological neuron.

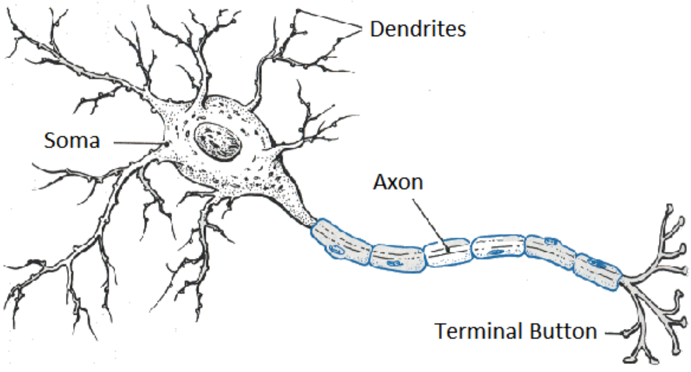


Fig 1.1 – Typical neuron cell

A typical neuron contains a soma (the cell body), an axon, dendrites and terminal buttons. A chemical (or electrical) synapse is formed when the axon of a presynaptic neuron forms a synaptic cleft with the dendrite of a postsynaptic neuron, Fig. 1.2. Communication between each neuron is achieved through chemical (or electrical) synapses. The terminal button has a membrane which is known as the presynaptic membrane and conversely on the cell or dendrite on which the synapse is acting there is a postsynaptic membrane.

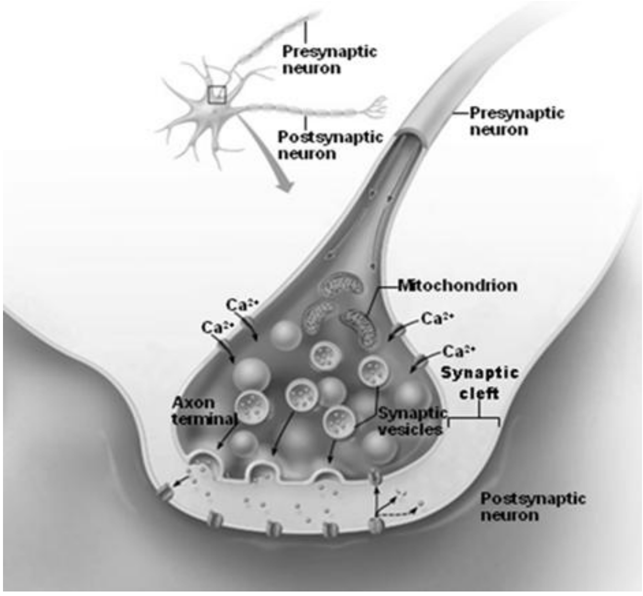


Fig 1.2 – Synaptic clef between two neurons [8]

Within the terminal button of the axon are synaptic vesicles, also known as neurotransmitter vesicles. Neurotransmitters are chemicals which are used to amplify, modulate and relay signals between one neuron and another neuron. The neurotransmitters are packaged within the vesicles and cluster beneath the presynaptic membrane. They are then released into the synaptic clef and bind to receptors located within the postsynaptic membrane causing ion channels to open or close. This affects the potential of the postsynaptic membrane. A synapse can be differentiated into two categories; Excitatory, where neurotransmitters serve to increase the potential of the membrane; Inhibitory, where neurotransmitters decrease the potential.

The soma performs temporal summation of incoming synaptic transmissions, and has a typical resting membrane potential of -65mV. If several combined synaptic inputs serve to increase the postsynaptic membrane potential (PSP), above a specific threshold voltage, then the neuron will generate an action potential. In this case the neuron is said to fire. This action potential propagates down the axon of the postsynaptic neuron causing synaptic connections to be activated. Fig. 1.3 shows the generation of an action potential when three successive synaptic inputs occur. The action potential occurs on the occurrence of the third input. After the action potential has been generated the neuron enters a refractory period. The refractory period occurs when the membrane potential undershoots its resting potential, causing the neuron to enter a state of hyperpolarisation. The refractory period is due to inactivity in the voltage-gated sodium channels and a lag in the closing of potassium channels within the neuron. As the potassium channels begin to close, the membrane potential begins to return to its resting potential, causing the refectory period to last up to milliseconds.

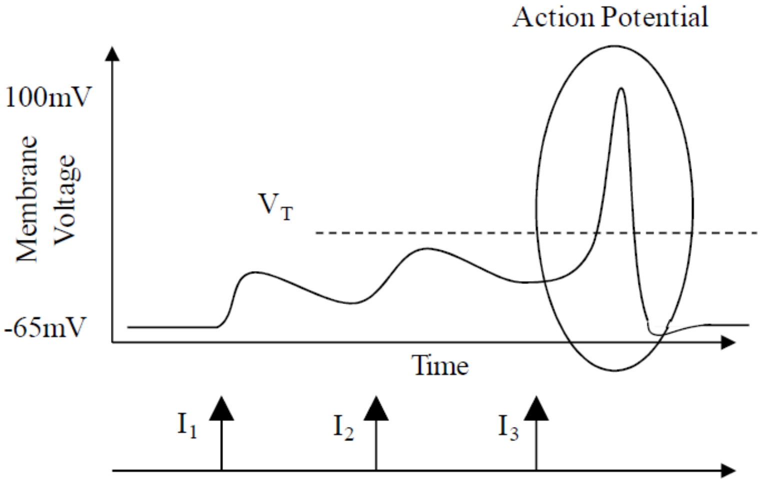


Fig 1.3 – PSP response to three synaptic inputs and generation of an action potential [9]

1.2.2 Synaptic Plasticity

The synapse is responsible for adaption and learning within the neural network, through the modification of the synaptic weight. A synapse has the ability to influence the firing of the postsynaptic neuron either as an individual or as part of several synaptic inputs. A strong synapse has the ability to cause the postsynaptic neuron to fire in the absence of additional synaptic inputs. In contrast a weak synapse may not have any effect on the firing of the postsynaptic neuron. However since a synapse has the ability to change its strength, a weak synapse could eventually become strong and vice versa. Hence the synaptic weight can either be increased, (potentiation), or decreased, (depression). This modification can either be; short term, STP (short term potentiation) or STD (short term depression); or long term, LTP or LTD change in the synaptic weight.

To explain what actually takes place within a biological synapse to cause a change in the synaptic weight, it is important to firstly understand that in biology there are two main types of synapse, namely electrical and chemical. It is the latter of these which exists in greater numbers within a biological neural network [11 12]. A chemical synapse, hereafter referred to as a synapse, passes information from the presynaptic neuron cell to the postsynaptic neuron cell via the release of neurotransmitters into the synaptic cleft, [11 12]. The synaptic vesicles are released from/contained within the presynaptic axon via the opening or closing of voltage-gated calcium ion, Ca^{2+} channels. In the postsynaptic neuron, neurotransmitter receptors interact with the released neurotransmitters to cause some change in potential of the postsynaptic cell [4 7, 8-12], Fig. 1.3.

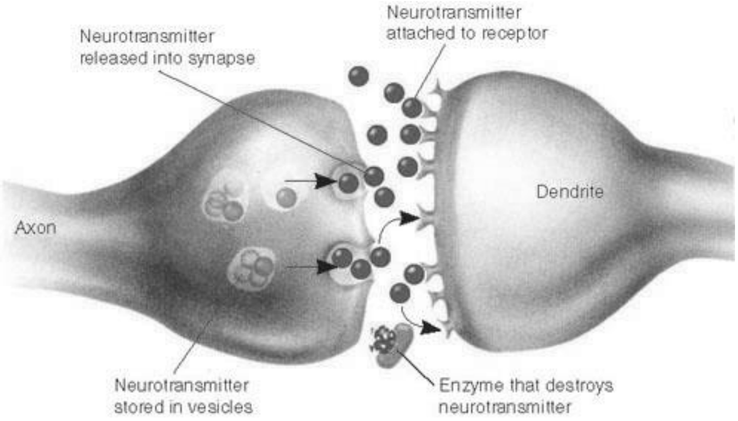

Neurotransmitter released into synapse
Neurotransmitter attached to receptor
Axon
Dendrite
Neurotransmitter stored in vesicles
Enzyme that destroys neurotransmitter

Fig 1.3 – Cross section of synaptic cleft between two neurons [146]

Within the postsynaptic neuron cell, there are two main types of neurotransmitter receptors; NMDA receptors, NMDARs, and AMPA receptors, AMPARs. LTP and LTD are the main causes of synaptic weight change. It is thought that LTP and LTD cause changes to both the presynaptic and postsynaptic neurons through changes in neurotransmitter receptors [12-19]. In [14] it is proposed that NMDARs play an important role in the induction of LTP and LTD within the synapse. Specifically the structure, composition and number of the NMDARs can change depending upon whether LTP or LTD is taking place. During LTP it is widely accepted that the concentration and permeability of NMDARs is increased within the postsynaptic neuron. Conversely during LTD the concentration of NMDARs are reduced and as such the probability of neurotransmitters interacting with the receptors is also reduced [11, 12]. In addition to NMDAR modifications, AMPARs also experience change during LTP and LTD [18]. During LTP the number of AMPARs increases in the plasma membranes at the synapse. This is done through activity-dependent changes in the biophysical properties of AMPARs in the synapse. This further enhances the effect that the presynaptic neuron has on the postsynaptic neuron [18]. LTD does not cause this modification. It is also suggested that due to the increase in NMDARs in the postsynaptic neuron during LTP, an increase in Ca^{2+} also occurs [14]. This indicates that LTP and LTD are also based upon the amount of Ca^{2+} which is present in the postsynaptic neuron. Synaptic weight can be quantified of as the number of NMDARs, the modification effect of AMPARs and the quantity of Ca^{2+}.

Hebb's theory [20] describes how the synaptic weight is allowed to change, based upon the inputs and outputs of each neuron within the neural network. Hebb states that;

"when an axon of cell A is near enough to excite a cell, B, and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency as one of the cells firing B, is increased".

When the input from neuron A, meets or exceeds the threshold value of neuron B (and this is a repeated process) then the synaptic weight between neuron A and neuron B is increased. Conversely if A has little or no effect on B then the synaptic weight is either kept constant or reduced, depending upon additional factors [20].

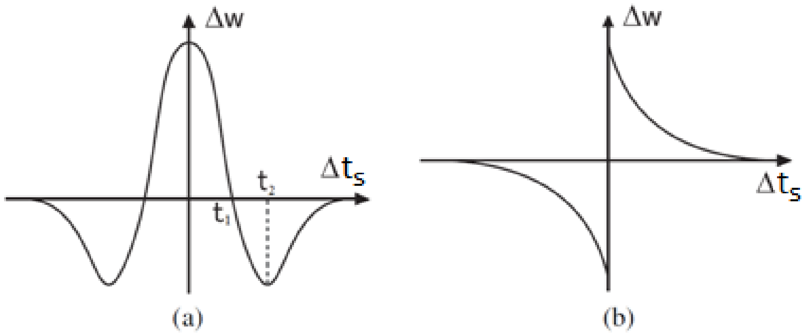


Fig 1.4 – (a) Symmetric (b) Asymmetric STDP Curves

A further development of the Hebbian learning concept was the introduction of spike timing dependent plasticity, STDP, in 1983 [21]. STDP is concerned with modification, (increase and decrease) of the synaptic weight based upon the relative timings of pre- and post-synaptic spikes. In biological neural networks, there are two main types of STDP; symmetric, Fig. 1.4(a), and asymmetric, Fig. 1.4(b), [21-28].

In symmetric STDP the synaptic weight is updated under the following conditions:

- If the time difference between the occurrence of the pre and post synaptic input, $\Delta t_s = t_{post}$-t_{pre}, is equal to 0, then the maximum positive weight update, Δw, occurs. This increases the stored synaptic weight.
- As $\Delta t_s \rightarrow t_1$, then Δw beings to decrease in magnitude, (but is still positive) until at $\Delta t_s = t1$, $\Delta w=0$. Again this still has the effect of increasing the stored synaptic weight. When Δt_s increase past t_1, Δw now becomes negative, thus beginning to decreasing the stored synaptic weight. As $\Delta t_s \rightarrow t_2$, Δw is such that it tends towards the maximum negative weight update, at $\Delta t_s = t_2$. Finally when $\Delta t_s > t_2$, $\Delta w \rightarrow 0$.

Symmetric STDP modification of the synaptic weight is dependent upon Δt_s only and not the temporal order of pre- and post-synaptic spikes occur. Asymmetric STDP modifies the synaptic weight based upon the temporal order of pre- and post-synaptic spikes and Δt_s. The synaptic weight, w, is updated under the following conditions;

- Increased if a pre synaptic spike occurs prior to a post synaptic spike (pre-post spiking) – this is LTP.
- Decreased if a post synaptic spike occurs prior to a pre synaptic spike (post-pre spiking) – this is LTD.

The magnitude by which the synaptic weight is modified, Δw, is determined by;

- $|\Delta t_s| \rightarrow 0$, $\Delta w \rightarrow |\Delta w_{max}|$
- $|\Delta t_s| \rightarrow \infty$, $\Delta w \rightarrow |\Delta w_{min}|$ ($\Delta w \rightarrow 0$)

Of the two STDP rules presented, it is asymmetric STDP, (hereafter referred to as STDP), which is thought to occur more frequently in biology [22, 26-28]. It is worth noting that the exponential functions shown in Fig. 1.4(b) are not a pre-requisite for STDP but rather are a mathematical convenience. It is only necessary for the relative timings to produce reinforcement or reduction of the weights to be realized.

1.2.3 Artificial Neural Networks

Artificial Neural networks, ANN, are mathematical or computational representations that are inspired by and aim to emulate the structure and operation of biological neural networks. ANNs consist of simplistic processing elements (neurons) which can be interconnected to generate a neural network. Typically ANNs are used when either a highly complex or no algorithm exits to solve the specified problem. Hardware [34-37] and software [38-41] applications include; pattern recognition, image processing, control and robotic systems. While software implementation is the easier of the two to implement however in order to simulate large complex networks additional computational resources are required. Additionally current computer processors operate in serial. Therefore it is not possible to accurately simulate a large, complex, parallel system using software. Hardware ANNs offer a greater deal of biological plausibility, flexibility and potential in terms of parallelism, real-time operation and speed. Fig 1.5 presents an example of an artificial neuron which can be trained to find an optimal solution and solve a specified problem. This is done by a learning rule which modifies the weighted connections (synapses) between each neuron.

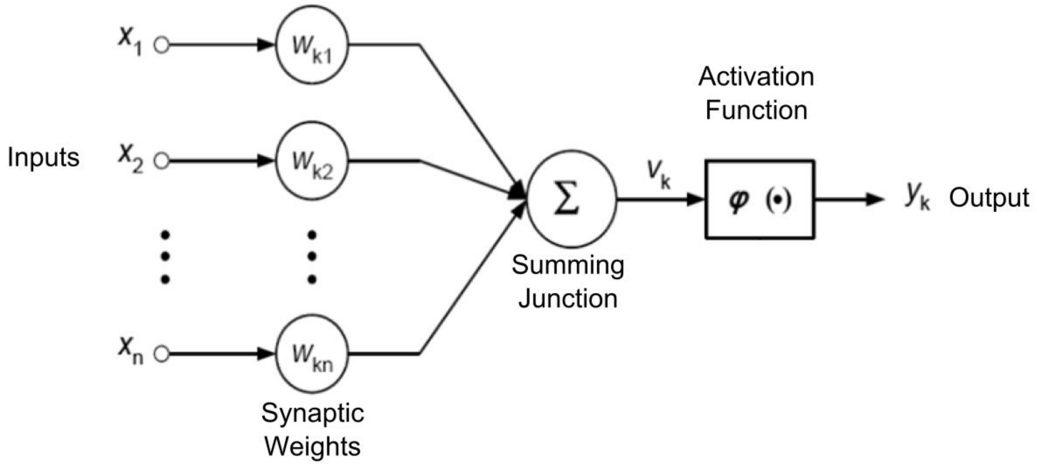


Fig 1.5 – Artificial neural network neuron model with X_1 – X_n represent the inputs to the neuron, W_{k1} - W_{kn} represent the synapses and their associated weight, and y_k is the output from the neuron.

The output from the summing junction is given as;

$$V_k = \sum_{j=1}^{n} w_{kj} x_j \tag{1.1}$$

and the output from the neuron, y_k, is;

$$y_k = \varphi(v_k) \tag{1.2}$$

$\varphi(\cdot)$ represents an activation function which determines when the neuron will fire. The first generation of ANN consisted of McCulloch-Pitts neurons. McCulloch-Pitts view neurons as computational units which use a threshold activation function. McCulloch-Pitts neurons produce an output of 1 when the weighted sum of its inputs is $V_k > 0$. If the weighted sum is $V_k \leq 0$ an output of 0 is produced. Within the network information is coded by the presence or absence of action potentials. Even though McCulloch-Pitts neurons produce a digital output, they have been successfully applied in networks of multilayer perceptron or Hopfield nets, where the overall output required is Boolean.

Threshold activation functions were replaced with continuous activation functions in the second generation of ANN. Continuous activation functions, such as sigmoid or hyperbolic tangent functions, allowed for analogue inputs and output to be used. An example of a logistic sigmoid function is shown in equation 1.3, where a represents the slope parameter for the activation function.

$$\varphi(v) = \frac{1}{1+exp(-av)} \tag{1.3}$$

Modification of the synaptic weight will alter the flow of information within the network. The strength of the inputs to the neurons are altered such that the neuron's weighted sum output is also changed. This forms the simplified basis for learning and synaptic plasticity. In the second generation of ANN, a continuous activation function was implemented. This allows for learning rules based upon gradient-decent algorithms to be used which can train and change the weights between neurons and aid in the network accomplishing its task.

Figure 1.6 shows an example of a fully connected feed forward neural network which can be implemented using so-called second generation artificial neurons. In this network information is only passed forward in the network from input to output. No feedback of information from output to input or hidden layers occurs. In more complex networks feedback paths are included as they offer greater stability and parallels with biological networks. Recurrent networks and networks which require digital outputs can also be implemented using the second generation of ANN.

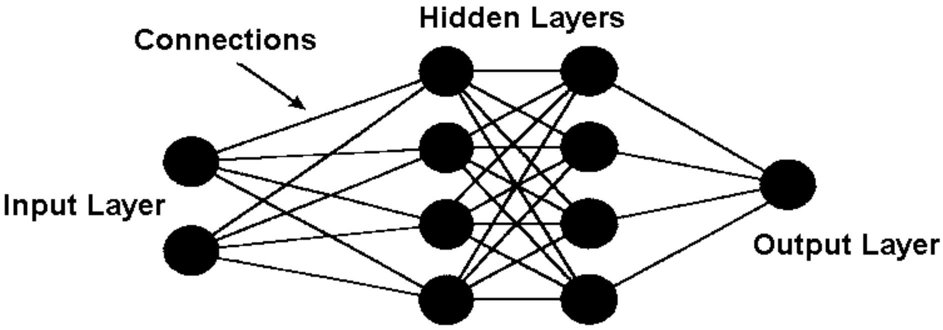


Fig 1.6 – A fully connected feed forward neural network consisting of two input neurons, 4 neurons in the first hidden layer, 4 neurons in the second hidden layer and 1 neuron in the output layer

1.2.4 Spiking Neural Networks

Both the first and second generation of ANNs have been shown to emulate the main features of biological neural networks; plasticity, summation and thresholding. However they do not capture all of the biological features of a neural network. Experimental results indicated that rate coding does accurately describe activity in the brain. Rate coding implies an averaging mechanism is used. However in biology, neurons operate by “spike or no spike” mechanism with a base firing rate, described as an intermediate frequency of spiking. While the second generation can implement this intermediate frequency of spiking, it still lacks some important biological features. Recent research has suggested that the timing of individual action potentials, spikes, is actually used to code information within the brain [29-30]. Neurologists also believe that the human brain’s computational power is in its ability to process large numbers of these spikes in parallel [30]. Thus spike-based coding schemes are considered to be more efficient than rate-based coding [31].

The third generation of artificial neural networks, spiking neural networks, SNNs, raises the biological plausibility by utilizing individual spikes. SNNs are designed such that the scale and connectivity observed in the brain is mimicked in hardware. Specifically hardware neural networks need to occupy a minimum circuit area and have low power consumption. Additionally SNNs build upon previous generations of ANN, by emulating various neuronal functions: spatial and temporal summation of input signals, threshold, plasticity, refractory periods and learning. Computation and communication of information within the network is done through the spatial-temporal summation of individual spikes, which is akin to biological neurons. Therefore since SNNs use pulse coding as opposed to rate coding, this allows for information to be multiplexed as both frequency and magnitude. SNN do not fire at the end of each propagation cycle, rather they only fire when the weighted sum of their inputs causes the membrane potential of the neuron to exceed its threshold value. In [30] it has been shown that spiking neural networks have more computational power over equivalent networks of static neurons.

1.2.5 Spiking Neuron Model

Hodgkin and Huxley in 1952 developed the first scientific model of a spiking neuron, establishing the idea that ionic currents give rise to action potentials, spikes, within the cell membrane [33]. They proposed that the ion current, conductance and membrane potential in the neuron must be considered when modelling neurons.

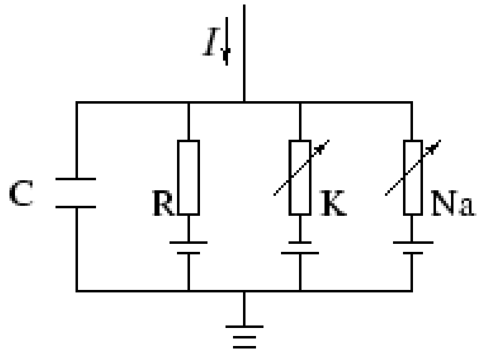


Fig 1.7 – Hodgkin and Huxley Neuron Model

In Hodgkin and Huxley's neuron model, shown in Fig. 1.7, the cell membrane of the neuron is represented by the capacitor. When the capacitor voltage exceeds the threshold, charge is released. The period of time for which it takes to discharge the capacitor can be analogous to the refractory period of a cell membrane once it has fired. The two main ion channels which occur in a biological neuron, the sodium and potassium ion channels are represented by the two non-linear conductance's K and Na respectively. The conductance R represents a leakage channel within the neuron. If a current is injected into the circuit, then it will either add more charge to the capacitor or leak away through the three channels. The circuit was used to model the shape of an action potential of a biological neuron within the axon of a giant squid [33].

While the Hodgkin-Huxley model accurately reproduces the shape of action potentials, refractory period, rest response and repetitive firing, it is complex and nonlinear. Simplified models with a higher level of abstraction, such as the integrate-and-fire (I&F) neuron were established to provide an intrinsic understanding of the structures of neural excitation and network behaviour. The I&F model considers a neuron as a homogeneous unit which will only generate a spike if the total excitation is sufficiently large [42]. An improvement to the I&F model, 'leaky I&F' shown in Fig. 1.8, which takes into account that the biological membrane potential will decay over a set period of time known as the membrane time constant. The leaky I&F model performs temporal summation of all the input spikes. If the total of this summation is greater than the threshold voltage, then this will cause the neuron to fire, producing an output spike.

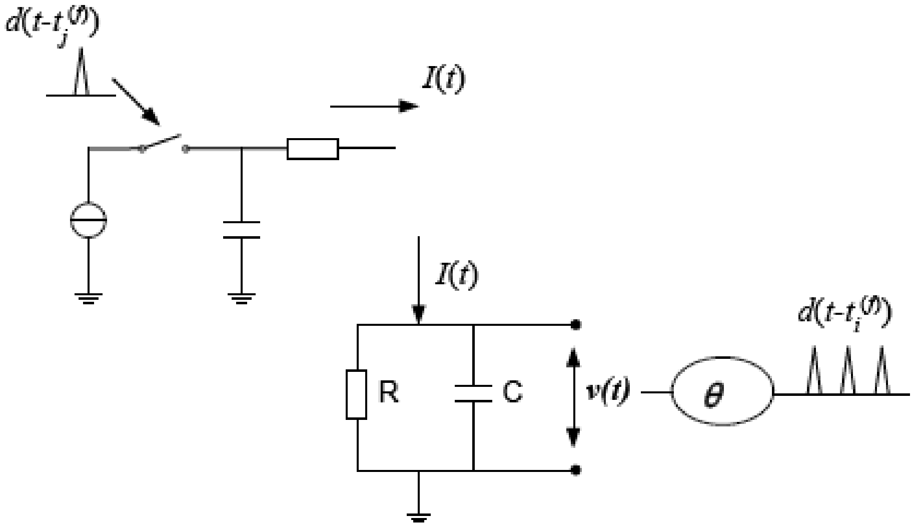


Fig 1.8 – Equivalent circuit for the leaky I&F neuron model [43]

If a current I(t) is injected into the neuron model, capacitor C is charged. Resistor, R, is provided such that the membrane potential can be reset when no current is present within the circuit. This is the path for the leaky current. Therefore I(t) can be given as follows [33];

$$I(t) - \frac{V(t)}{R} = C\frac{dV(t)}{dt} \tag{1.4}$$

$$I(t) = C\frac{dV(t)}{dt} + \frac{V(t)}{R} \tag{1.5}$$

$$\tau\frac{dV}{dt} = V(t) + RI(t) \tag{1.6}$$

Voltage V(t), represents the membrane potential and τ is the time constant of the leaky integrator. Considering the case when the membrane has a constant current, I(t)=I_0, resetting potential of zero, and $t^{(1)}$ denotes the time of the occurrence of the first spike, then;

$$V(t) = RI_0\left[1 - exp\left(\frac{t-t^{(1)}}{\tau}\right)\right] \tag{1.7}$$

If V(t) < θ, the threshold value, then no output spike is produced. However if V(t) > θ then a spike will occur a time $t^{(2)}$. Once the second spike has occurred, the membrane potential is reset and the process can start again. Finally the threshold condition, the time at which the second spike occurs, $t^{(2)}$, can be found from 1.8;

$$\theta = RI_0\left[1 - exp\left(\frac{t^{(2)}-t^{(1)}}{\tau}\right)\right] \tag{1.8}$$

Biological neurons use spikes and spike trains to encode information [29, 30]. Additionally spatial-temporal information is also incorporated in a biological network [44]. Therefore any SNN model should incorporate the inter-spike interval, ISI. The ISI is defined as the time between two successive spikes within a spike train [45]. If the ISI between spikes can be found, then an idea of how spatial-temporal coding of information takes place within neural networks can be established. Rearranging 1.8 gives the equation for the ISI, T.

$$T = \tau\ln\left(\frac{RI_0}{RI_0-\theta}\right) \tag{1.9}$$

A significant amount of research has been undertaken in assessing the computational power and implementation of SNN in both software and hardware [34-41]. While the implementation of SNN in software is often easier to realise, there are disadvantages. If the network size is continually increased, the computational resources required to simulate the networks increases significantly. This can result in an increase in the processing power as well as a reduction in the operational speed of the proposed SNN. Hardware neural networks by contrast offer a greater degree of plausibility and benefits in terms of operational speed and real-time parallel processing ability. By increasing the size of a SNN in hardware, the complexity of the network is increased, but the operational speed of the network can remain unaffected. Additionally it is possible to build a hardware SNN with neurons that operate in parallel in real-time.

However there are drawbacks and problem areas which need to be taken into consideration when implementing hardware neural networks. Firstly the brain shows that parallel processing can be achieved through massive connectivity between neurons in a small area. Therefore any SNN in hardware must achieve a similar plausible scale of connectivity while occupying a small circuit area. In addition, hardware SNN must have minimal power consumption, akin to that of biological networks. Secondly hardware SNN must capture the computational power of the brain. This is to say that hardware SNN must implement; spatial-temporal summation of signals, thresholding, synaptic plasticity, refractory periods (to prevent unwanted synaptic weight change) and learning. The final area which must also be considered is that of interconnect. The ability to connect multiple neurons using traditional metal connections can hinder the massive connectivity which occurs in biology. Therefore alternative interconnects between neurons are to be sought; RF, optical, AER (address event representation) and Network on a Chip [36 47-50] are a few of the possible methods which can be used.

1.2 VLSI Neural Networks

The field of neuromorphic engineering has evolved over the last decade’s years [53-69]. Neuromorphic engineering has focused on the implementation of biological neural networks in hardware using VLSI techniques and custom designed silicon circuits. Semiconductor devices in particular are becoming common device building blocks for emulating neural system in hardware. This is because semiconductor devices contain physical properties which aid in the modelling of neural systems. Designers exploit these features to build massively parallel systems. In biology, in particular nervous tissue, information is encoded and manipulated in terms of charge conservation, which occurs naturally in semiconductor circuits. Electrons in semiconductor devices, like ions within nerve tissue, are in thermal equilibrium with respect to their surroundings. The fundamental forces which cause ion flow within biology are akin to those which cause electron flow within MOS transistors operating at low currents, [52]. In addition, semiconductor devices have mechanisms which can be used for long-term memory storage which allows synaptic plasticity and learning mechanisms akin to that found in biology [70].

The use of VLSI has provided a powerful resource which allows for analogue, digital and mixed-signal implementation of SNN in hardware. Analogue implementations can exploit the physical properties of semiconductor devices to emulate neural features in hardware. The main advantage of analogue VLSI is that operational speed and packing densities are higher compared to digital VLSI. This allows for more efficient and massively parallel systems in analogue VLSI. In addition to this analogue VLSI neural networks can operate and integrate with real time, real analogue world systems without the need for DAC and ADC [52]. However disadvantage of analogue VLSI is that they are sensitive to noise and prone to process variations, particularly at low currents, as well as interference.

Digital VLSI systems offer a greater computational power, higher precision, programmability and reliability over analogue VLSI. In addition, synaptic weights can be stored either locally on chip or more frequently, externally off chip. However in order to achieve these features, digital VLSI circuits suffer from reduced computational speeds and an increase in the circuit density and power. Additionally there still exist several limitations in VLSI networks in that consideration and attention needs to be given to the adaptability, scalability, flexibility and a maximization of speed with respect to conventional sequential processors. Hybrid analogue and digital systems in VLSI offer advantages over purely analogue or digital VLSI networks in that, dense, massively parallel, integrated neural circuits that operate in real time can be built. These circuits have the propensity to capture the computational power and efficiency of biological systems in hardware.

Examples of elementary biological circuits which replicate the functionality of various neural systems in silicon have been established over the years. These massively parallel signal processing systems have successfully implemented; silicon retinas [54, 59, 63, 68-76], silicon cochlear [77, 78, 84,85], auditory midbrain [79], olfaction chips [88, 89] and motion sensing [77, 79, 80-83].

In [54] Mahowald and Mead proposed a silicon retina which generates an analogue output in response to its detection of the contours of a moving input stimuli. An alternative silicon retina by Zaghloul and Boahen [63, 76] consist of an array of 60x96 phototransistors and processing circuit. Zaghloul and Boahen's retina can generate an output spike which mimics the response of both ON-sustained and OFF-sustained ganglion cells within the retina. An adaptive olfaction chip produced by Koickal et al [89] using analogue VLSI has also been studied. The inclusion of both an on-chip chemosensor array and sensor interface has allowed them to integrate with on-chip spike timing dependent learning circuits to dynamically control synaptic weights to allow for odour detection and classification. Finally Rasche [68] presents an adaptable and excitable membrane within a network of spiking units. The units have been designed such that they can be integrated in to other neuromorphic systems to allow implementation of various visual neural networks including but not limited to; contour detection and propagation, image segmentation and processing as well as motion detection [83].

In order to implement higher levels of cognition and processing, several novel multichip approaches and communication protocols between multiple chips have been examined [66, 69, 90-93]. The neuromorphic systems presented use a design strategy akin to that found in biology; local computations are performed in analogue and the results are communicated between neurons using all-or-none binary spiking events. AER, address-event representation is a common communication protocol language used within neuromorphic chips [67, 93-101]. AER uses time-multiplexing to emulate the massive connectivity found in biology. The address encoder generates a unique address for each neuron when a spike occurs. The address is then transmitted along a digital bus to the receiving chip which proceeds to decode the address and selects the corresponding location. This is an asynchronous protocol where the time that the address appears on the bus directly encodes the spike time. In [92] another neuromorphic multichip is outlined which implements orientation hypercolumns in mammalian primary visual cortex. These consist of a single silicon retina which feed multiple orientation selective image filtering chips [91]. Each of these chips contains 2-D arrays of neurons each tuned to the same orientation and spatial frequency but are in different retinal locations. All of the chips operate in continuous time and communicate through spike-encoded inputs and outputs which are transmitted using digital asynchronous AER protocol.

Other projects have instead focused on implementing more complex and biologically accurate neural networks in hardware. The Blue Brain project aims to create a detailed simulation model of the neural activity of a human brain. The first stage of the project completed in 2006 was to simulate the neocortical column of a rat using IBM's BlueGene supercomputers [102]. BlueGene successfully simulated a neocortical column of 100,000 complex neurons down to the cellular level. In addition to this BlueGene can also simulate 100 million 'simple' neurons in order to understand how a mouse's brain functions. The project main focus is more on obtaining and simulating a biologically accurate model to further the functionality, potential and understanding of the human brain, than to find potential applications for the models generated.

The SpiNNaker project aims to simulate in real-time a billion neurons [102-106]. The SpiNNaker chip is a massively-parallel multiprocessor chip with a fault-tolerant architecture designed to implement various neuron models while mimicking neural computation [107]. The SpiNNaker chip contains 20 ARM968 processing cores with dedicated memory, and 1GB of off-chip SDRAM which is used to hold the synaptic weight of each synapse. SNN have been implemented using SpiNNaker chips, [103], which implement a modified Izhikevich model, [109]. In addition, STDP has also been implemented, [108].

Finally the FACETS project aimed to implement computational architectures inspired by biology which exploit both the connectivity and biophysical behaviour of neural networks which underline human cognition [110-114]. The main focus of the FACETS project was to gain an understanding of the computational power of the human brain, echoing the aims of the Blue Brain Project. The project aimed to achieve this through the wafer scale integration of analogue processing elements to produce a neuronal chip consisting of 384 analogue Hodgkin-Huxley neurons with a total of 100,000 synaptic connections on a single silicon wafer. In addition to this several new novel computational paradigms have been identified. Specifically a general model, the Liquid State Machine, has been developed which provides a biologically realistic version of computation in cortical microcircuits. This is a universal model for analogue computations and is an improvement on the alternative Turing machine model [146].

The aforementioned projects while useful in gaining an understanding in the accurate modelling and functionality of the computational power of the human brain have drawbacks. Specifically all three projects require large areas of silicon (physical space), have a high cost to implement and power consumption compared to customized silicon circuits. Customized circuits aim to emulate the functionality of brain functions without the high level of biological accuracy employed by the BlueBrain, SpiNNaker and FACETS projects.

1.2.1 Silicon Neural

The neuron is the main building block of biological neural networks. It is the central processing unit which acts upon all inputs, deciding whether an output should be generated. There are two main design routes, principles, which determine the design of hardware neurons. The first route is to design conduction based neurons [115, 116]. In this case the silicon neuron faithfully recreates and mimics the channel conductance which is seen in biological neurons. In particular the membrane and ionic currents which occur within the neuron are modelled accurately. Ion currents cause nerve impulses, action potentials, as well as the discharge of these impulses within a neuron. The most common type of conduction model which is realized in hardware is the model by Hodgkin-Huxley [115]. In [115] it is proposed that the ion currents within a neuron can be modelled using standard MOS technology.

Farquhar & Hasler, [116], echoed the idea that exploiting MOS device characteristics can allow the ion currents of a biological neuron to be emulated. Farquhar & Hasler proposed that the two main ion currents can be modelled individually. A sodium circuit using four transistors and two capacitors models the sodium ion current, Fig. 1.9. The potassium circuit model (two transistors and one capacitor) models the potassium ion current, Fig. 1.10.

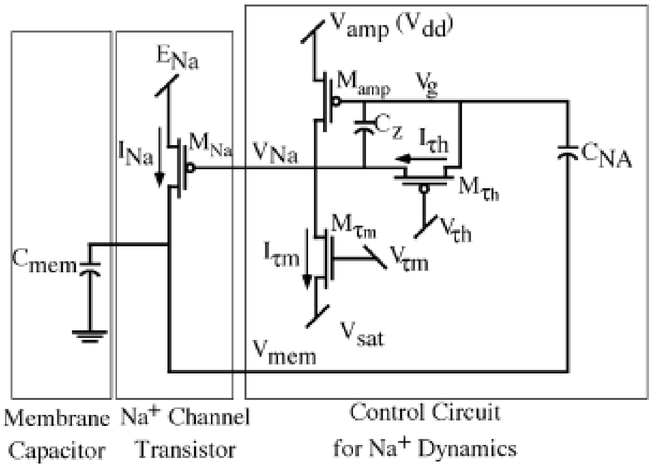


Fig 1.9 – Equivalent Farquhar & Hasler Sodium Current Model [46]

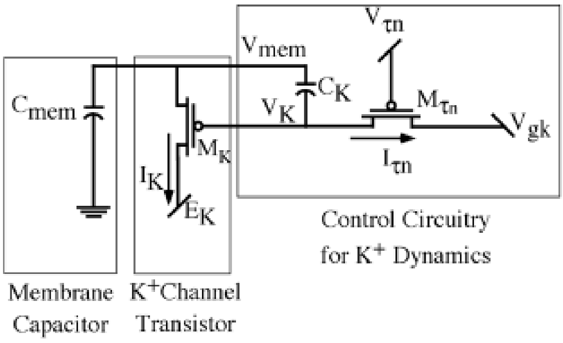


Fig 1.10 – Farquhar & Hasler Potassium Current Model [46]

The ion currents flow through a membrane and are non-linear, having an exponential relationship to the voltage across the membrane. In biological systems the sodium channel is voltage gated; it will respond to changes in voltages across the membrane. Additionally it contains both activating and inactivating mechanisms which cause changes in the currents magnitude as these increase or decrease, a similar response to that of a band pass filter. The potassium channel is slightly different in that it is only activating, thus is akin to a low pass filter. In [117], the MOSFET is proposed to be analogous to the bio-channel. When it is operated in sub-threshold it produces a current which is comparable to the ion currents within a biological neuron. The voltage controlled channel which exists between the source and the drain is comparable to a feature of a biological neuron; the pore, the physical structure through which the ion currents flow. The gating membrane, which controls the pore, is analogous to the gate of the MOSFET.

Duperyron et al [118] developed a more complex conductance-based neuron using multiple integrated circuits. The neurons are based Hodgkin and Huxley formulism, replicating the ionic currents and its dependency upon the potential of the membrane in hardware. The circuit sums three currents at a membrane capacitor, producing an action potential, akin to biological neurons, if greater than the membrane potential. Despite the comparisons to the dynamic features and functionality of a biological neuron, the overall size and power requirements of the conductance based neurons make them unfeasible for use in real applications. In [55] an alternative silicon neuron is proposed which is akin to the designs by Hodgkin and Huxley model [33]. The proposed neuron is a sophisticated circuit designed to replicate, efficiently, the ion channels with a biological neuron to a high degree of biological accuracy. The circuit operates in real time and has low power consumption. However the neuron circuit is complex requiring over twenty MOSFETs.

While conduction-based neurons aim to faithfully recreate the channel conductance as well as other features of biological neurons. there is a trade-off between the complexity and functionality, and the size and power requirements [55, 115-118.] This has led to second design route and the rise of phenomenological neurons. Phenomenological neurons are designed to reproduce certain computational properties, features, of biological neurons as efficiently as possible; including spiking, plasticity and leakage.

Fig 1.11 presents several phenomenological neurons realised in silicon. In [52] Mead presents a axon-hillock circuit consisting of an integrated capacitor connected to two inverters and a feedback capacitor with a reset transistor driven by an output inverted, Fig 1.11(a). Output spikes are the generated when a voltage across the integrator capacitor (representing the membrane potential) exceeds the switching threshold of the first inverter. The circuit in [52] is commonly used in VLSI neural networks [62]. There are drawbacks to the approach undertaken in [52, 62] in that it requires a large amount of power and the switching threshold of the inverter is dependent upon process parameters. It does not model any additional biological characteristics such as refractory periods. In [119] a neuron cell is proposed which utilizes an electrically programmable conductance, Fig. 1.11(b).

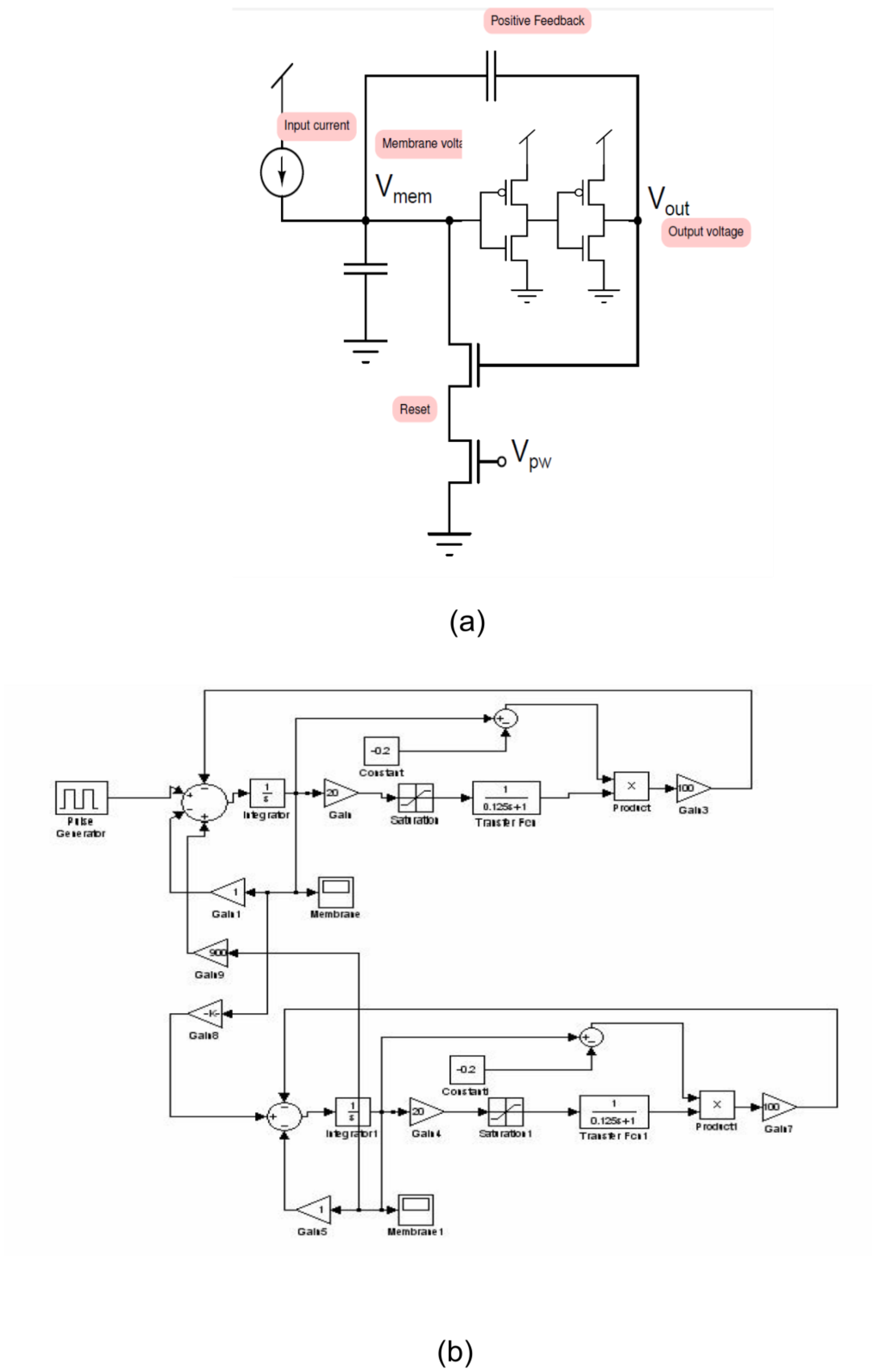
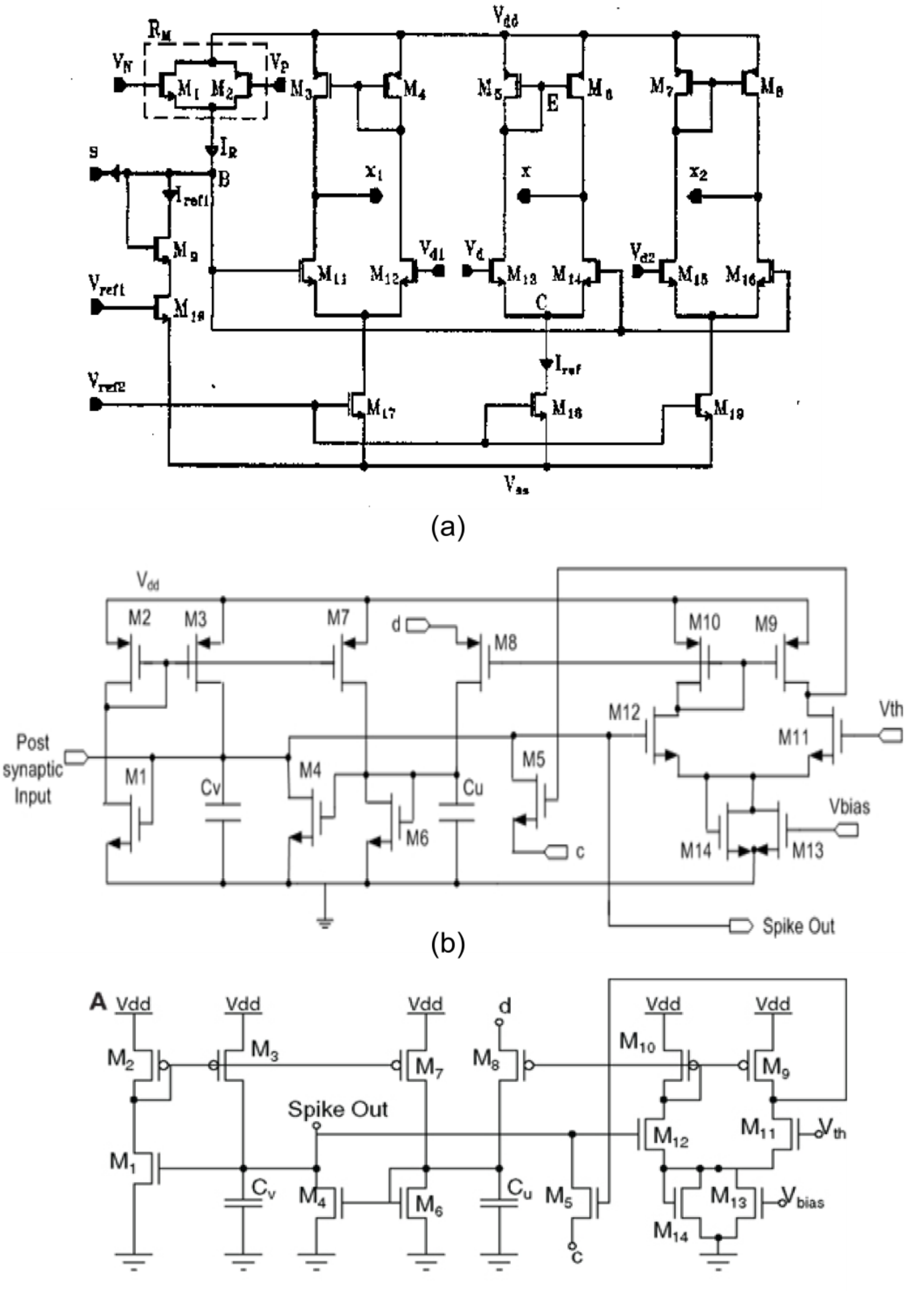


Fig 1.11 – Neuron Circuits – (a) [52], (b) [119]

Chun et al [120] propose a neuron circuit comprising of 19 MOSTs, Fig. 1.12(a). The circuit contains programmable parameters which can be used to adjust the gain of the output from the neuron as well as the threshold of membrane potential of the neuron. Fig. 1.12(b) [121] presents an oscillator circuit for use with Izhikevich neurons [122], Fig 1.12(c). The combination of the two circuits, requiring up to 14 MOSFETs, allows for the simulation and implementation of a range of cortical neuronal cells.



(a)

(b)

(c)

Fig 1.12 – Neuron Circuits – (a) [120], (b) [121], (c) [122],

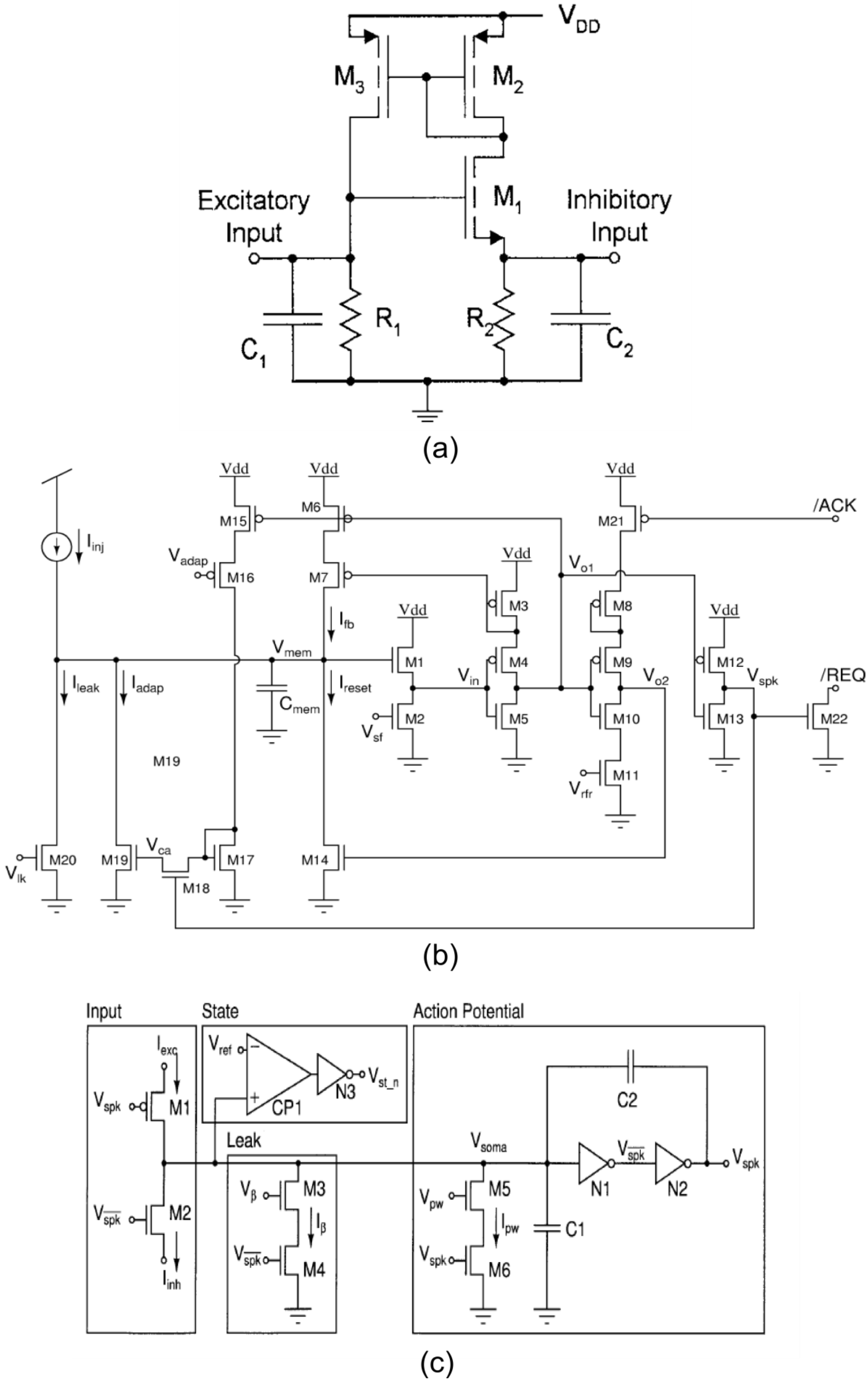
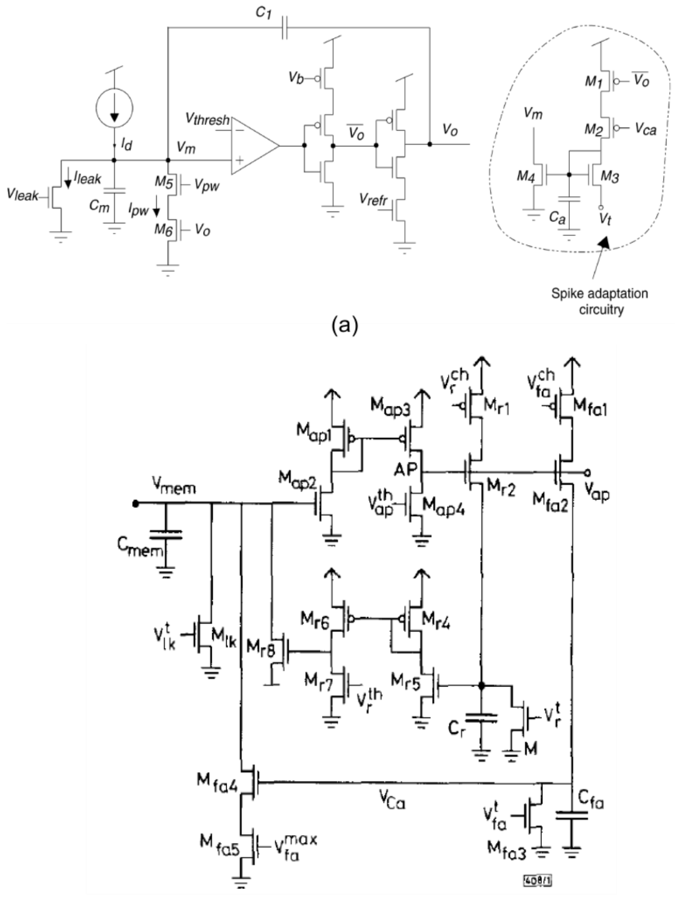


Fig 1.13 – Neuron Circuits – (a) [123], (b) [36], (c) [62]

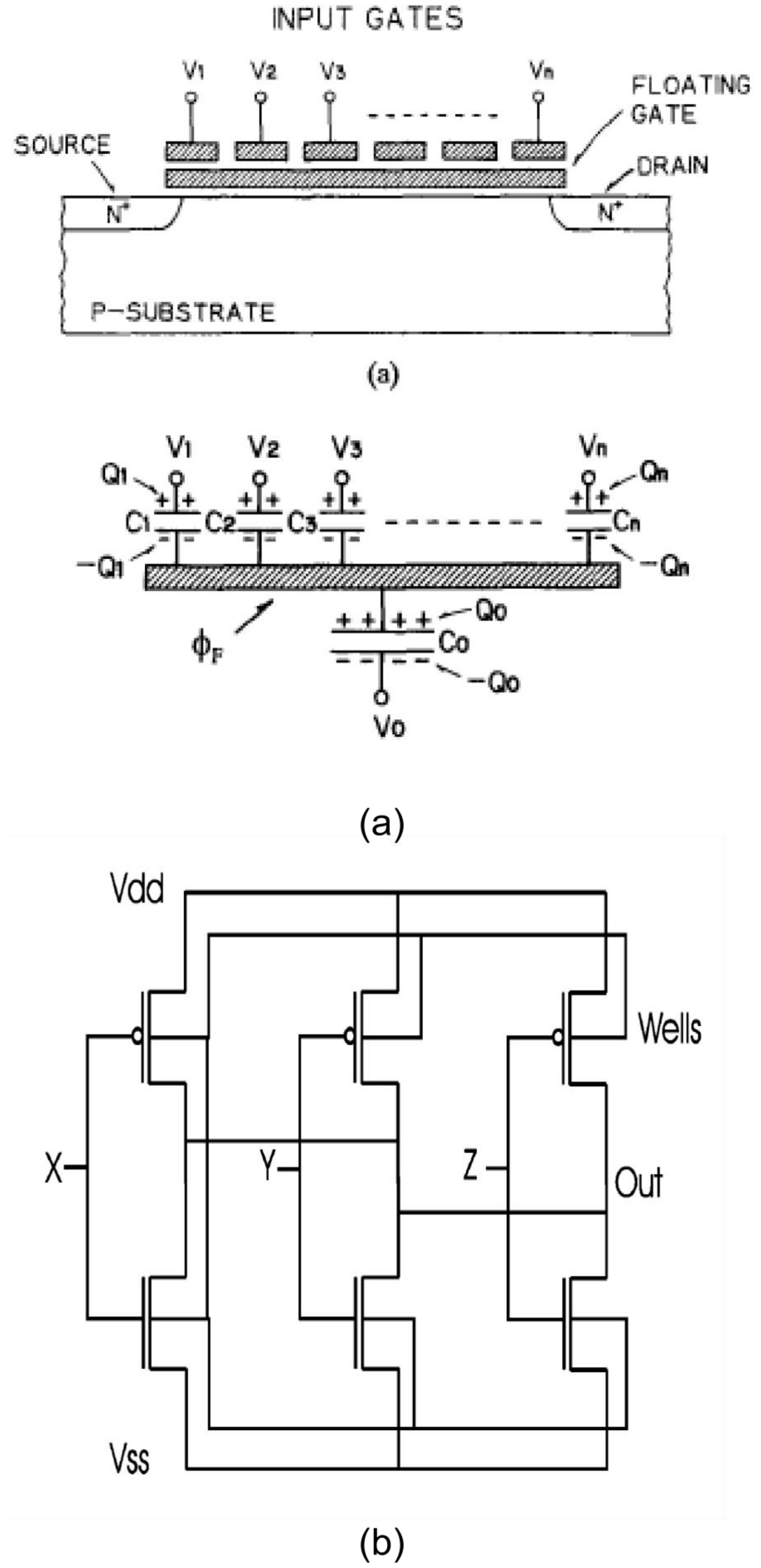
Fig. 1.13(a) presents a spiking neuron cell comprising of 5 transistor and 2 capacitors [123]. The neuron cell allows for the implementation of both inhibitory and excitatory spiking activity within the neuron. In addition to the aforementioned neuron models, several phenomenological neurons are also based upon the leaky integrate and fire model [36, 62, 124, 125], Fig. 1.13(b, c) and Fig. 1.14(a, b). In [36] Indiveri et al propose a I&F neuron constructed using various semiconductor devices to mimic several biological features in hardware including; using a source follower to control the spiking threshold voltage; a positive feedback inverter to aid in reducing the circuits power consumption. Refectory periods are also implemented using an inverter with controllable skew rate. These models [36, 62, 124, 125] are more complex than the aforementioned models containing a larger number of useful features, including; variable threshold voltages, refractory periods. Additionally the inclusion of variable output pulse duration, spiking frequency adaptation, controllable leakage paths and temporal summation of input signals make these models more suitable for implementation of VLSI neural networks [36, 62, 124, 125].



(a)

(b)

Fig 1.14 – Neuron Circuits – (a) [124], (b) [125]



(a)

(b)

Fig 1.15 – Neuron Circuits – (a) [126, 127] (b) [128]

Shiabata and Ohmi [126, 127] present a point neuron model constructed using a n-channel floating gate MOSFET, Fig. 1.15(a). The neuMOSFET performs temporal summation of all input signals at the gate level where the potential of the floating gate is given by equation 1.10. If the potential of the floating gate exceeds the threshold voltage of the device (as seen from the floating gate) an output is produced.

$$\phi_F = \frac{C_1V_1+C_2V_2+C_3V_3+\cdots C_iV_i}{C_{TOT}} \qquad (1.10)$$

$$C_{TOT} = \sum_{n=0}^{n=i} C_n \qquad (1.11)$$

Like the neuMOSFET, Wong et al also propose a floating gate neuron device which produces an output based upon the temporal summation of input signals. The circuits proposed by Aunet et al [128], Shiabta et al [126, 127] and Wong et al [129] all exhibit low power consumption. In [128] a perceptron is presented which achieves low power dissipation by utilising subthreshold MOSFETs, Fig. 1.15(b).

Each of the neuron circuits presented here all contain various limitations when considering their ability to create large networks of inter-connected neurons. There is a fundamental design trade-off between biological accuracy and the complexity in their design to emulate these features in hardware. Circuits which are biologically accurate tend to be complex, requiring a large area of silicon and in some cases have a large power consumption [36, 62, 124, 125]. While devices which emulate basic, minimal, features are simpler, smaller, circuits with low power consumption [126-129].

1.2.2 Silicon Synapse

Synapses are junctions between neurons which allow a neuron to transmit electrical or chemical signals to another cell. Synapses have the ability to adapt and change based upon internal and external inputs. These changes contribute to development of both long- and short- term memory aiding in learning within the network. It is believed that within the human nervous system each neuron can have up to 10^4 synaptic connections [130].

Therefore any synapses which are realised in hardware can potentially account for a large portion of the silicon required to implement neural networks in hardware. Therefore consideration to both the functionality, size and power consumption when developing a synapse in silicon.

Several silicon synapses have been proposed which aim to replicate the functionality of a biological synapses in VLSI. Hasler et al. [131] propose that a silicon synapse should contain five main properties;

- Synaptic weight should be stored permanently in the absence of learning
- Synaptic output should be a product of the neuronal input signal and synaptic weight
- Have a minimal area
- Have a low power dissipation
- Capable of Hebbian or back-propagation learning such that the synaptic weight can be modified

In Fig. 1.16(a) a compact silicon synapse is proposed, which is activated by an active-low input spike [52]. While this synapse has been used in a variety of VLSI neural networks, it does not integrate input spikes into a continuous output current [63, 135, 134]. This means that it cannot differentiate between input spikes of the same mean firing rate, but with different spike timing distributions. A CMOS based synapse is presented in Fig.1.16(b) [136, 137, 144, 145]. The synapse is an integrator synapse which consists of a weight transistor, M3, a spike input transistor, M4, and a current mirror, M1 and M2. The output from the current mirror is a mean output current which increases with repetitive input spikes.

A charge transfer synapse, CTS is presented in Fig.1.16(c) [138]. The synapse comprises of a MOST transistor operating in subthreshold with two MOST capacitors in close proximity. One capacitor is biased into strong inversion, such that the density of charge underneath the gate represents the synaptic weight. When a presynaptic spike is applied to the second MOS capacitor the charge density in the well falls and produces a current spike at the output. The amplitude of the output spike is related to the charge density in the well, controlled by the associated gate voltage. Spikes from an array of CTSs are aggregated using a current mirror.

The transistor serves to restore the charge in the well, the rate at which this occurs is determined by its associated gate voltage and is akin to the refractory period. An advantage of this synapse is that it is compact and has a low power consumption due to its ability to operate in transient mode.

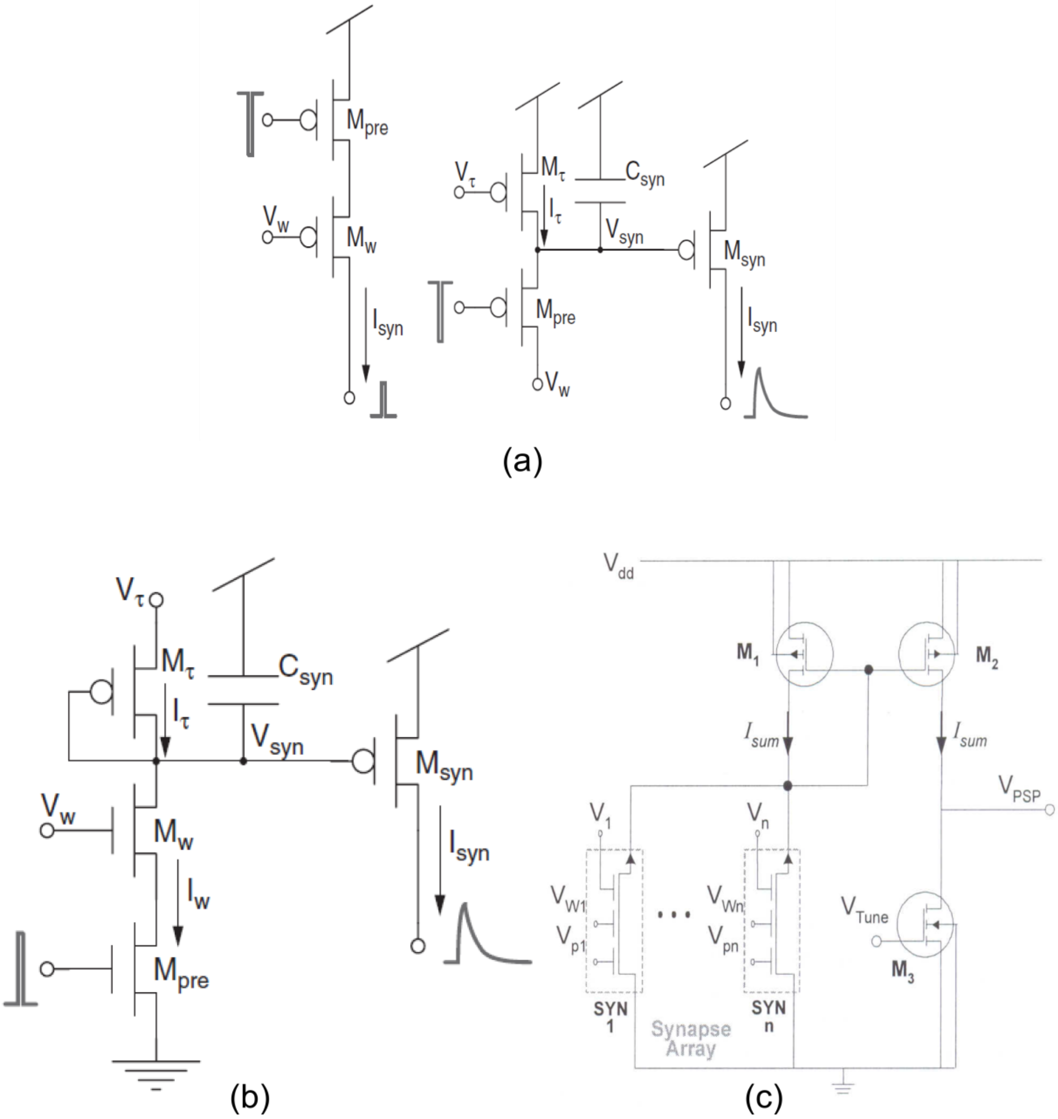


Fig 1.16 – Synapse Circuits – (a) [52], (b) [136, 137, 144, 145], (c) [138]

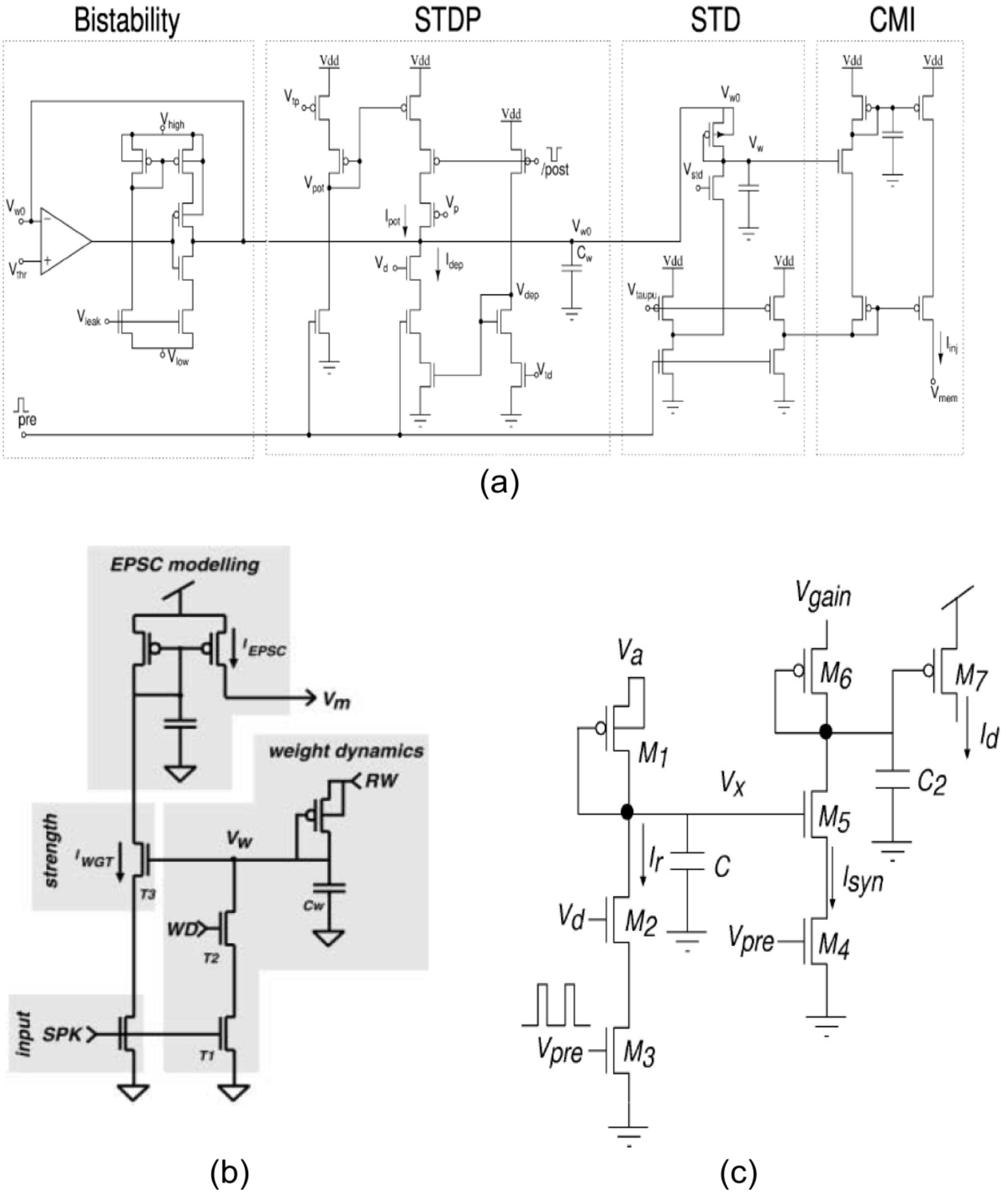
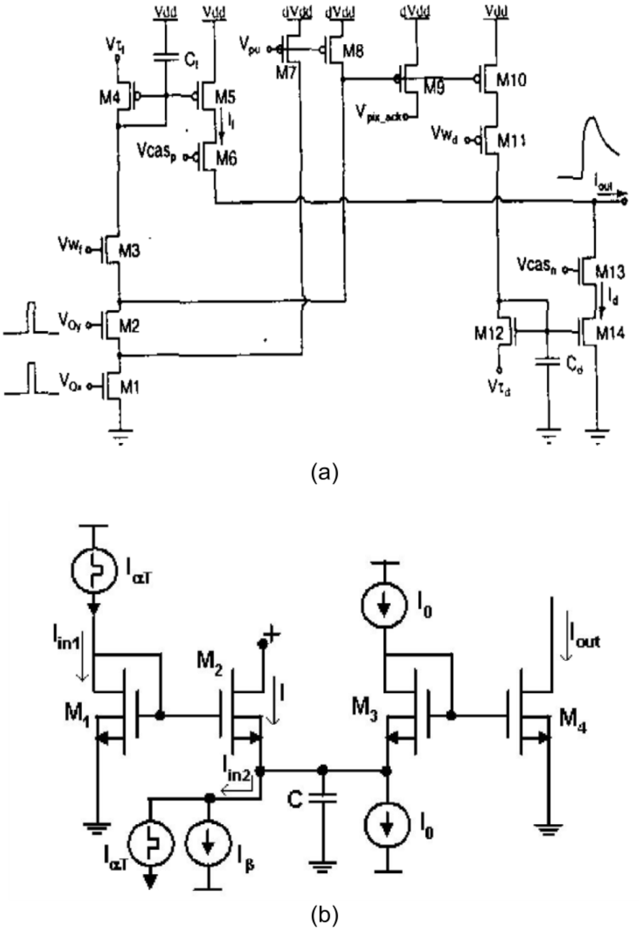


Fig 1.17 – Synapse Circuits – (a) [36, 139], (b) [140], (c) [141]

Fig.1.17(a) presents a bistable synapse proposed by Indiverdi et al. [36, 139]. The proposed synapses can implement several complex biological features such as short- and long-term synaptic plasticity. The synapse implements long-term plasticity through the STDP and bistability blocks, while short term plasticity is handled by the STD block, Fig.1.17(a). While the bistable synapse can implement this complex biological behaviour it requires a large amount of silicon to be implemented and is not suitable for larger neural networks. Additionally the synaptic weight is stored locally on a capacitor, no non-volatile memory, NVM, is included for weight storage. Other synapse circuits have been proposed which implement synaptic depression, Fig.1.17(b) [140] and (c) [141]. In [140] the synapse circuit presented implements short-term synaptic depression via computational features such as the 1/f law, and detection of long intervals of presynaptic silence. The circuit comprises of seven transistors and two capacitors. In [141] short term synaptic depression is again implemented using a simplistic seven transistors and two capacitor circuit.

1.2.2.1 Floating Gate Synapses

Floating gate synapses have also been implemented, where the synaptic weight is stored locally as charge, Fig.1.18(a) [142] and Fig.1.18(b)[143]. Charge can then be updated by either hot electron injection or Fowler-Nordheim tunnelling, the resulting weight charge can then be used to manipulate the output response of the synapse. For a large synaptic weight the synapse has a much greater effect on the associated neuron, conversely a synapse with a small synaptic weight has a reduced effect.



(a)

(b)

Fig 1.18 – Synapse Circuits – (a) [142], (b) [143]

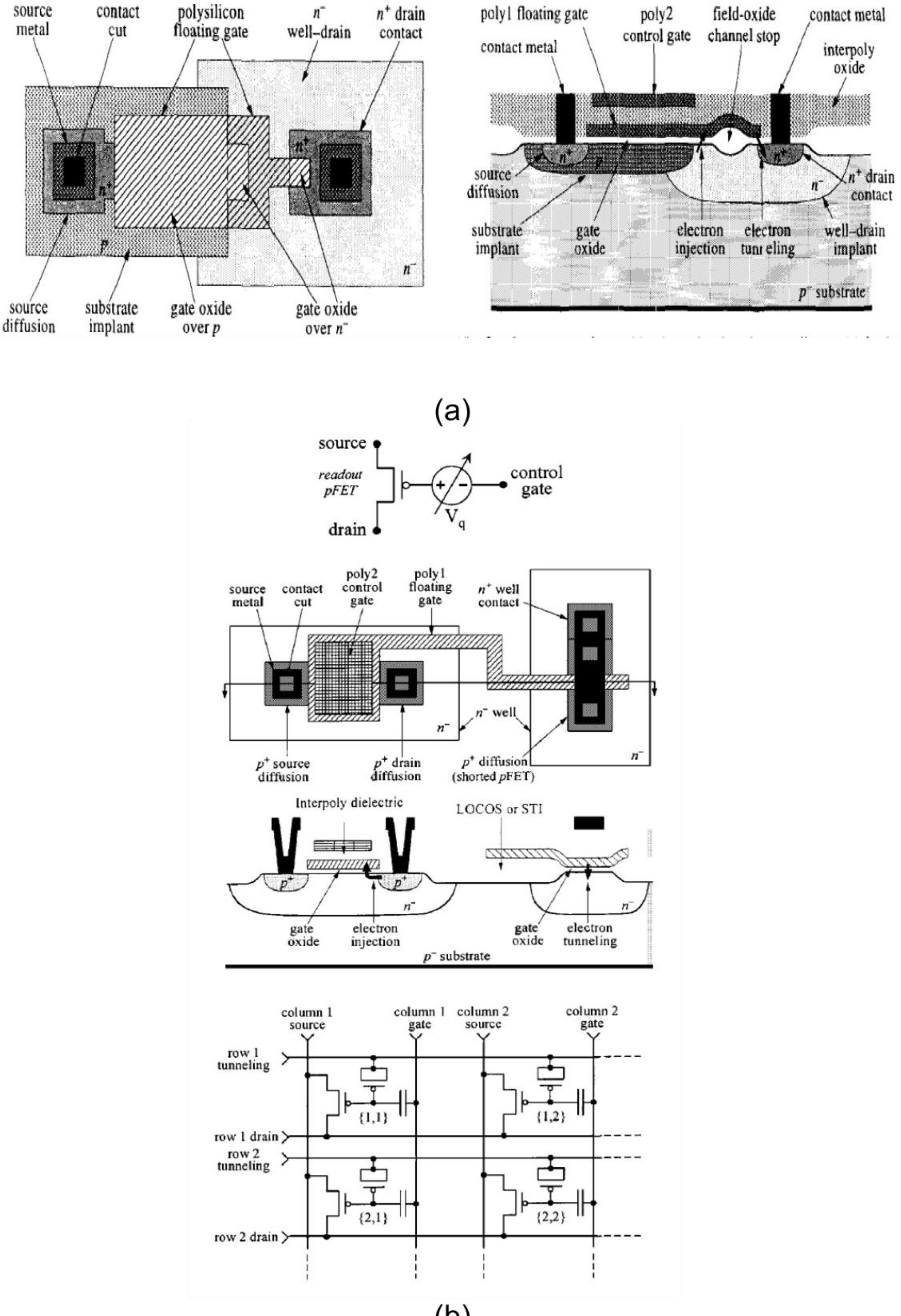


Fig 1.19 – Synapse Circuits – (a) [131, 132], (b) [133]

Hasler et al [131, 132] present a single transistor learning synapse which uses a floating gate to store the associated synaptic weight, Fig. 1.19(a). The proposed synapse simultaneously performs long term synaptic weight storage as well as computation of the product of the input signal and floating gate value. The product of input signal and synaptic weight produces an output current to an associated neuron. Additionally the weight can also be updated via Hebbian or back propagation learning rules. Diorio et al propose an improved CMOS synapse, which includes the ability to implement local long-term adaptation, Fig. 1.19(b) [133]. The proposed synapses allows for self-tuning analogue circuits for use in mixed-signal system-on-chip implementations.

In Figs.1.16-1.19, it can be said that silicon synapses can be implemented in a variety of ways; CMOS based circuits [34-36, 52, 62, 68, 133, 136-139, 142, 143], and floating gate devices [37, 131, 132]. Additionally various biological features including; long-term [36, 100], short-term plasticity [140, 141], facilitation and depression [34, 142, 143] as well as learning and adaptation can also be emulated in hardware. However there still exists a trade-off between functionality and synapse size when realised in hardware.

1.3 Thesis Overview

Effective realisation of neural networks in hardware requires circuits which can capture the physical biological features such as elasticity, parallelism (interconnection of neurons), facilitation and depression, temporal summation, thresholding, and refractory periods. To fully implement these efficiently in silicon there is a pressing need for small geometry, low power and biological plausible synapse and neuron circuits.

Two of the key features of a synapse is its ability to learn and modify its weight. In biology there are several different methods which govern the charge in synaptic weight, and these need to be captured and emulated in hardware neural networks. One of the key synaptic plasticity rules is spike-timing-dependent plasticity, STDP. This governs the long-term memory of a synapse which aids in learning. For this to be implemented in hardware a compact and low-power circuit needs to be designed.

The key contributions of the work presented in this thesis are;

- A review of the implementation of STDP in hardware neural networks, HNN. This review assess the critical biological features which are required and identifies the main trade-offs between biologically accuracy and circuit complexity which need to be considered.
- A floating gate, FG, device is presented which is outlined which can be integrated with silicon synapses to provide local, long-term storage of synaptic weight. The synaptic weight is represented as charge stored on the FG. This thesis also presents a model of the charge storage characteristics of the FG device to assess the potential range of synaptic weights possible.
- A compact circuit which emulates a biologically plausible version of STDP in HNN is presented. The circuit has been designed such that it is to operate in-conjunction with the FG device presented in this thesis. The circuit captures the basic notion that pre-post spiking cause an increase in the synaptic weight, while post-pre spiking cause a decrease in the synaptic weight. In both cases the time difference between spikes determines the magnitude of the synaptic weight change.

The remainder of this thesis is organized as follows; Chapter 2 presents an overview and review of the implementation of spike-timing-dependent plasticity, STDP, in hardware. An overview of the relevant semiconductor physics is given in Chapter 3. A model and experimental results of a floating gate device fabricated in a standard CMOS process is presented in Chapter 4. The floating gate device is to be integrated with a charge transfer synapse to provide long term synaptic weight storage [8, 138]. A compact STDP circuit for use with floating gate synapse is presented in Chapter 5 together with simulation and experimental results to demonstrate its operation. In Chapter 6 an overview of limited precision weights, LPW, and the advantages for use in analogue hardware neural networks is given. Floating gate devices are limited in the amount of charge which can be stored. Since the charge represents the associated synaptic weight, there will be limited number of weights which can be implemented in hardware. LPW allows a network to be trained to provide a useful function when the number of weights is limited. A summary of the thesis and future work are given in Chapter 7.

1.4 References

[1] R. G. Arns, “The other transistor: Early history of the metal-oxide semiconductor field-effect transistor,” Engineering Science and Educational Journal , vol. 7, no. 5, pp. 233-240, 1998.

[2] G. Moore, “Cramming more components onto integrated circuits.” Proceedings of the IEEE, vol. 86, no. 1, pp. 82-85, 1998.

[3] “Introduction to Intel’s 32nm Process Technology” Available: http://www.intel.com/content/www/us/en/pdf-pages/32nm-process-technology-paper.html

[4] “ITRS Roadmap 2011” Available: http://www.itrs.net

[5] D. Purves, G. J. Augustine, D. Fitzpatrick,. L. C. Katz, A. LaMantina, J. O. McNamara and S. M. Willians, Neuroscience, 2nd Edition, Sinauer Associates Inc, 2001

[6] K. Mehrotra, Elements of Artifical Neural Networks, MIT Press, 1997.

[7] D. H. Goldberg, G. Cauwenberghs and A. G. Andreou, “Probabilistic synaptic weighting in a reconfigurable network of VLSI integrate-and-fire neurons”, Neural Networks, vol. 14, pp. 781-793, 2001

[8] E. N. Marieb and K. Hoehn, Human Anatomy and Physiology 8th Edition, Pearson Benjamin Cummings, 2010

[9] T. Dowrick, Biologically Motivated Circuits For Third Generation Neural Networks, PhD Thesis, 2011

[10] G.Q. Bi and M.M Poo, “Synaptic modification in cultured hipocampl neurons: Dependence on spike timing, synaptic strength and postsynaptic cell type,” J. Neuroscience, vol. 18, pp. 10462-10472, 1993

[11] M.Nishiyama, K. Hong, K. Mikoshiba, M.M. Poo and K. Kato, “ Calcium stores regulate the polarity and input specificity of synaptic modification,” Nature, vol. 408, pp. 584-588, 2000.

[12] I. B. Levitand and L. K. Kaczmarek, The Neuron – Cell and Molecular Biology, 3rd Edition, Oxford University Press, 2002

[13] N. Rebola, B. N. Srikumar and C. Mulle, “Activity-dependent synaptic plasticity of NDMA receptors”, J. Physiol, vol. 588, no. 1, pp. 93-99, 2010.

[14] G. C. Castellani, E. M. Quinlan, L. N. Cooper, and H. Z. Shouval, “A biophysical model of bidirectional synaptic plasticity: Dependence on AMPA and NMDA receptors”, PNAS, vol. 98, no. 22, pp. 12772-12777, 2001.

[15] G. C. Castellani, E. M. Quinlan, F. Bersani, L. N. Cooper, and H. Z. Shouval, "A model of bidirectional synaptic plasticity: from signaling network to channel conductance", Learning & Memory, vol. 12, pp. 423-432, 2005.

[16] H. Z. Shouval, M. F. Beat, and L. N. Cooper, "A unified model of NMDA receptor-dependent bidirectional synaptic plasticity", PNAS, vol. 99, no. 16, pp. 10831-10836, 2002.

[17] J. Mellor, "Synaptic plasticity at hippocampal synapses", Springer Series in Computational Neuroscience, vol. 5, Part I, pp. 163-18, 2010.

[18] R. C. Malenka and M. F. Bear, "LTP and LTD: An embarrassment of riches", Neuron, vol. 44, pp. 5-21, 2004.

[19] D. J. Linden, "The return of the spike: Postsynaptic action potentials and the induction of LTP and LTD", Neuron, vol. 22, pp. 661-666, 1999.

[20] D.O. Hebb. The Organisztion of Behaviour. Wiley 1949

[21] W.B. Levy and O. Steward, "Temporal contiguity requirements for long-term associative potentiation/depression in the hippocampus," Neurosience, vol. 8, no. 4, pp. 791-797, 1983.

[22] G.Q. Bi and M.M Poo, "Synaptic modification in cultured hipocampl neurons: Dependence on spike timing, synaptic strength and postsynaptic cell type," J. Neuroscience, vol. 18, pp. 10462-10472, 1993.

[23] M.Nishiyama, K. Hong, K. Mikoshiba, M.M. Poo and K. Kato, " Calcium stores regulate the polarity and input specificity of synaptic modification," Nature, vol. 408, pp. 584-588, 2000.

[24] M. Tsukada, T. Aihara, Y. Kobayashi and H. Shimazaki, " Spatial analysis of spike-timing-dependent ltp and ltd in the ca1 area of hipocample slices using optical imaging," Hippocampus, vol. 15, no. 1, pp. 104-109, 2005.

[25] H. Tanaka, T. Morie, and K. Aihara, "A CMOS spiking neural network with symmetric/asymmetric STDP function," IEICE Transcations on Fundamentals, vol E92-A, no. 7, pp. 1690-1698, 2009.

[26] G.Q. Bi and M.M Poo, "Synaptic modification of corrolated activity: Hebbs postulate revisited," Annu. Rev. Neurosci, vol. 24, pp. 139-166, 2001

[27] N. Caporale and Y. Dan, "Spike timing-dependent plasticity: A Hebbian learning rule," Annu. Rev. Neurosci, vol. 31, pp. 25-46, 2008

[28] L. F. Abbott and S. B. Nelson, "Synaptic plasticity: taming the beast," Nature Neuroscience Supplment, vol 3, pp. 1178-1183, 2000.

[29] W. Maass, “Networks of spiking neurons: The third generation of neural network models.” Neural Networks, vol. 10, no. 9, pp. 1659-1671, 1997.

[30] W. Maass and C. M. Bishop. Pulsed Neural Networks. Cambridge, MA: MIT Press, 1999.

[31] S.J. Thorpe, A. Delorme and R. Vanrullen, “Spike based strategies for rapid processing.” Neural Networks,vol. 14, no. 6-7, pp. 715-725, 2001.

[32] J. Vreeken, “Spiking neural networks, an introduction”, Technical Report UU-CS-2003-008, Institute for Information and Computing Sciences, Universiteit Utrecht, 2002.

[33] A.L. Hodgkin and A.F. Huxley, “A quantitative description of ion currents and its application to conduction and excitation in nerve membranes,” Journal of Physiology, vol. 117, pp. 500–544.

[34] L. Shih-Chii and R. Douglas, “Temporal coding in a silicon network of integrate-andfire neurons,” IEEE Transactions on Neural Networks, vol. 15, pp. 1305-1314, 2004.

[35] S.-C. Liu, J. Kramer, G. Indiveri, T. Delbruck, T. Burg, and R. Douglas, “Orientationselective aVLSI spiking neurons,” Neural Networks, vol. 14, pp. 629-643, 2001.

[36] G. Indiveri, E. Chicca, and R. Douglas, “A VLSI array of low-power spiking neurons and bistable synapses with spike-timing dependent plasticity,” IEEE Transactions on Neural Networks, vol. 17, pp. 211-221, 2006.

[37] C. Diorio, D. Hsu, and M. Figueroa, “Adaptive CMOS: from biological inspiration to systems-on-a-chip,” Proceedings of the IEEE, vol. 90, pp. 345-357, 2002.

[38] R. Brette, M. Rudolph, T. Carnevale, M. Hines, D. Beeman, J. M. Bower, M. Diesmann, A. Morrison, P. H. Goodman, F. C. Harris Jr, M. Zirpe, T. NatschlÃüger, D. Pecevski, B. Ermentrout, M. Djurfeldt, A. Lansner, O. Rochel, T. Vieville, E. Muller, A. P. Davison, S. El Boustani, and A. Destexhe, “Simulation of networks of spiking neurons: A review of tools and strategies,” Journal of Computational Neuroscience, vol. 23, pp. 349-398, 2007.

[39] M. Migliore, C. Cannia, W. W. Lytton, H. Markram, and M. L. Hines, “Parallel network simulations with NEURON,” Journal of Computational Neuroscience, vol. 21, pp. 119-129, 2006.

[40] A. Delorme, J. Gautrais, R. van Rullen, and S. Thorpe, “SpikeNET: A simulator for modeling large networks of integrate and fire neurons,” Neurocomputing, vol. 26-27,pp. 989-996, 1999.

[41] M. Schaefer, T. Schoenauer, C. Wolff, G. Hartmann, H. Klar, and U. Ruckert, “Simulation of spiking neural networks - Architectures and implementations,” Neurocomputing, vol. 48, pp. 647-679, 2002.

[42] L.F. Abbott, 1999. “Lapique's introduction of the integrate-and-fire model neuron (1907).” Brain Research Bulletin, 50(5-6), pp. 303-304

[43] Y. Chen, 2008. “Low power small geometry building blocks for neural networks based on charge transfer devices” PhD Thesis, University of Liverpool, UK

[44] D. Frester, and N. Spruston, 1995. “Cracking the neuronal code.” Science, vol. 270, no. 5237, pp. 756-757

[45] Available: http://www.nature.com/nrn/journal/v6/n5/glossary/nrn1668_glossary.html

[46] C. Grassmann and J. K. Anlauf, "Fast digital simulation of spiking neural networks and neuromorphic integration with SPIKELAB," International journal of neural systems, vol. 9, pp. 473-478, 1999.

[47] M. S. Bakir, T. K. Gaylord, O. O. Ogunsola, E. N. Glytsis, and J. D. Meindl, "Optical transmission of polymer pillars for chip I/O optical interconnections," Photonics Technology Letters, IEEE, vol. 16, pp. 117-119, 2004.

[48] B. A. Floyd, H. Chih-Ming, and K. K. O, "Intra-chip wireless interconnect for clock distribution implemented with integrated antennas, receivers, and transmitters," Solid- State Circuits, IEEE Journal of, vol. 37, pp. 543-552, 2002.

[49] W. J. Dally and B. Towles, "Route packets, not wires: On-chip interconnection networks," in Proceedings - Design Automation Conference, Las Vegas, NV, 2001,pp. 684-689.

[50] J. Harkin, L. McDaid, S. Hall, B. McGinley, and S. Cawley, "A Reconfigurable and Biologically Inspired Paradigm for Computation Using Network- On-Chip and Spiking Neural Networks," International Journal of Reconfigurable Computing, 2009.

[51] D. D. Coon and A. G. U. Perera, “Integrate-and-fire coding and Hodgkin-Huxley circuits employing silicon diodes,” Neural Networks, vol. 2, no. 2, pp. 143–151, 1989.

[52] C. Mead, Analog VLSI and Neural Systems. Reading, MA: Addison-Wesley, 1989

[53] C. Mead, “Neuromorphic electronic systems,” Proceedings of the IEEE, vol. 78, no. 10, pp. 1629–1636, 1990.

[54] M. Mahowald and C. Mead, “The silicon retina,” Scientific American, vol. 264,

no. 5, pp. 76–82, 1991.

[55] M. Mahowald and R. Douglas, “A silicon neuron,” Nature, vol. 354, pp. 515–518, 1991.

[56] C. Song and K. P. Roenker, “Novel heterostructure device for electronic pulse-mode neural circuits,” IEEE Transactions on Neural Networks, vol. 5, no. 4, pp. 663–665, 1994.

[57] R. Douglas, M. Mahowald, and C. Mead, “Neuromorphic analogue VLSI,” Annual Review of Neuroscience, vol. 18, pp. 255–281, 1995.

[58] C. Diorio and R. P. N. Rao, “Neural circuits in silicon,” Nature, vol. 405, no. 6789, pp. 891–892, 2000.

[59] R. H. R. Hahnloser, R. Sarpeshkar, M. A. Mahowald, et. al., “Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit,” Nature, vol. 405, no. 6789, pp. 947–951, 2000.

[60] G. Indiveri, “A neuromorphic VLSI device for implementing 2-D selective attention systems,” IEEE Transactions on Neural Networks, vol. 12, no. 6, pp. 1455–1463, 2001.

[61] C. Diorio, D. Hsu, and M. Figueroa, “Adaptive CMOS: from biological inspiration to systems-on-a-chip,” Proceedings of the IEEE, vol. 90, no. 3, pp. 345–357, 2002.

[62] E. Chicca, D. Badoni, V. Dante, et al., “A VLSI recurrent network of integrate-and-fire neurons connected by plastic synapses with long-term memory,” IEEE Transactions on Neural Networks, vol. 14, no. 5, pp. 1297–1307, 2003.

[63] K. A. Zaghloul and K. Boahen, “Optic nerve signals in a neuromorphic chip I: Outer and inner retina models,” IEEE Trans. Biomed. Eng., vol. 51, no. 4, pp. 657–666, Apr. 2004.

[64] A. Bofill-i-Petit and A. F. Murray, “Synchrony detection and amplification by silicon neurons with STDP synapses,” IEEE Transactions on Neural Networks, vol. 15, no. 5, pp. 1296–1304, 2004.

[65] S. Liu and R. Douglas, “Temporal coding in a silicon network of integrate-and-fire neurons,” IEEE Transactions on Neural Networks, vol. 15, no. 5, pp. 1305–1314, 2004.

[66] R. J. Vogelstein, U. Mallik, J. T. Vogelstein, and G. Cauwenberghs, “Dynamically reconfigurable silicon array of spiking neurons with conductance-based synapses,” IEEE Trans. Neural Networks, vol. 18, no. 1, pp. 253–265, 2007.

[67] K. A. Boahen, “Point-to-point connectivity between neuromorphic chips using address-events,” IEEE Trans. Circuits Syst. II: Analog Digit. Signal Process., vol.

47, no. 5, pp. 416–434, May 2000.

[68] C. Rasche, “Neuromorphic excitable maps for visual processing,” IEEE Trans. Neural Networks., vol. 18, no. 2, pp. 520–529, 2007.

[69] P. A. Merolla, J. V. Arthur, B. E. Shi, and K. A. Boahen, “Expandable networks for neuromorphic chips,” IEEE Trans. Circuits Syst. I: Reg. Papers, vol. 54, no. 2, pp. 301–311, 2007.

[70] B. Hille, Ionic Channels of Excitable Membranes. Sunderland, MA: Sinauer, 1992.

[71] C. Koch, “Computation and the single neuron,” Nature, vol. 358, no. 6613, pp. 207–210, 1997.

[72] T. Delbruck, “Silicon retina with correlation-based velocity-tuned pixels,” IEEE Trans. Neural Networks, vol. 4, no. 3, pp. 529–541, 1993.

[73] C. Koch and B. Mathur, “Neuromorphic vision chips,” IEEE Spectrum, vol. 33, pp. 38–64, 1996.

[74] K. Boahen, “Retinomorphic vision system,” in Proceedings of International Conference on Microelectronics for Neural Networks (MicroNeuro), Lausanne, 1996, pp. 2–14.

[75] Z. K. Kalayjian and A. G. Andreou, “A silicon retina for polarization contrast vision,” in Proc. Int. Joint Conf. on Neural Networks, Washington, 1999, vol. 4, pp. 2329–2332.

[76] K. A. Zaghloul and K. Boahen, “Optic nerve signals in a neuromorphic chip II: Testing and results,” IEEE Trans. Biomed. Eng., vol. 51, no. 4, pp. 667–675, Apr. 2004.

[77] R. F. Lyon and C. A. Mead, “An analog electronic cochlea,” IEEE Trans. Acoust., Speech, Signal Process., vol. 36, no. 7, pp.1119–1134, Jul. 1988.

[78] E. Fragniere, A. van Schaik, and E. Vittoz, “Design of an analogue VLSI model of an active Cochlea,” Analog Integrated Circuits and Signal Processing, vol. 13, no. 1-2, pp. 19–35, 1997.

[79] T. Horiuchi and K. M. Hynna, “A VLSI-based model of azimuthal echolocation in the big brown bat,” Auton. Robots, vol. 11, no. 3, pp. 241–247, 2001.

[80] R. Sarpeshkar, J. Kramer, G. Indiveri, and C. Koch, “Analog VLSI architectures for motion processing: From fundamental limits to system applications,” Proc. IEEE, vol. 84, pp. 969–987, 1996.

[81] T. K. Horiuchi and C. Koch, “Analog VLSI-based modeling of the primate

oculomotor system," Neural Comput., vol. 11, no. 1, pp. 243–265, 1999.

[82] R. R. Harrison and C. Koch, "A robust analog VLSI motion sensor based on the visual system of the fly," Auton. Robots, vol. 7, no. 3, pp. 211–224, 1999.

[83] R. R. Harrison, "A biologically inspired analog IC for visual collision detection," IEEE Trans. Circuits Syst. I: Regular Papers, vol. 52, no. 11, pp. 2308–2318, 2005.

[84] R. Sarpeshkar, R. F. Lyon, and C. Mead, "A Low-Power Wide-Dynamic-Range Analog VLSI Cochlea," Analog Integrated Circuits and Signal Processing, vol. 16,pp. 245-274, 1998.

[85] B. Wen and K. Boahen, "A silicon cochlea with active coupling," IEEE Transactions on Biomedical Circuits and Systems, vol. 3, pp. 444-455, 2009.

[86] R. R. Harrison, "A biologically inspired analog IC for visual collision detection," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 52, pp. 2308-2318, 2005.

[87] C. Rasche, The Making of a Neuromorphic Visual System. New York: Springer-Verlag, 2005.

[88] J. C. Principe, V. G. Tavares, J. G. Harris, and W. J. Freeman, "Design and implementation of a biological realistic olfactory cortex in analog VLSI," Proc. IEEE, vol. 89, no. 7, pp. 1030–1051, 2001.

[89] T. J. Koickal, A. Hamilton, S. L. Tan, et. al., "Analog VLSI circuit implementation of an adaptive neuromorphic olfaction chip," IEEE Trans. Circuits Syst. I: Regular Papers, vol. 54, no. 1, pp. 60–73, 2007.

[90] G. Indiveri, R. Murer, and J. Kramer, "Active vision using an analog VLSI model of selective attention," IEEE Trans. Circuits Syst. II: Analog Digit. Signal Process., vol. 48, no. 5, pp. 492–500, 2001.

[91] R. J. Vogelstein, U. Mallik, E. Culurciello, G. Cauwenberghs, and R. Etienne-Cummings, "A multichip neuromorphic system for spike-based visual information processing," Neural Comput., vol. 19, no. 9, pp. 2281–2300, 2007.

[92] T. Y. W. Choi, P. A. Merolla, J. V. Arthur, K. A. Boahen, and B. E. Shi, "Neuromorphic implementation of orientation hypercolumns," IEEE Trans. Circuits Syst. I: Regular Papers, vol. 52, no. 6, pp. 1049–1060, 2005.

[93] J. Lazzaro, J.Wawrzynek, M. Mahowald, M. Sivilotti, and D. Gillespie, "Silicon auditory processors as computer peripherals," IEEE Trans. Neural Networks, vol. 4, no. 3, pp. 523–528, 1993.

[94] M. Mahowald, An Analog VLSI System for Stereoscopic Vision. Norwell, MA:

Kluwer, 1994.

[95] M. Sivilotti, “Wiring considerations in analog VLSI systems, with application to field-programmable networks,” Ph.D. dissertation, Dept. Comp. Sci., California Inst. Technol., Pasadena, 1991.

[96] K. A. Boahen, “Communicating neuronal ensembles between neuromorphic chips,” in Neuromorphic Systems Engineering, T. S. Lande, Ed. Norwell, MA: Kluwer, 1998, pp. 229–259.

[97] K. A. Boahen, “A burst-mode word-serial address-event link-I: Transmitter design,” IEEE Trans. Circuits Syst. I: Regular Papers, vol. 51, no. 7, pp. 1269–1280, 2004.

[98] K. A. Boahen, “A burst-mode word-serial address-event link-II: Receiver design,” IEEE Trans. Circuits Syst. I: Regular Papers, vol. 51, no. 7, pp. 1281–1291, 2004.

[99] K. A. Boahen, “A burst-mode word-serial address-event link-III: Analysis and test results,” IEEE Trans. Circuits Syst. I: Regular Papers, vol. 51, no. 7, pp. 1292–1300, 2004.

[100] S. Mitra, S. Fusi, and G. Indiveri, "A VLSI spike-driven dynamic synapse which learns only when necessary," ISCAS 2006.

[101] R. J. Vogelstein, U. Mallik, and G. Cauwenberghs, "Silicon spike-based synaptic array and address-event transceiver," 2004, pp. V-385-V-388 Vol.5.

[102] H. Markram, "The Blue Brain Project," Nature Reviews Neuroscience, vol. 7, pp. 153-160, 2006.

[103] X. Jin, A. Rast, F. Galluppi, M. Khan, and S. Furber, "Implementing learning on the SpiNNaker universal neural chip multiprocessor," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). vol. 5863 LNCS, 2009, pp. 425-432.

[104] A. D. Rast, Y. Shufan, M. Khan, and S. B. Furber, "Virtual synaptic interconnect using an asynchronous network-on-chip," in Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on, 2008, pp. 2727-2734.

[105] A. D. Rast, M. M. Khan, X. Jin, L. A. Plana, and S. B. Furber, "A universal abstract time platform for real-time neural networks," in Proceedings of the International Joint Conference on Neural Networks, 2009, pp. 2611-2618.

[106] A. D. Rast, S. Welbourne, X. Jin, and S. B. Furber, "Optimal connectivity in hardware-targetted MLP networks," in Proceedings of the International Joint Conference on Neural Networks, 2009, pp. 2619-2626.

[107] M. M. Khan, D. R. Lester, L. A. Plana, A. Rast, X. Jin, E. Painkras and S. B. Furber, “SpiNNaker: Mapping nerual networks onto a massively-parallel chip multiprocessor,” Internation Joint Conference on Neural Networks 2008, pp.2850-2857, 2008.

[108] X. Jin, A. Rast, G. Galluppi, S. Davies, and S. B. Furber, “Implementing spike-timing-dependent plasticity on SpiNNaker neuromorphic hardware,” World Congress on Computational Intelligence 2010, pp. 2302-2309, 2010

[109] M. E. Izhikevich, “Simple model of spiking neurons,” IEEE Transactions on Neural Networks, vol. 14, no. 6, pp. 1569-1572, 2003

[110] A. Daouzli, S. Saighi, L. Buhry, Y. Bornat, and S. Renaud, "Weights convergence and spikes correlation in an adaptive neural network implemented on VLSI," in BIOSIGNALS 2008 - Proceedings of the 1st International Conference on Bio-inspired Systems and Signal Processing, Funchal, Madeira, 2008, pp. 286-291.

[111] Q. Zou, Y. Bornat, S. Saghi, J. Tomas, S. Renaud, and A. Destexhe, "Analog-digital simulations of full conductance-based networks of spiking neurons with spike timing dependent plasticity," Network: Computation in Neural Systems, vol. 17, pp. 211-233, 2006.

[112] S. Renaud, J. Tomas, Y. Bornat, A. Daouzli, and S. Saighi, "Neuromimetic ICs with analog cores: An alternative for simulating spiking neural networks," in Proceedings - IEEE International Symposium on Circuits and Systems, New Orleans, LA, 2007, pp. 3355-3358.

[113] J. Tomas, Y. Bornat, S. Saghi, T. Lavi, and S. Renaud, "Design of a modular and mixed neuromimetic ASIC," in Proceedings of the IEEE International Conference on Electronics, Circuits, and Systems, Nice, 2006, pp. 946-949.

[114] S. R. N. Lewis, "Spiking neural networks “in silico”: from single neurons to large scale networks," in Fourth International Multi-Conference on Systems, Signals & Devices Hammamet, Tunisia, 2007.

[115] C. Rasche and R. Douglas, “An improved silicon neuron,” Analog Integrated Circuits and Signal Processing, vol. 23, pp. 227–236, 2000

[116] E. Farquhar and P. Hasler, “A bio-physically inspired silicon neuron,” IEEE Trans. Circuits Syst. I: Reg. Papers, vol. 52, no. 3, pp. 477–488, 2005.

[117] J. V. Arthur and K.. Boahen, “Synchrony in silicon: the gamma rhythm,” IEEE Transactions on Neural Networks, vol. 18, no. 6, pp. 1815–1825, 2007.

[118] D. Dupeyron, S. Le Masson, Y. Deval, G. Le Masson, and J. P. Dom, “A BiCMOS implementation of the Hodgkin-Huxley formalism,” in Proc. 5th Int.

Conf. Microelectronics for Neural Networks and Fuzzy Systems, pp. 311–316, 1996.

[119] I. S. Han, "Biologically Inspired Hardware Implementation of Neural Networks with Programmable Conductance," in International Joint Conference on Neural Networks, 2007.

[120] L. Chun, S. Bingxue, and C. Lu, "Hardware implementation of an expandable on-chip learning neural network with 8-neuron and 64-synapse," Proceedings. 2002 IEEE Region 10 Conference on Computers, Communications, Control and Power Engineering, pp. 1451-1454 vol.3, 2002

[121] J. H. B. Wijekoon and P. Dudek, "A Simple Analogue VLSI Circuit of a Cortical Neuron," in EEE International Conference on Electronics, Circuits and Systems, ICECS 2006.

[122] E. M. Izhikevich, "Simple model of spiking neurons," Neural Networks, IEEE Transactions on, vol. 14, pp. 1569-1572, 2003.

[123] Y. Ota and B. M. Wilamowski, "Analog implementation of pulse-coupled neura networks," IEEE Transactions on Neural Networks, vol. 10, pp. 539-544, 1999.

[124] S. Lui and R. Douglas, “Temporal coding in a silicon network of integrate-and-fire neurons,” IEEE Transactions on Neural Networks, vol. 15, no. 5, pp. 1305–1314, 2004.

[125] S. R. Schultz and, M. A. Jabri, “Analogue VLSI integrate-and-fire neuron with frequency adaptation,” Electron. Lett., vol. 31, pp. 1357–1358, 1995.

[126] T. Shibata, and T. Ohmi, “An intelligent MOS transistor featuring gate-level weighted sum and threshold operations.”, International Electron Devices Meeting, pp. 919-922, 1991.

[127] T. Shibata, and T. Ohmi., “A functional MOS transistor featuring gate-level weighted sum and threshold operations.” Electron Devices, vol. 39, no. 6, pp. 1444-1455, 1992.

[128] S. Aunet, B. Oelmann, P. A. Norseng, and Y. Berg, "Real-Time Reconfigurable Subthreshold CMOS Perceptron," IEEE Transactions on Neural Networks, vol. 19,pp. 645-657, 2008.

[129] Y. L. Wong, X. Peng, and P. Abshire, "Ultra-low Spike Rate Silicon Neuron," in Biomedical Circuits and Systems Conference, pp.95-98, 2007

[130] D. H. Goldberg, G. Cauwenberghs and A. G. Andreou, “Probabilistic synaptic weighting in a reconfigurable network of VLSI integrate-and-fire neurons”, Neural Networks, vol. 14, pp. 781-793, 2001.

[131] P. Hasler, C. Diorio, B. A. Minch, and C. Mead, “Single transistor learning synapses with long term storage,” IEEE Int. Symp. on Circuits and Systems, vol. 3, pp. 1660-1663, 1995

[132] C. Diorio, P. Hasler, B. A. Minch, and C. Mead, “A single-transistor silicon synapse,” IEEE Trans. Electron Devices, vol. 43, no.11, pp. 1972–1980, 1996

[133] C. Diorio, D. Hsu, and M. Figueroa, “Adaptive CMOS: from biological inspiration to systems-on-a-chip,” Proceedings of the IEEE, vol. 90, no. 3, pp. 345–357, 2002

[134] W. Maass and C. M. Bishop, Pulsed Neural Networks. Cambridge, MA: MIT Press, 1999.

[135] P. Merolla and K. Boahen, “A recurrent model of orientation maps with simple and complex cells,” in Advances in Neural Information Processing Systems, vol. 16, pp. 995–1002, MIT Press, 2003.

[136] K. Boahen, “Retinomorphic vision system,” in Proceedings of International Conference on Microelectronics for Neural Networks (MicroNeuro), Lausanne, 1996, pp. 2–14.

[137] K. A. Boahen, “Point-to-point connectivity between neuromorphic chips using address-events,” IEEE Trans. Circuits Syst. II: Analog Digit. Signal Process., vol. 47, no. 5, pp. 416–434, May 2000.

[138] Y. Chen, L. McDaid, S. Hall, and P. Kelly, “A programmable facilitating synapse device.”, 2008 International Joint Conference on Neural Networks, IJCNN 2008, 2008, Institute of Electrical and Electronics Engineers Inc pp1615-1620, 2008.

[139] G. Indiveri, “Neuromorphic bistable VLSI synapses with spike timing dependent plasticity,” in Advances in Neural Information Processing Systems, vol. 15, 2002.

[140] C. Rasche and R. Hahnloser, “Silicon synaptic depression,” Biological Cybernetics, vol. 84, no. 1, pp. 57–62, 2001.

[141] M. Boegerhausen, P. Sutter, and S. C. Liu, “Modeling short-term synaptic depression in silicon,” Neural Computation, vol. 15, no. 2, pp. 331–348, 2003

[142] E. Chicca, G. Indiveri, and R. Douglas, "An adaptive silicon synapse," Proceedings of the 2003 International Symposium on Circuits and Systems, pp. I-81 - I-84 vol.1, 2003

[143] E. Lazaridis, E. M. Drakakis, and M. Barahona, "A biomimetic CMOS synapse," IEEE International Symposium on Circuits and Systems, 2006. ISCAS 2006

[144] C. Bartolozzi and G. Indiveri, “Synaptic Dynamics in Analog VLSI,” Neural

Computation, vol. 19, no. 10, pp. 2581-2603, 2007.

[145] K. A. Boahen, Retinomorphic Vision Systems: Reverse Engineering the Vertebrate Retina, PhD Thesis, California Institute of Technology, 1997.

[146] W. Maass, ‘Motivation, theory and applications of liquid state machines’, Computability in Context: Computation and Logic in the Real World, Imperial College Press, 2011.

Chapter 2 – Implementing STDP in Hardware Neural Networks: A Review and Perspective

2.1 Introduction

In recent years there has been increasing interest in the design and implementation of devices and circuits which mimic biological neural network features. The ITRS roadmap for silicon indicates that alternatives to traditional computational circuits will be required since Moore's Law is nearing an end. Neural Networks in hardware are seen as one potential paradigm.

Neural networks are massively parallel computational systems constructed using neurons and synapses. Research has suggested that the number of neurons in the human brain is of the order of 10^{14}[1] each with up to 10^3 synaptic connections. The third generation of so-called neuromorphic engineering is based in Spiking Neural Networks, SNNs. It is believed that the behaviour of SNNs is more biologically plausible since a spiking neuron operates by using a train of spikes which incorporate spatial-temporal information.

Significant research has been conducted to develop very large-scale integrated circuit (VLSI) implementations of biologically inspired neural networks [25]. While these types of circuits can be a powerful resource for constructing neural networks, they often lack the ability to produce simple compact structures which can be used to build biological scale networks. This is mainly due to the additional complex circuitry required to emulate the functionality of a synapse [2,3]. Thus it is this inability to build biological scale networks and the notion that semiconductor devices possess some similarities to biological neurons and synapses which has led to additional research into single transistor type devices which can be used to implement synapses and neurons, [5-11]. In both cases in order for neural networks to be able to do anything useful, that is, learn and adapt to various internal and external influences, there must be some method to realize plasticity. Synaptic plasticity is described as the increase or decrease of the synaptic weight between a pre- and post-synaptic neuron within the neural network [12].

Hebb's theory [11] describes how the synaptic weight is allowed to change based upon the inputs and outputs of each neuron within the NN. Hebb states that; "*when an axon of cell A is near enough to excite a cell, B, and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency as one of the cells firing B, is increased*". Essentially it is stated in [11] that when the input from neuron A, meets or exceeds the threshold value of neuron B (and this is a repeated process) then the synaptic weight between neuron A and neuron B is increased. Conversely [11] also states that if A has little or no effect on B then the synaptic weight is either kept constant or reduced, depending upon additional factors.

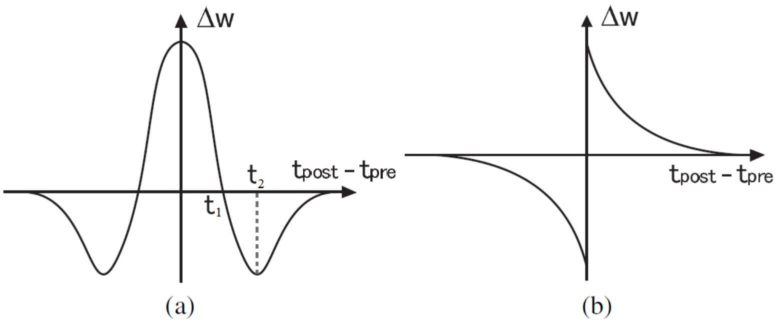


Figure 2.1 - (a) Symmetric STDP Curve. (b) Asymmetric STDP Curve

A further development of the Hebbian learning concept was the introduction of spike timing dependent plasticity, STDP, in 1983 [13]. STDP is concerned with updating, both increasing and decreasing the weight of a synapse based upon the relative timings of pre- and post-synaptic spikes. In biological neural networks, there are two main types of STDP, namely symmetric, Fig. 2.1(a), and asymmetric, Fig. 2.1(b), [14-17].

In symmetric STDP the synaptic weight is updated under the following conditions:

- If the time difference between the occurrence of the pre and post synaptic input, $\Delta t = t_{post} - t_{pre}$, is equal to 0, then the maximum positive weight update, Δw, occurs. This increases the stored synaptic weight.
- As $\Delta t \rightarrow t_1$, then Δw beings to decrease in magnitude, (but is still positive) until at Δt=t1, Δw=0. Again this still has the effect of increasing the stored synaptic weight. When Δt increase past t_1, Δw now becomes negative, thus beginning to decreasing the stored synaptic weight. As $\Delta t \rightarrow t_2$, Δw is such that it tends towards the maximum negative weight update, at $\Delta t = t_2$. Finally when $\Delta t > t_2$, $\Delta w \rightarrow 0$.

Therefore, it can be said that in symmetric STDP, the update of the synaptic weight is dependent upon Δt and not the order in which the pre and post spikes occur. Unlike symmetric STDP, asymmetric STDP updates the synaptic weight based upon the order in which the pre- and post-synaptic spikes occur and the time difference between them. Therefore the synaptic weight is updated under the following conditions;

the weight update, Δw is;

- Increased if a pre synaptic spike occurs prior to a post synaptic spike.
- Decreased if a post synaptic spike occurs prior to a pre synaptic spike.

The magnitude of Δw is determined by;

- $|\Delta t| \rightarrow 0$, $\Delta w \rightarrow |\Delta w_{max}|$
- $|\Delta t| \rightarrow \infty$, $\Delta w \rightarrow |\Delta w_{min}|$ $(\Delta w \rightarrow 0)$

Of the two STDP rules presented, it is the latter of these, namely asymmetric STDP, (hereafter referred to as STDP), shown in Fig 2.1(b), which is thought to occur more frequently in biological neural networks, [14], [18-20]. This method is seen as the chosen STDP rule to implement in hardware, as it is more biologically plausible. It is worth noting that the exponential functions shown in Fig. 2.1b) are not a pre-requisite for STDP but rather are a mathematical convenience. It is only necessary for the relative timings to produce reinforcement or reduction of the weights to be realized.

Once possible alternative to STDP in hardware could be to increase or decrease the synaptic weight by a constant fixed amount, ΔV_w. This would be based solely on the order in which the pre- and post-synaptic spikes occur. If the presynaptic spike occurred prior to the postsynaptic spike then the synaptic weight would be increased by ΔV_w. Similarly if the postsynaptic spike occurred prior to the presynaptic spike then the synaptic weight would be decreased by ΔV_w. This rule would remove the need to calculate Δt, therefore $\Delta V_w \propto 1/\Delta t$ does not occur. However additional research would be need to see if this rule is biologically plausible.

An alternative to STDP is the BCM, Bienenstock, Cooper and Munro, learning rule. This rule can be used to evoke LTP and LTD within a neural network. LTP and LTD are induced based upon the comparison of correlated pre and post-synaptic firing rates to a threshold value [59]. Specifically the change in synaptic weight is dependent upon the instantaneous pre and post- synaptic activity as well as a slow varying time-averaged value of post-synaptic activity. Additionally it is the threshold value which varies as a function of the post-synaptic activity in order to ensure that the model is stable. BCM like STDP is an improvement upon Hebb's theory in that it has the ability to decrease the effect that a pre-synaptic neuron has on a post-synaptic neuron by decreasing the synaptic weight between the two neurons.

Another potential learning rule could be implemented in hardware is the idea of synaptic scaling [20]. Like the BCM rule this attempts to address the instability in Hebbian learning by introducing a stabilizing mechanism which is based upon the firing rate of the post synaptic neuron [20]. Synaptic scaling works by adjusting the weight of the synapse at a particular neuron by either subtracting an arbitrary amount or by multiple of a specified synaptic efficacy. This is done such that competition between neurons is introduced.

The two alternative learning rules presented are both based upon Hebbian learning, and while they promote stability and a change in the synaptic weight, they are complex learning rules based upon the activity of pre, post-synaptic activity as well as prior activity of post-synaptic responses. Therefore these rules may require large complex circuitry to implement in hardware. It is proposed that a simplified form of Hebbian learning which can also decrease the synaptic weight should be implemented in hardware. One such rule uses subtractive normalization such that pre-post spiking introduces an increase in the synaptic weight between two neurons and repeated pre-post spiking subsequently cause further increase to this weight. However should a post-pre spiking event occur then all synaptic weights are decreased by a fixed amount [60].

The remainder of this chapter is organized as follows; in sections 2.2-2.5, an overview is presented of the reported methods of implementing synaptic plasticity in hardware neural networks, HNN, with specific emphasis on STDP Section 2.6 contains a discussion on the various methods presented with emphasis on the complexity and scalability of the circuits presented. In this paper scalability refers to the ability to use the STDP circuits in conjunction with hardware neurons and synapses to generate a neural network consisting of sufficient neurons and synapses to undertake useful tasks. Additionally suggestions are made as to how synaptic plasticity can be implemented in future HNNs. Conclusions are made in section 2.7.

2.2 STDP in Hardware – Symmetric Circuits

In this section a review of reported methods of implementing synaptic weight update using symmetric circuits, with particular reference to STDP in HNN is presented and explained.

In [21], [22] a VLSI model of a neural network which uses STDP to update the synaptic weights between two associated neurons is presented. The model contains an STDP circuit and this is shown in Fig. 2.2. This STDP circuit is used with every synapse within the network and is symmetric with respect to pre- and post-synaptic inputs. MOSFETs, M_1-M_4 are used to form a latch which can operate within two states; a delay measurement mode and result, and an accumulation mode. For the case when M_1 and M_3 are conducting, then M_6 will isolate the three MOSFETs connected in series, M_8-M_{10} from the supply. Transistor M7 connects capacitor C_1 to the supply via the current source M_{11}. M_{11} in turn keeps the gate voltage of M_{13} high. Additionally if both row and column resets are held at ground, C_2 will maintain its charge. Thus the voltage on C_2 represents ΔV_w, the change in synaptic weight since the last reset signal occurred.

If a pre-synaptic input occurs, M_2 and M_4 will begin to conduct and the latch is put into measurement mode. C_1 is thus isolated from the supply and M_{12} charges C_1 to an initial voltage, V_1. Transistors M_8-M_{10} are isolated to the positive supply by M_6 during the measurement mode. Since these MOSFETs operate in the sub-threshold regime, C_1 is discharged if any other pre-synaptic spikes occur, then C_1 is again pre-charged to V_1, thus resetting the measurement mode. When a post-synaptic spike occurs, the measurement mode is ended and the latch switches into the accumulation mode. C_1 is now discharged via M_7, this activating M_{13} for a period of time which is proportional to the remaining charge on C_1. This in turn removes an amount of charge on C_2, equivalent to $F(\Delta t)$ which can be thought of as ΔV_w.

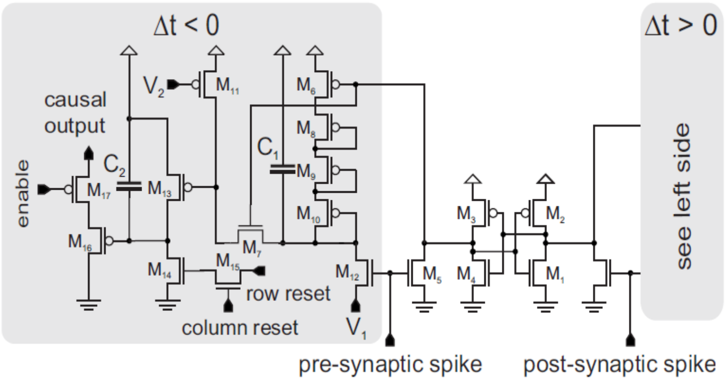


Figure 2.2 - STDP circuit [22]

It should be noted that this circuit is used in conjunction with a DAC and digital memory. The synaptic weights are stored in a digital form within the proposed neural network. The overall size of the STDP circuit presented in [22] is approximately 5x10μm^2 using 0.18μm process.

A VLSI spike-driven, dynamic synapse is presented in Fig. 2.3[23]. The synapse has the ability to update its weight based upon the occurrence of pre- and post-synaptic input spikes.

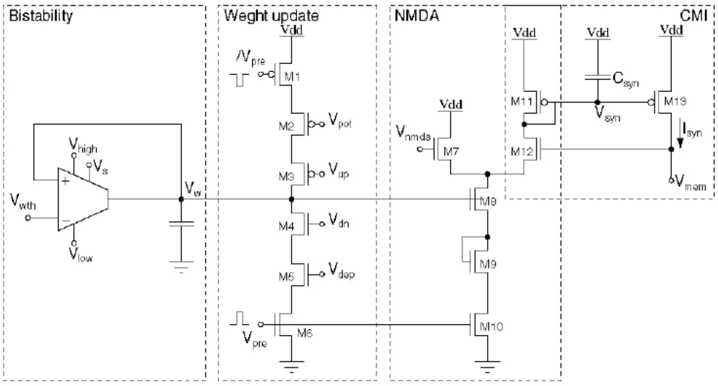
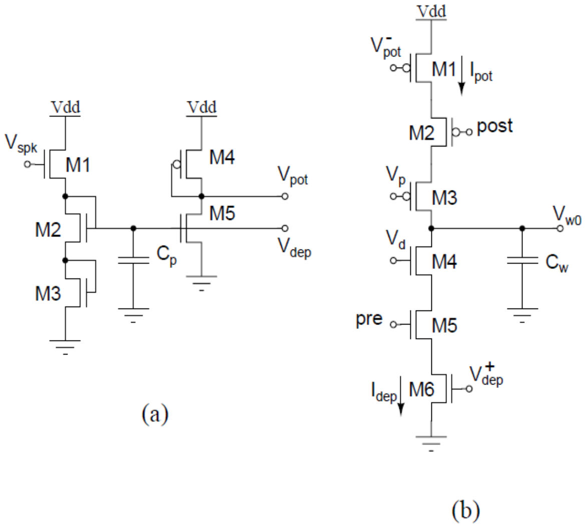


Figure 2.3 - Synapse circuit proposed by Mitra el al. in [23]

The weight update circuit receives both pre- and post-synaptic spikes and will update the stored synaptic weight, V_w, on each pre-synaptic spike. In order for the synaptic weight to be updated, V_{pot} must be low when the pre-synaptic spike occurs. This allows for the node V_w to receive a positive charge packet. The magnitude of this charge packet is dependent upon V_{up}. Conversely if V_{dep} is high during a pre-synaptic spike, then V_w is decreased through the application of a negative charge packet to V_w. In the event that V_{up} is high and

V_{dep} is low when a pre-synaptic spike occurs then the synaptic weight, V_w, is not modified. While the circuit presented has the capability to update the synaptic weight, it should be noted that the method used to update the synaptic weight is not strictly STDP but rather it is a variation on the STDP rules presented previously.

An improvement to the synaptic circuit presented in [23] is presented in [24]. The weight update circuit presented in Fig. 2.3 is updated such that it now takes into account the post-synaptic spike after V_{pot}, Fig. 2.4(b). The circuit blocks presented in Fig. 2.4 [24] serve to update the synaptic weight. In Fig. 2.4(a), the pulse-shaping circuit is used to generate the two signals V_{pot} and V_{dep}. These voltages are used in the synaptic weight update circuit of Fig. 2.4(b), when a fast digital pulse V_{spk} occurs due to the corresponding neuron emitting a spike. Voltages V_{pot} and V_{dep} both have a sharp onset and a logarithmic decay, and are used to realize the potentiation (increase in the synaptic weight) and depression (decrease in the synaptic weight), characteristics of the associated synapse.



(a)

(b)

Figure 2.4 – (a) Pulse shaping circuit. (b) Improved weight update circuit proposed in [24]

Voltage, V_{w0} represents the synaptic weight of the synapse and is updated according to the relative timings of the pre- and post-synaptic spikes. Transistors M1 and M6 operate in the sub-threshold regime, and are used to generate positive and negative currents which are proportional to V_{pot} and V_{dep} respectively. M2 and M5 act as switches which are only ‘on’ for the duration of the pre- and post-synaptic spikes. M3 and M4 are used to regulate the maximum amount of current which can be used to inject or remove charge on the capacitor C_w during the input spikes.

If a pre-synaptic spike is followed by a post-synaptic spike after a time Δt, then M1 generates a current, I_{pot}, which is proportional to Δt and increases the stored charge on C_w, as $Q_{inj}= I_{pot}\Delta t$. If a post-synaptic spike is followed by a presynaptic spike after a time Δt, M6 generates a current, I_{dep}, which is proportional to Δt and decreases the stored charge on C_w, as $Q_{inj} = I_{dep}\Delta t$. By varying V_p and V_d, various incarnations of the STDP curve can be generated by the circuit presented in [24], shown in Fig. 2.5(a). However by keeping both V_p and V_d set to constant values a typical STDP can be generated, as shown in Fig. 2.5(b).

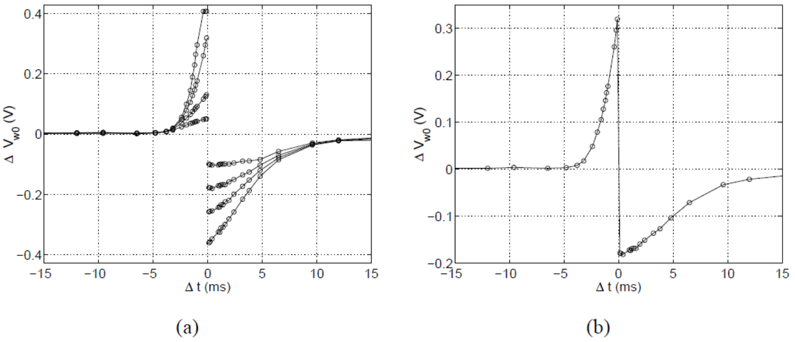


Figure 2.5 - (a) 4 STDP curves obtained by varying V_p and V_d. (b) STDP plot obtained for V_p=4.0V and V_d=0.6V [24]

A further improvement to the synaptic weight update circuit is presented in [25, 26], shown in Fig. 2.6. The circuit of Fig. 2.4(a) has been adapted and integrated with that of Fig. 2.4(b) such that the circuit shown in Fig. 2.6 is obtained. This new STDP circuit is now symmetric and can be used to increase or decrease V_{w0}.

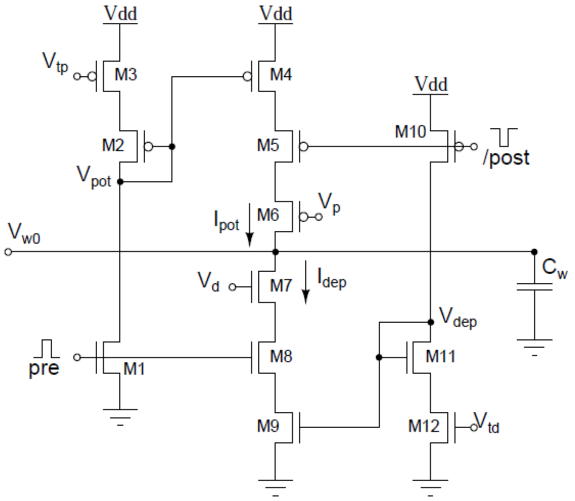


Figure 2.6 - STDP circuit proposed in [25,26]

The operation of the new STDP circuit is now considered. If a pre-synaptic spike occurs prior to a post-synaptic spike, then upon the arrival of the pre-synaptic spike V_{pot} is generated, with the decay set by V_{tp}. As in [24], V_{pot} is used to generate current I_{pot}, and as such I_{pot} will cause an increase in the charge stored on capacitor C_w, provided that the pre-synaptic spike is still occurring. Similarly if a post-synaptic spike occurs first, the arrival of the post-synaptic spike causes V_{dep} to be generated. The decay of V_{dep} set by V_{td}. V_{dep} generates current I_{dep}, which serves to decrease the charge stored on capacitor C_w, provided that the post-synaptic spike is still occurring. Bias voltages V_p and V_d are used to set an upper limit for the amount of charge which can be injected on to or removed from C_w. The change in V_{w0}, ΔV_{w0} is given by Eqns. (1) and (2), where C_p and C_d are the parasitic capacitances of nodes V_p and V_d respectively and Δt_{spk} is the pre- and post-synaptic spike width.

$$\Delta V_{w0} = \Delta t_{spk}(I_{pot}/C_p) \qquad \text{if } t_{pre}<t_{post} \qquad (1)$$

$$\Delta V_{w0} = \Delta t_{spk}(I_{dep}/C_d) \qquad \text{if } t_{post}<t_{pre} \qquad (2)$$

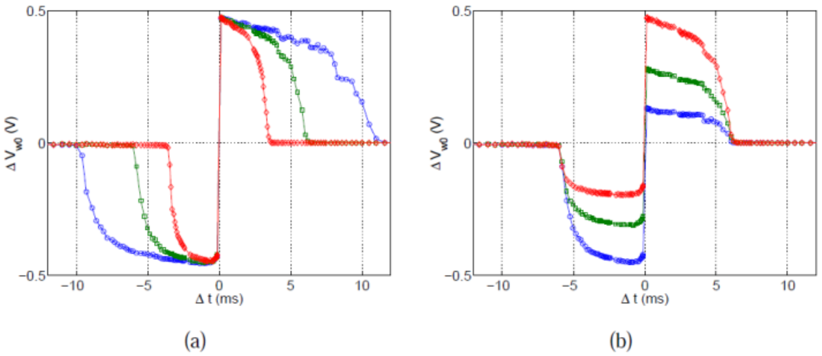


Figure 2.7 - (a) STDP curves obtained from experimental results by varying V_{pot} and V_{dep}. (b) STDP curves obtained for varying values of V_p and V_d [26]

Fig. 2.7(a) presents experimental results obtained for only varying V_{pot} and V_{dep} respectively, while Fig. 2.7(b) presents the situation for varying V_p and V_d only. In both cases it is clear to see that STDP type curves are obtained for the circuit. While the curves presented do not show the exponential decay of Fig. 2.1(b), they do show an STDP relationship in that if a pre-synaptic spike occurs first then the synaptic weight is increased and the magnitude of this increase is dependent upon Δt. Similarly if a post-synaptic spike occurs first, then the weight is decreased and again the magnitude is dependent upon Δt.

A variation on the standard STDP rule is presented in [27] using the circuit presented in Fig. 2.8. The STDP rule is changed such that instead of maximizing the synaptic weight change when pre- and post-synaptic spikes occur near-to simultaneously, the circuit aims to update the synaptic weight such that spikes become more coincident. The STDP curve for this circuit is presented in Fig. 2.9. Fig. 2.8 can be split into two main parts, N1-3 and P1-3 (MOSFETS) control the decrease in V_w (the synaptic weight), while N4-9 and P4-9 control the increase in V_w, which operates in “mirror image” to the decrease mode. To explain how the circuit operates, the decrease in synaptic weight is considered. When spike 3 occurs, capacitor C1 is discharge via N1, and then via P1 is charged up to V_{DD}. This causes a voltage pulse to occur at the gate of N8. This pulse defines the window in which synaptic weight change can occur. When spike 3 occurs at time t(3), capacitor C2 is discharged and then charged via N3 and P3 respectively. The resulting voltage pulse occurs at the gate of N7 and defines the magnitude of the weight change. As the voltage across C2 increases with time, the amount of weight change is also increased proportionally to the delay between t(3) and t(3p), (t(3p) is the time at which spike 3p occurs), until the maximum weight change of V_n is reached. Thus if spike 3p occurs within the time window, C_w is discharged through N7, N8 and N9 in proportion to the voltage across C_2. The reduction in V_w reduces V_{ramp} causing the next spike, 3p, to occur at a time closer to spike 3.

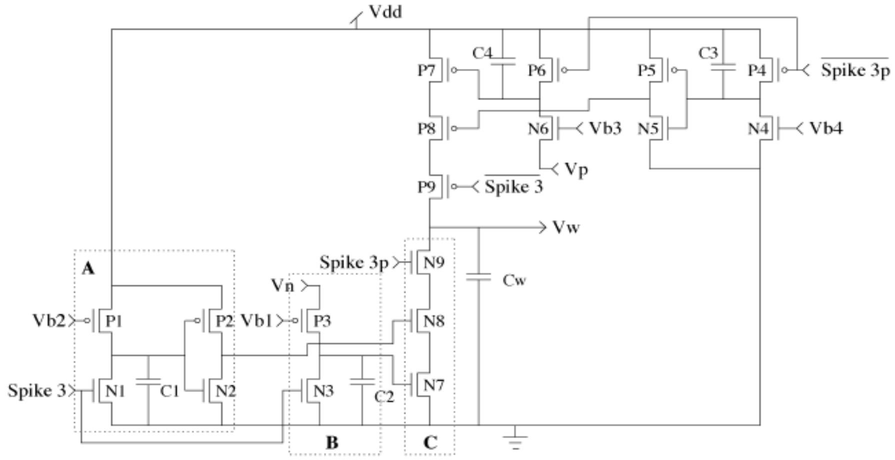


Figure 2.8 - Weight adaptation circuit proposed in [27]

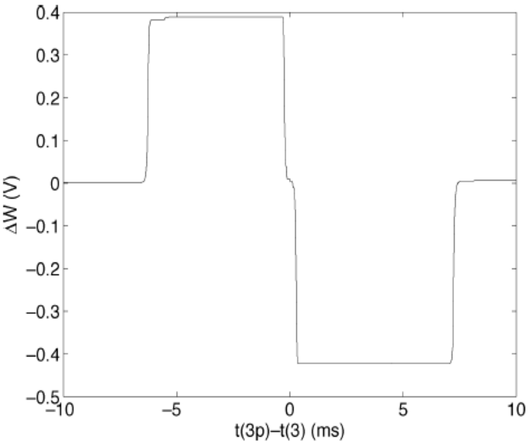


Figure 2.9 - Simulated STDP curve [27]

Another novel STDP circuit is presented in [28] and is constructed using decay-stages, integrators and SRAM. As with the circuit presented in [22], the circuit proposed in [28] is symmetric with respect to pre- and post-synaptic inputs, Fig. 2.10. The decay stage and integrator are used to implement potentiation and depression of the synapse. The SRAM is used to hold the binary state of the synapse, that is, if it is potentiated or depressed. Considering the case when the pre-synaptic spike occurs first; when the pre-synaptic spike occurs, the capacitor within the decay circuit is charged and then begins to discharge linearly. This charge is passed through an exponential function circuit and the resulting charge is passed to the integrator circuit when the post --synaptic spike occurs. Once in the integrator, the charge decays linearly again.

When the post-synaptic spike occurs, the cross-coupled SRAM reads the voltage which is on the integrator capacitor. If this voltage is greater than the threshold voltage of the SRAM, the SRAM switches state and becomes potentiated. Conversely if a post-synaptic spike occurs prior to the pre- synaptic spike, then the SRAM state is switched to depression-mode.

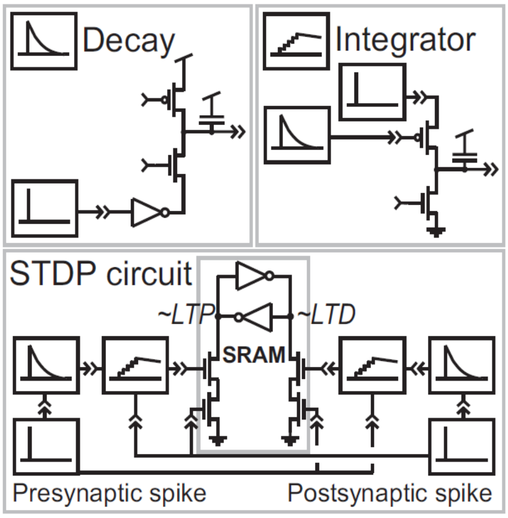


Figure 2.10 - STDP circuit constructed using decay, integrator and SRAM [28]

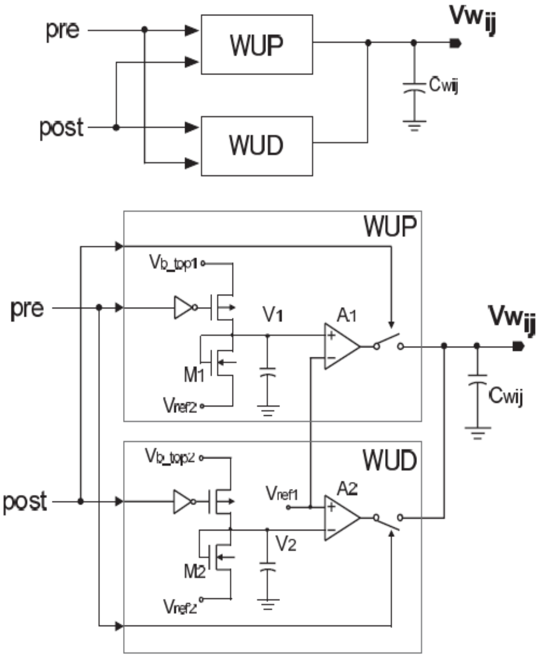


Figure 2.11 – Asymmetric STDP circuit [11]

In [11] an asymmetric STDP weight update circuit is presented. Fig. 2.11 depicts a circuit which can be used to generate the asymmetric STDP curve, shown in Fig. 2.1(b). The circuit block of Fig. 2.24 updates V_{wij} based upon the temporal order and time difference between the pre- and post-synaptic spikes. Increasing the synaptic weight is done using transconductance amplifier, A1. If a pre-synaptic spike occurs first, a non-linear waveform is generated, V_1 which occurs at the input to A1. If V_{ref1}=V_{ref2}+V_{th}, then A2 cannot update the synaptic weight. When the post-synaptic spike occurs, V_{wij} is increased proportionally to V_1(t)-V_{ref}. Decreasing the synaptic weight is done using transconductance amplifier A2, such that V_{wij} is decreased proportionally to $V_2(t)$-V_{ref}. In addition if pre- and post-synaptic spikes occur at the same time, then the current generated by A1 and A2 cancel each other out.

2.3 STDP in Hardware – Asymmetric Circuits

In this section we present a review of asymmetric circuits which implement STDP. An alternative method to implement STDP is described in [29-31] and shown in Fig. 2.12. The circuits allow for the implementation of an asymmetric decaying learning window, similar to that shown in Fig. 2.1(b). The circuit Fig. 2.12(a), updates the synaptic weight by increasing or decreasing the charge stored on capacitor C_w, represented by its equivalent voltage V_w, (V_w is inversely proportional to the weight). If a pre-synaptic spike occurs which has a pulse width in the order of µs, preLong, then I_{bpot} occurs and a voltage on transistor N5 is connected to N2, which decays with time across C_{pot}. When the post-synaptic spike occurs, this causes N3 to turn on. This causes the weight voltage V_w to decrease by an amount which reflects the time which has elapsed since the last pre-synaptic spike. Therefore, the synaptic weight is increased.

The synaptic weight is decreased in the circuit presented in Fig. 2.12(b). In this case when a non-causal interaction between pre- and post-synaptic spike occurs, the post-synaptic spike, (again with a pulse width in the order of µs), postLong, charges C_{dep}. The charge which is accumulated leaks through N3 and a set of nonlinear currents I_{dep_x} are sent to weight update circuit instead of I_{dep}. Thus when the pre-synaptic spike occurs, P1 (Fig. 2.4(a)) is switched on and V_w is pulled up to V_{dd}, causing a decrease in the synaptic weight. Experimental results are shown in Fig. 2.13.

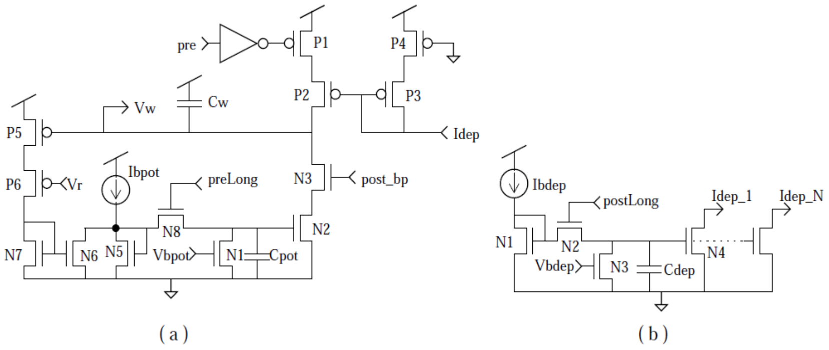


Figure 2.12 - (a) Weight update circuit. (b) Depression decay waveform circuit [30]

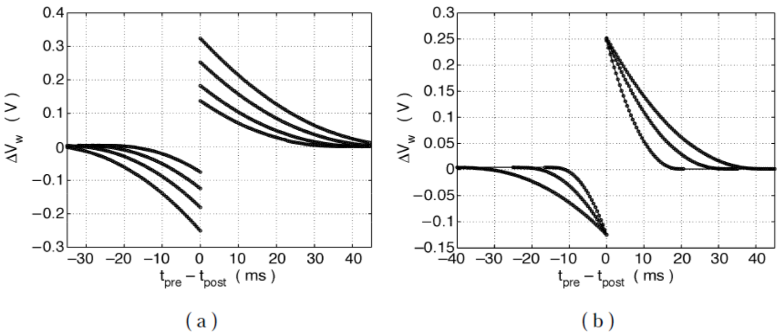


Figure 2.13 - (a) Variation in peak value of ∆Vw, caused by varying I_{bpot} and I_{bdep}. (b) Variation in decay rate of STDP curve due to varying V_{bpot} and V_{bdep} [31]

The results presented in Fig. 2.13 indicate that STDP can be implemented in HNN by the circuits presented in Fig. 2.12 and should be compared to those of the notional STDP curve presented in Fig. 2.1(b). From the results, it is clear to see that the shape of the STDP, the decay of the curve and maximum magnitude of the weight update are dependent upon I_{bpot}, I_{bdep} and V_{bpot} and V_{bdep} respectively. By varying these four values, various incarnations of the STDP curve can be implemented. Therefore, these values must be chosen prior to the implementation of the circuit within a neural network, and must be set as constants for subsequent use of the circuit when used with other synapses within the neural network.

Another possible method for implementing STDP is presented in [32]. In this method an STDP block is used which consists of two temporal summation blocks used in conjunction with a synaptic weight control block, as shown in Fig. 2.14.

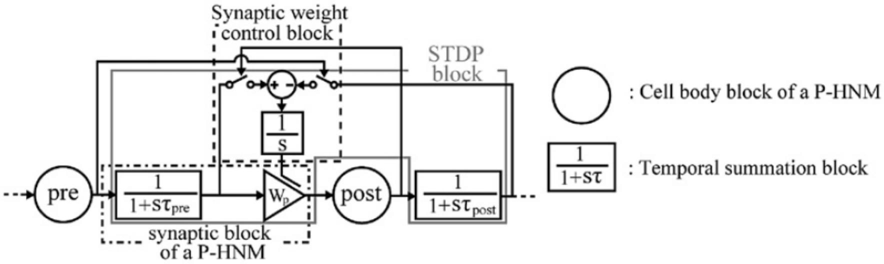

Synaptic weight control block
STDP block
+
-
$\frac{1}{s}$
pre
$\frac{1}{1+s\tau_{pre}}$
W_p
post
$\frac{1}{1+s\tau_{post}}$
synaptic block of a P-HNM
: Cell body block of a P-HNM
$\frac{1}{1+s\tau}$
: Temporal summation block

Figure 2.14 - STDP block diagram containing pulse-type neuron model (P-HNM) [32]

Fig. 2.15 shows the temporal summation circuits and synaptic weight control circuit which is used to update the synaptic weight using STDP.

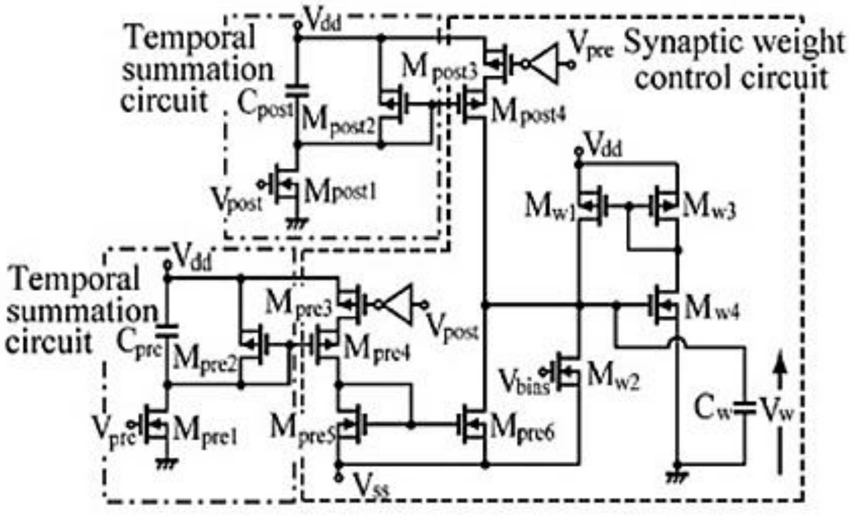


Figure 2.15 - Synaptic weight generation circuit [32]

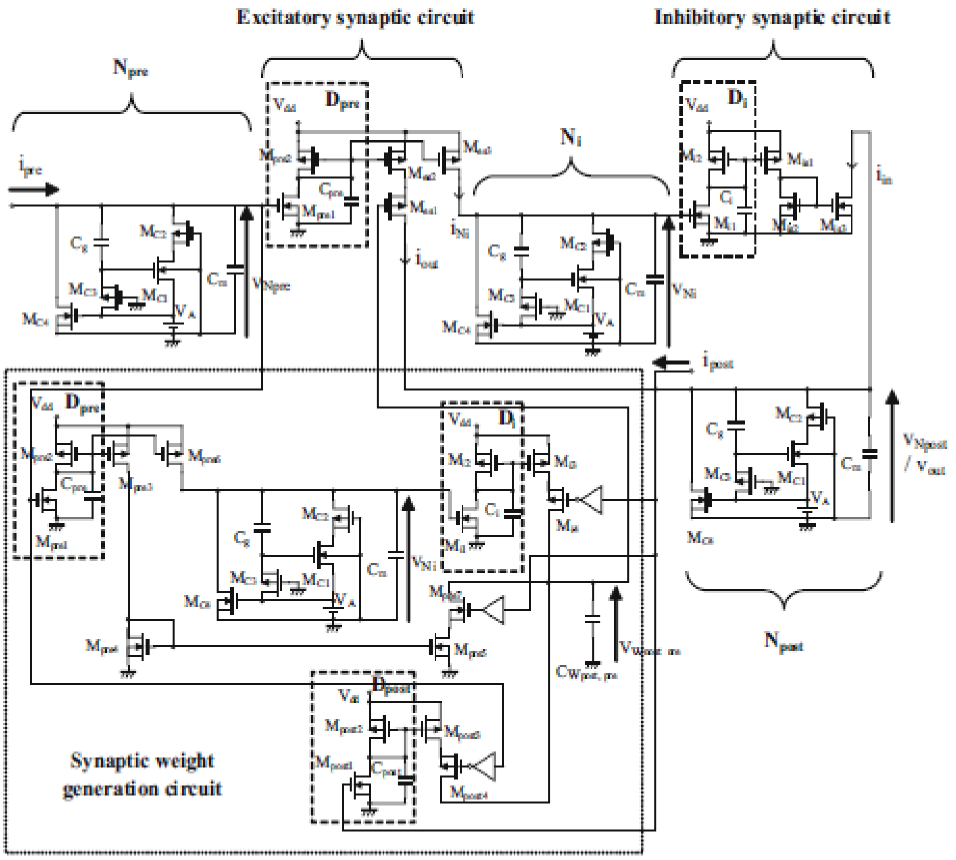


Figure 2.16 - Pulse-type hardware neural network with STDP circuit [33]

An improvement to the circuit presented in Fig. 2.15 is presented in [33], and shown in Fig. 2.16. In both cases, the synaptic weight is stored on a capacitor, C_w and this is updated by a change in current, additionally the STDP circuit is used in conjunction with inhibitory inter neurons. Fig. 2.17 indicates that this circuit [33] can be used to generate both asymmetric and symmetric STDP, where as Δt→0, ΔV_w is suppressed, due to the use of inhibitory inter neurons.

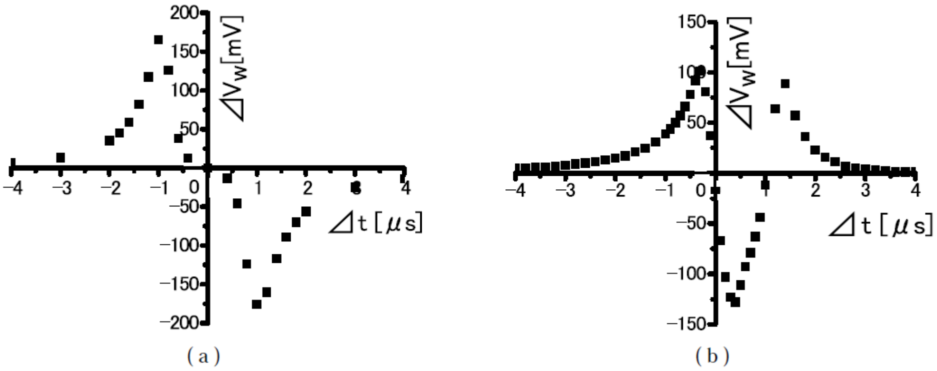


Figure 2.17 - Experimental results (a) Asymmetric time window, (b) Mexican-hat time window [33]

An STDP circuit constructed using two limiters, a delay-stage and correlator is shown in Fig. 2.18[34]. The limiters within the circuit are used to convert voltage pulses of U_1 and U_2 into sub-threshold currents pulses. A bias current I_1 drives m_4 and m_5 and m_6 is used to generate current I_2 since m_4 and m_6 have a common gate. Transistor m_7 is turned on by the application of V_{DD} to U_1 (subsequently it can be turned off by the application of 0V), and I_1 is copied to m_8, which forms part of a current mirror with m_8 and m_9. A pMOS common-source amplifier is formed by m_9 and m_{10}, with a gain which is proportional to V_{b1}. As V_{b1} tends to 0, the gain of the amplifier increases. Parasitic capacitor, C_2, is Miller-multiplied by the pMOS amplifier, thus temporal changes of U_2 are delayed at the output of the amplifier, D_{out}.

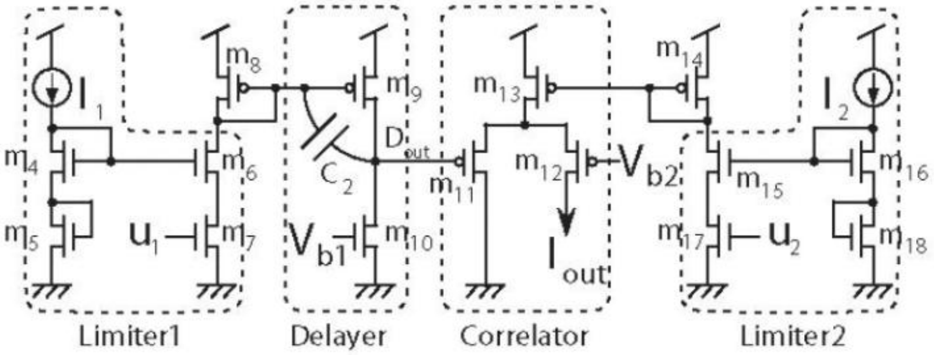


Figure 2.18 - STDP circuit constructed using two limiters, delayer and correlator [34]

The correlator is constructed using a pMOS differential pair, m_{11},$_{12}$ with bias transistor m_{13}. When U_2 has a voltage of V_{DD} applied to its gate, I_2 is copied to m_{13} through the current mirror constructed using m_{15} and m_{16}. Current I_{out} is only obtained when $U_2 = V_{DD}$. I_{out} is then integrated and normalized to produce the weight change; the results are presented in Fig. 2.19 and Fig. 2.20. Fig. 2.19 presents half of the ideal symmetric STDP curve which is generated by using Fig. 2.18 without limiters. Fig. 2.20 presents a full symmetric STDP curve which is generated by the circuit of Fig. 2.18 but this time with the limiters in place. From both sets of results it is clear to see that the integration of I_{out}, represents ΔV_w, and that as Δt is increased a decrease in I_{out} occurs, hence ΔV_w is decreased, until it reaches its minimum value and then begins to increase back to 0. This corresponds to the STDP curve shown in Fig. 2.1(a).

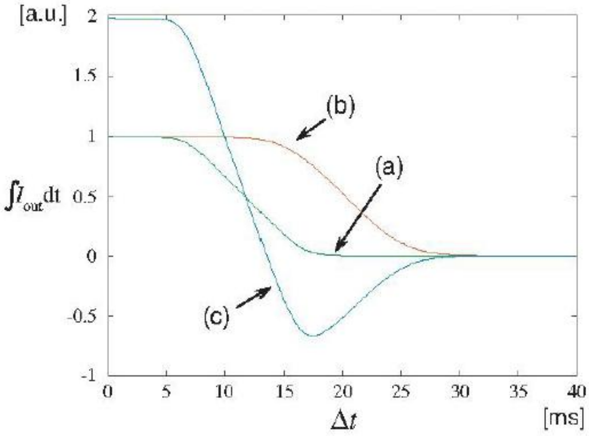


Figure 2.19 - STDP curve generated without using limiters [33]

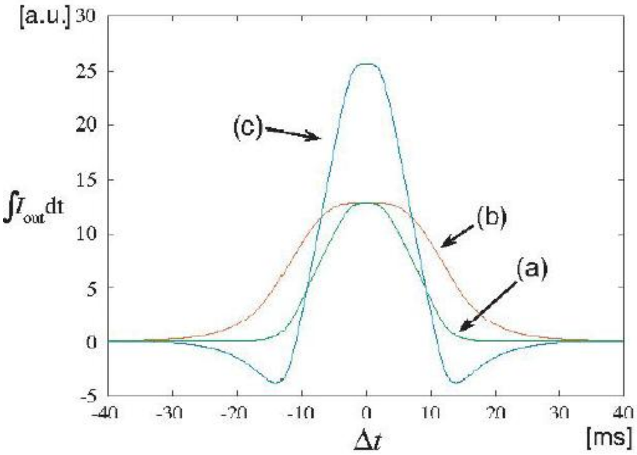


Figure 2.20 – Asymmetric STDP curve generated with limiters [33]

A more complex implementation of STDP in hardware is shown in Figs. 2.21, 2.22 [35]. In Fig. 2.21(a) the floating gate node, fg, is updated through the application of the control signals generated by the circuits presented in Fig. 2.21(c) and Fig. 2.22. Fig 2.21(b) represents the synapse proposed in [35].In Fig. 2.21(c), the circuits which generate P and M respectively are similar to that shown in Fig. 2.21(b) with the exception that fg is now replaced with a fixed bias voltage. Additionally the control pulses, vtunctrl and vinjctrl, which are generated upon the application of post- and pre-synaptic spikes respectively are generated by the circuit shown in Fig. 2.22 where vmem is replaced with P* and M* respectively.

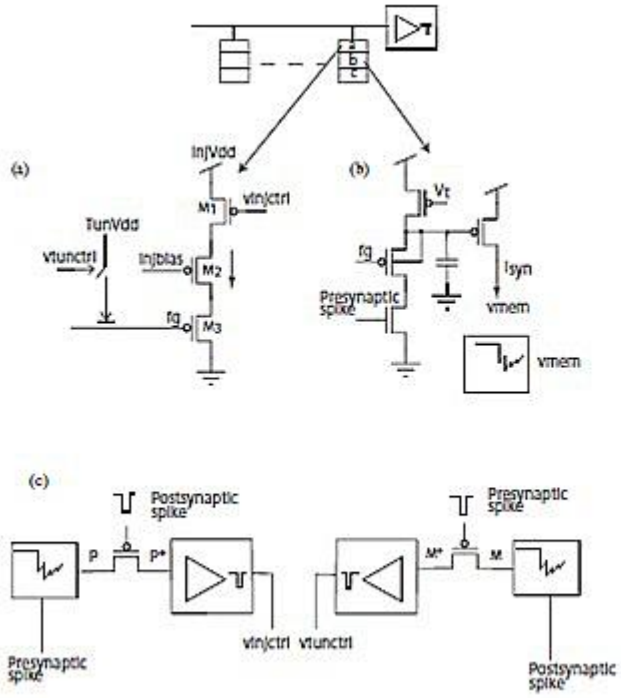


Figure 2.21 – (a) Floating gate node update circuit. (b) Proposed synapse. (c) Control pulse circuits, vinjctrl and vtunctrl are generated upon the occurrence of pre and post synaptic spikes. [35]

The circuit has been designed such that the synaptic weight is updated with a modified STDP rule. The modification which is implemented in [35] is that tunnelling and injection currents are only activated when the integrated sample or pre- and post-synaptic spike activity exceeds a predefined threshold value. This is done so that the gate oxide is not degraded over the life time of the network.

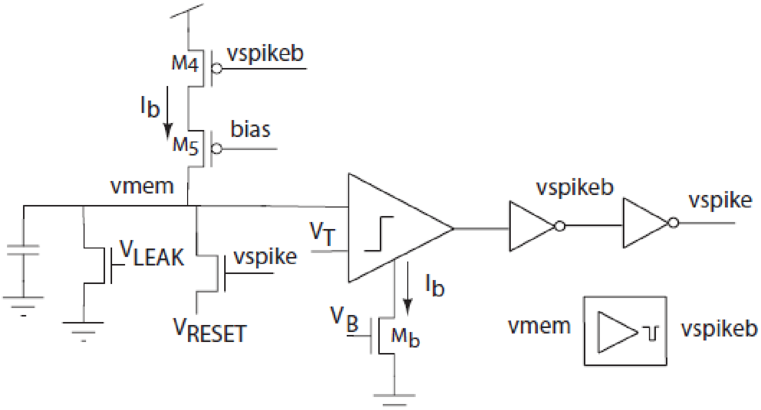


Figure 2.22 – Soma type circuit proposed in [35], which is used to generate P* and M* used in Fig. 21(c).

In [11] both symmetric and asymmetric STDP circuits are presented where the former is also presented in [36]. The symmetric STDP circuit is shown in Fig. 2.23. Additionally Fig. 2.24 presents the delay and inversion circuit, D&I, which is used in Fig. 2.23. The symmetric STDP circuit consists of two main blocks, namely that for spike-detection, SD, and weight update, WU. The toggle flip-flop (T-FF) in the SD block changes twice when pre- and post-input spikes occur, one after the other (note that the order in which they occur is not important). The changes in the T-FF are detected by the D&I and NOR circuits. The result of this detection is to send a first spike to WU via in1 and the second spike via in2.The synaptic weight, $V_{wij,}$ is updated based upon the time difference between pre- and post-synaptic spikes, by the WU block. The block WU updates V_{Wij} as follows; when a spike on in1 occurs, MOSFET M_A generates a ramp signal $V_A(t)$ on capacitor C_A which is controlled by bias voltage V_{b_rmp}. Simultaneously V_{SW} generated by D&I goes high and V_A is turned into a non-linear waveform via M_B and C_B and occurs as the positive input to the transconductance amplifier. When V_{SW} returns low, V_B returns to V_{ref} through resistor R. If in2 occurs while V_B is not equal to V_{ref}, then V_{wij} is updated through the charging or discharging of C_{wij}. However, if in2 occurs while V_B=V_{ref}, then the synaptic weight is not updated. The shape of the STDP curve, Fig. 2.1(a) is determined by the bias voltages within the circuit.

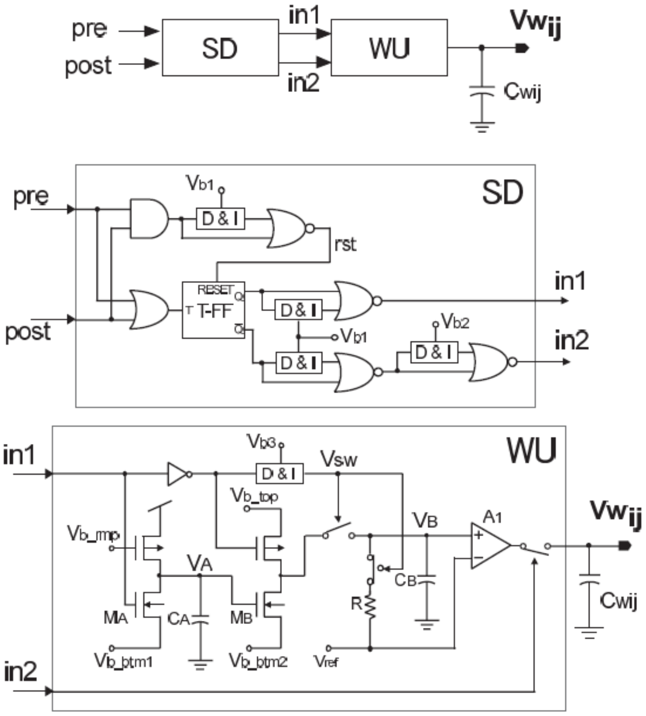


Figure 2.23 – Symmetric STDP circuit [11]

The circuit can be reset by the AND, delay and inversion, D&I, and NOR circuit. The reset occurs when pre- and post-synaptic spikes occur simultaneously and in this case the T-FF switches once. The D&I circuit used in Fig. 2.23 consists of three inverters and a bias MOSFET. The bias MOSFET is used to determine the delay time for the circuit.

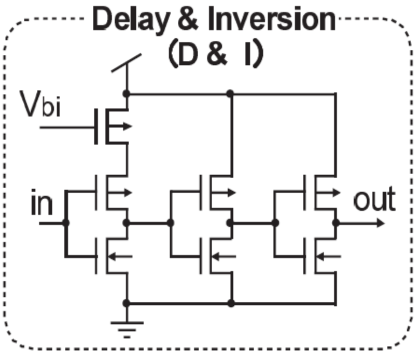


Figure 2.24 – Delay and Inversion circuit [11]

2.4 STDP in Hardware – Compact Approaches

A more compact implementation of STDP is presented in [43]. A two-terminal synapse which is constructed using PFET transistors with a shared, floating gate is shown in Fig. 2.25. The floating gate is updated based upon the timings of pre- and post-synaptic spikes. The pre-synaptic spikes are asserted to the sources of both transistors and body of the left transistor, while the post- synaptic spike is asserted to the drain of the left PFET only. Note that the post-synaptic spike defines the potential of the drain on this programming transistor. The drain of the right hand side PFET passes a current to an integration node.

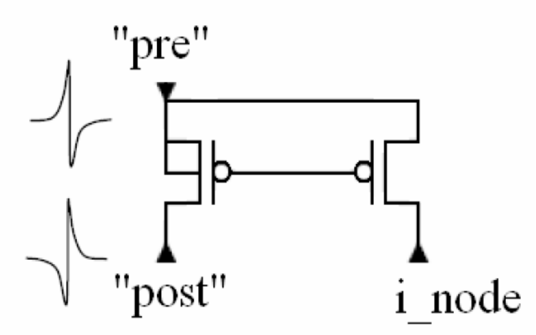


Figure 2.25 – Two terminal synapse which updates stored charge (representing the synaptic weight) via STDP. [43]

When a pre-synaptic spike is followed by a post-synaptic spike, both pre- and post-synaptic spikes will overlap resulting in a large transient difference in the source-drain voltage of the programming PFET. This causes hot electron injection to occur, such that electrons tunnel onto the floating gate, thus decreasing the stored voltage, and hence increasing the synaptic weight. If a post-synaptic spike is followed by a pre-synaptic spike, then FN tunnelling occurs. In this case electrons tunnel of the floating gate which is increases the stored voltage, thereby decreasing the associated synaptic weight. Simulations showing STDP are shown in Fig. 2.26.

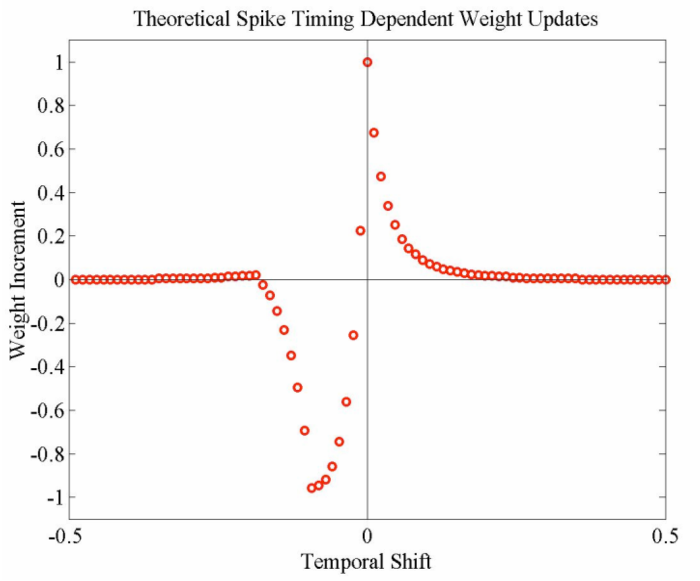


Figure 2.26 – Simulated STDP curve using circuit presented in Fig. 27 [43]

2.5 STDP in Hardware – Systems Level Approaches

SpiNNaker is a massively-parallel multiprocessor chip, [39], which is designed to mimic neural computation. Fig. 2.27 presents the SpiNNaker chip layout which contains 20 ARM968 processing cores with dedicated memory, and 1GB of off-chip SDRAM which is used to hold the synaptic weight of each synapse between neurons. Spiking neural networks have been implemented using SpiNNaker chips, [40], which implement a modified Izhikevich model, [49]. In addition, STDP has also been implemented, [41].

The approach used in the design of the SpiNNaker concept is to use Izhikevich neurons with STDP synapses, while maintaining a biologically plausible amount of connectivity within the network. This allows real-time simulations of neural networks which are biologically inspired rather than biologically accurate, [39],[40],[50]. Biological data is used as a source of inspiration for the models but not as a constraint for the design and implementation of the network [50],

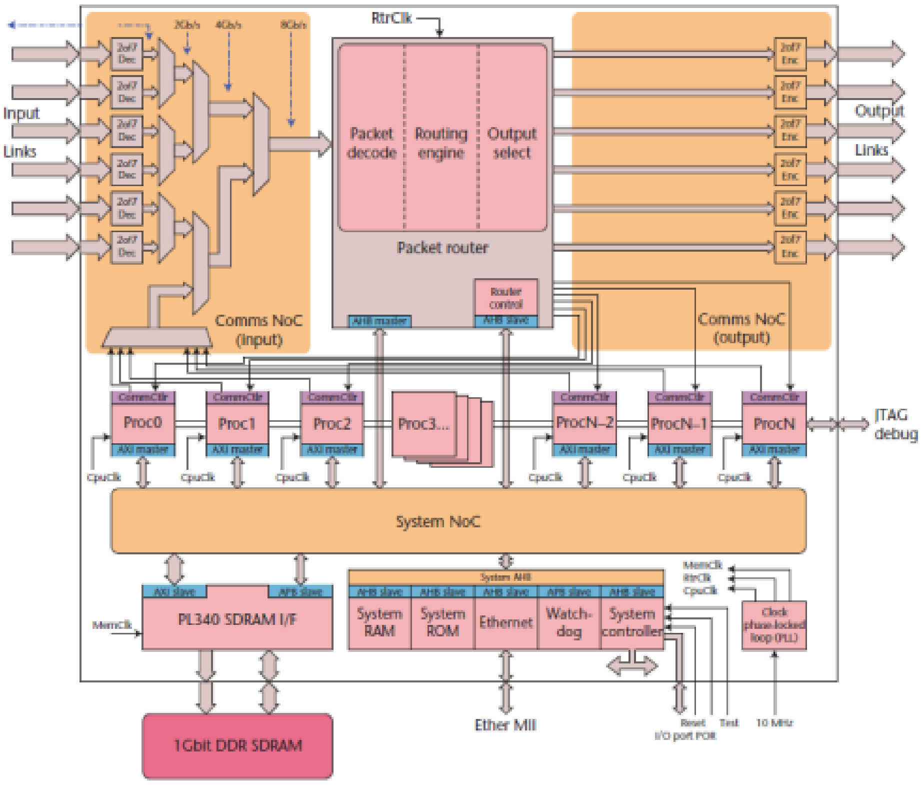


Figure 2.27 – SpiNNaker chip architecture [40]

In addition to the SpiNNaker project, another massively parallel on-chip system is the Blue Brain project. The Blue Brain project aims to build a complete biologically accurate model of the human brain for simulation on a custom designed IBM Blue Gene supercomputer [51]. Detailed biologically plausible models of the various neurons within the brain have been established [51-53] to facilitate the building of a model of the human brain. These models are then used to realise the fundamental building block of the cerebral cortex, the neocortical column. The neocortical column consists of approximately 10,000 interconnected neurons, and within the human brain there are over a million instances of the neocortical column [51, 52]. Synaptic connections are modelled using results from physiological recordings [52] such that synaptic biophysics and dynamics are taken in to account. Additionally, synaptic plasticity is implemented on both local and global levels within each neocortical column.

In order to simulate a neocortical column, 10 neurons and their synaptic connections are mapped on to each processor within the Blue Gene supercomputer. With this mapping it is possible for over 100,000 complex neurons to be simulated [51-58]. As with SpiNNaker, the Blue Brain project uses mathematical equations to represent each neuron and synapse together with their connectivity and learning methods. The equations used to model the synapse and their functionality use a biological approach based on ion channels contained within the synaptic clef of two connecting neurons. This is considered to make the overall simulation approach biological accuracy rather than simply biologically plausible or inspired [51-58]. Hence STDP is implemented within the Blue Brain project using biologically relevant algorithms rather than as physical circuits.

An alternative approach uses a custom field programmable neural network architecture (EMBRACE). EMBRACE merges the programmability features of FPGAs and the scalable interconnectivity of Networks-on-Chip (NoC) with low-area/power programmable synapse cells to realise large scale SNNs [63]. The strategy uses individual NoC routers to group multiple synapses and the associated neurons using a novel structure referred to as a neural tile. The neural tile is viewed as a macro-block of EMBRACE and merges analogue synapse/neuron circuitry (see Fig. 27 and Fig. 28) with NoC digital interconnect to provide a scalable and reconfigurable neural building block. The EMBRACE NoC architecture is a mesh-based two-dimensional array of interconnected neural tiles, where spike exchanges between tiles are achieved by routing packet-based spike events across the array. This system level approach is potentially a candidate for mobile neural computing architectures.

In [75] the Perplexus project presents a scalable platform consisting of custom reconfigurable bio-inspired and compatible devices is proposed which can simulate complex neural network of 10000 with 3000 synaptic connections [76]. The platform aims to do this through rich interaction with its intended environment through sensors and also by replacing artificial constraints imposed by the programmer with physical constraints imposed by the hardware. The platform consists of a set of VLSI ubiquitous computing modules, ubidules. Each ubidule contains two ubidule bio-inspired chips, ubichips. The ubichip is comprised of four main parts, a configurable array, encoder/decoder, memory controller and system manager, Fig. 2.28. It is this architecture which allows for the implementation of biological based reconfigurable mechanisms such as dynamic routing, self replication through self reconfiguration and simple connectivity. Using these mechanisms it is possible for biological features such as learning, plasticity and evolution (extrinsic, intrinsic, complete and open-ended) to be implemented by the ubichip [75, 76].

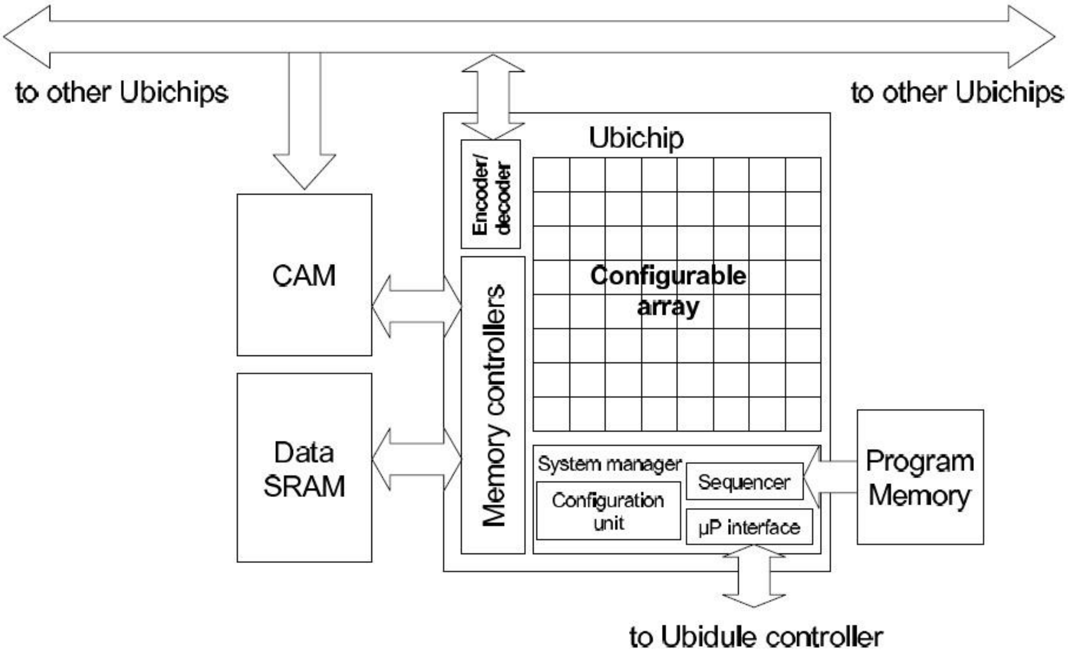


Figure 2.28 – Ubichip architecture [75, 76]

2.6 Discussion

Given the reliability and maturity of CMOS technology, it is looking increasingly unlikely that alternative charge-based technologies will ever compete in the building of large scale computing engines. Furthermore, with the inevitable breakdown in Moore’s law, the semiconductor industry is actively pursuing new mobile computing devices and architectures to augment CMOS with a new set of computational capabilities. A likely candidate for this new generation of computing devices is neural networks because they offer the possibility of autonomous self-repairing architectures. The ability to self-repair is becoming even more crucial because the continual reduction in feature sizes is now resulting in a significant reduction in yield due to vulnerable parametric failures. Neural network computing architectures offer self-repair because each computing node only supports a fragment of the overall problem representation. Therefore, they can tolerate a scattering of faults with minimal loss to the representation and additionally minute losses in the information processing capability can be partially or in some cases fully undone by a continual repair process [74]. However, the devices and circuits that implement the computing nodes must be compact and consume minimal power for mobile computing devices, while at the same time be faithful to biology.

Existing STDP circuits have been reviewed in the previous section and can be segregated into two main categories. The first type are circuits which are symmetric with respect to pre- and post-synaptic inputs, [11], [21-25],[27,28,33,35,36], they use two identical circuits one to increase and another to decrease the synaptic weight. The second type serve to function as decision circuits and are not symmetric with respect to inputs, [11], [29-32], [35], [39-41], [43], in that it is one dedicated circuit which will update the synaptic weight on average, and with the exception of [39-41] and [43], the majority of the circuits proposed require between 6-30 MOSTs in order to implement STDP and in some cases the smaller circuits require additional peripheral circuitry. We have also reviewed system level implementations. The SpiNNaker and Blue Brain architectures are not suitable platforms for general purpose or mobile computing because of power and area constraints. The approach in EMBRACE, which uses compact CMOS to implement the synapse and learning function and NoC routers to communicate between cells, holds potential. Given the state of the art in neural network implementation it would be interesting and valuable to explore a roadmap to useful computing architectures and we address this in the next sections.

The hardware implementations presented in the previous sections indicate that there is a fundamental trade-off between biological accuracy and engineering constraints on HNN design [50]. A biological neuron is a flexible component which can be used in a diverse number of functions from image processing in the visual cortex, to long term memory in the brain. The challenge in HNNs is to replicate this functionality of the neuron while implementing a compact low power device. There are two main approaches which are used in HNNs [50].

The first is to produce a network which is biologically accurate. Biologically accurate HNNs are designed such that they model the large connectivity of biological NNs, which leads to problems as most CMOS process only have a limited number of metal layers. This limitation can be reduced by implementing multiplexing however this can lead to additional problems. A key problem relates to when spikes occurring simultaneously within a short amount of time, causing the multiplexer to drop one of them. This event then introduces timing errors into the system. In addition the neuron, synapse and weight update circuitry are generally complex with commensurately poor power efficiency [50]. The second approach is to capture only the required functionality of neurons, synapses and synaptic plasticity. The point neuron is the simplest type which can be implemented, the so-called leaky integrate and fire neuron [50]. This model can be made more complex by introducing habituation, a decrease in the neurons response to repeated stimuli over a set period of time, by increasing its threshold value when input spikes occur. Furthermore, refractory periods can be built into the model. For the case of synaptic plasticity, this means that only a STDP or similar weight update method is implemented rather than STDP enhanced by a reward mechanism. The STDP circuits presented in previous section constitute only a single, synapse. The circuits are relatively complex particularly when the weight update functionality is incorporated. However the papers, do not present considerations of scaling to build neural networks which can perform useful functions.

Before continuing with consideration of construction of a complex neural network, the overall complexity and size of the weight update, STDP circuits presented in the previous section will be considered. The circuits presented can be split into two main categories. The first type are circuits which are symmetric with respect to pre- and post-synaptic inputs, [11], [21-25],[27,28,33,35,36], they use two identical circuits one to increase and another to decrease the synaptic weight. The second type serve to function as decision circuits and are not symmetric with respect to inputs, [11], [29-32], [35], [39-41], [43], in that it is one dedicated circuit which will update the synaptic weight on average, and with the exception of [39-41] and [43], the majority of the circuits proposed require between 6-30 MOSTs in order to implement STDP and in some cases the smaller circuits require additional peripheral circuitry.

To get some idea of scalability, consideration of the implementation of a network to solve the simplest benchmark problem, namely the exclusive-OR (XOR) function. The XOR problem requires only two inputs which are linearly inseparable. The smallest possible neural network which can implement and solve this problem is presented in Fig. 2.29. The network consists of two input neurons, two hidden layer neurons and one output layer neuron. If the network is fully connected, all neurons in each layer connect to all others in the next layer giving a total synapse count of six. Therefore six weight update circuit blocks are required in addition to the six synapses and five neurons.

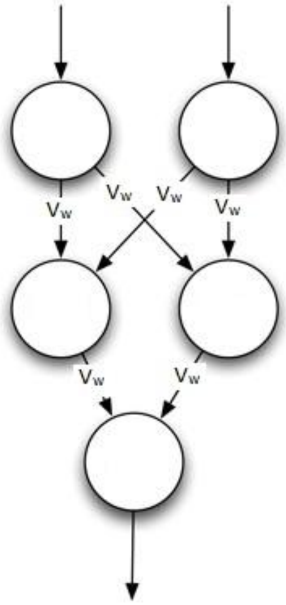


Figure 2.29 – XOR neural network architecture

An example of a more complex neural network incorporating so-called hidden layers is presented in Fig. 2.30. This unspecified network consists of two input neurons, four neurons in the first hidden layer, four in the second and one output neuron. Again it is assumed that the network is fully connected. The total number of synapse in this network is now 28; hence 28 instances of the weight update circuit are required.

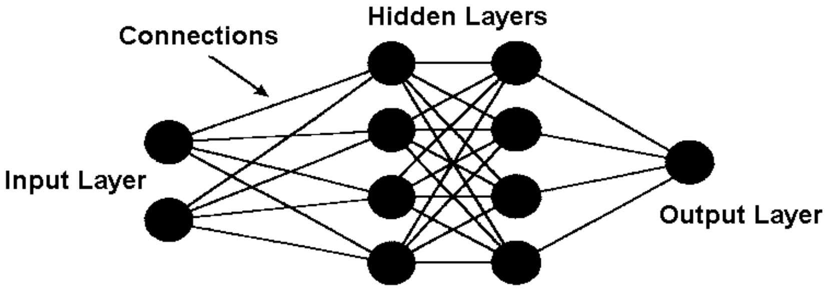


Figure 2.30 – A more complex neural network consisting of two input neurons, 4 neurons in the first hidden layer, 4 neurons in the second hidden layer and 1 neuron in the output layer

We now estimate the footprint of these circuits in a 0.35µm Austria MicroSystems, AMS, process where;

- A minimum feature size NMOS with width = 0.4 µm and length = 0.35 µm has an approximate area of 3.58µm^2.
- A minimum feature size PMOS (including n-well) with width = 0.4 µm and length = 0.35 µm has an approximate area of 14.1µm^2.

Thus for the smallest circuit proposed in [24] which contains 3 NMOS and 3 PMOS devices the approximate area for the circuit block is 53 µm^2. This excludes metal and capacitors. For the largest circuit proposed in [21] which contains 12 NMOS and 18 PMOS devices the approximate area is ≈300 µm^2; again this excludes metal and capacitors.

The two extreme weight update, STDP, circuits proposed in the previous section are now considered for use in either of the two aforementioned neural networks, (with the neuron and synapse proposed in [44-47]). For the simple XOR problem:

- The 6 transistor circuit would occupy 6x53= 318µm^2
- The 30 transistor circuit would occupy 6x300= 1800µm^2

and for the complex neural network proposed in Fig. 29;

- The 6 transistor circuit would occupy 28x53= 1484µm^2
- The 30 transistor circuit would occupy 28x300= 8400µm^2

In comparison the total area occupied by a single synapse whose area is 25 µm^2 [47];

- In the XOR network the synapses would occupy 6x25= 150µm^2
- In the complex network the synapses would occupy 28x25= 700µm^2

By comparing the area taken up by the synapses to that required to update the synaptic weight for both neural networks it is clear that the majority of the silicon would be taken up purely by the weight update circuit. This exercise serves to emphasis the pressing need to design and implement a weight update circuit which is compact and low-power for scalable NNs which can implement useful functions.

A useful neural based computing architecture is one which supports a network of neurons of sufficient density to solve useful problems. However, neural network sizes vary greatly depending on the application. For example, a character recognition problem was implemented using a three layered network with just over 300 neurons [69] whereas the implementation of the locomotion system for a C-elegant used a total of 302 neurons[70] and a model for the retina employed 51 neurons [71]. A more complex application involving an automatic speech recognition problem using the TI46 speech corpus dataset used approximately 8000 spiking neurons [76] while an image processing applications required a more complex network of the order of one million spiking neurons [75]. Considering the latter as our target network size, there are several problem areas which must be considered in achieving this density of neurons. There are technological issues, which currently exist for conventional device/circuit processing, such as reducing the gate dielectric thickness and gate length, increasing the channel doping to control short channel effects and minimizing signal delay and power consumption on interconnect wires. While these processing issues give rise to device parameter variability and consequent function failure across the chip, they may not be as severe when using a neural network information processing platform. This is because in these architectures the processing is distributed across many nodes, so they are fault-tolerant to an extent, and furthermore these architectures can be tuned using the learning capability to restore functionality.

From an architectural perspective, biological systems do not appear to be very scalable in 2D. However, for HNN, a mixed signal approach looks to be appropriate because the core cell functions are analogue while information between cells is communicated in, essentially, a digital format. It follows from this observation that there is a need for small geometry, low-power analogue synapses and neuron cells, since brain development is constrained to maximize functionality within minimal space. Furthermore, considerations must be given to capturing plausible learning rules such as STDP and the associated weight storage mechanism. While off-chip learning and weight storage has been proposed it appears that there is a scale limitation due to memory weight bandwidth [73]. Consequently, we need to consider localized circuitry for the learning rule as well as some form of floating-gate mechanism with drivers for charge control into and off the storage node. Clearly the footprint associated with the circuits required for implementing learning synapses dominates these architectures, particularly as there are orders of magnitude more synapses than neurons. Consequently small geometry and low power synapses are a further key requirement for scalable neural network architectures.

One of the major issues with all modern electronic circuits is interconnect because it consumes large areas of potential logic real estate and leads to longer critical path delays. In particular, modern digital architectures, such as FPGAs, typically exhibit switching requirements which grow non-linearly with the mesh sizes or number of logic units on the device [72]. Also with the ever increasing demand for larger logic densities, new strategies are required to support scalable interconnect and the routing of larger customizable and dedicated logic clusters. Current manufacturers of devices cannot sustain the growth in densities and simultaneously support the configurable routing performance requirements of larger future platform devices.

Researchers have investigated several routing optimizations and topologies in attempts to improve the routing latency-performance of FPGAs. For example, optimizing the performance of switch blocks located within logic clusters by replacing selected buffers with pass transistor, providing an average 7% decrease in circuit delays for a given set of benchmarks has been reported [63]. Techniques to reduce the routing interconnect density of devices with the introduction of a bus-based routing topology were multi-bit buses were used to interconnect between clusters of multi-bit logic blocks have also been reported [65]. Similar topology work has been investigated with the use of network-on-chip (NoC) where packet and switch-based routers are realized on FPGAs to support connectivity [66]. The EMBRACE architecture has opted for the NoC based routing scheme with analogue cores [63]. It has the potential to provide core packing density with an inter-neuron communication scheme that avoids some of the issues associated with FPGAs. If this approach could take advantage of multi-level metallization then we envisage that our target neuron density could be met.

They are of course other techniques available to improve interconnect density and minimize parasitic's. For example, the use of nanowires for creating routing topologies. In particular Snider [67] and Gojman [68] consider using existing Manhattan style layouts and incorporating nanowires to create the interconnect medium. Similarly, workers at the California Institute of Technology utilised multi-level nanowires on a programmable logic array and achieved an average 18% reduction in routing delays. Such new innovations sound promising in providing increased routing performances, however, nanotechnology is currently limited by susceptibility to defects and hence the production yield required to implement working circuits. Ultimately future routing performance improvements will demand both innovative architectural topologies and migration to new process technologies, addressing the reduction in routing complexity, flexibility of programmability, and ensuring scalability with the ever increasing demand for traffic throughput with minimal power consumption.

The constraints for designing and implementing a STDP weight update circuit within hardware are that the circuit has to implement the asymmetric STDP curve, Fig. 2.1(b), (or a variation on this curve). While the symmetric STDP curve, Fig. 2.1(a) is biologically plausible, its occurrence in biological neural networks is small compared to the asymmetric STDP curve. In addition to this, and as mentioned previously, the STDP circuit should be compact.

Additionally there is a trade-off between the ‘accuracy’, (functionality) of the biological equivalence and the limitations of physical design and implementation of the NN systems: biological plausibility versus engineering. Over the years hardware implementations of neurons and synapses have focused upon implementing devices which provide the same functionality as their biological counter parts, [39-41]. In order to do this, the circuits, for both neurons and synapses have become large and complex [2 4 5], [22-25], [39-41]. However more recent models have been proposed which replicate some of the main basic functions of neurons [9], [46 47] and synapses [6-8],[10], using simpler circuit designs.

Similarly the weight update circuits presented here have shown that biologically plausible STDP can be implemented in hardware but the circuits are complex. Thus there is a limit to the biological accuracy of the HNNs, with respect to biological equivalence and what can be physically implemented in hardware with a small enough footprint to allow scaling. One key requirement in building scalable blocks is to seek alternative learning rules so as to produce simpler implementation of STDP in hardware to allow scaling to produce useful functionality.

2.7 Conclusions

Implementation of spike-timing-dependent plasticity, STDP, in hardware neural networks has been reviewed. The circuits are either symmetric with respect to pre/post synaptic spikes or are of a decision circuit type. However while it has been shown to be possible to implement STDP within HNN, little consideration has been given as to both the functionality of more complex circuits and scalability of the circuit blocks. In order for the synaptic weight to be updated the proposed synapse(s) need additional circuitry, which can be used to increase or decrease the stored synaptic weight between associated neurons. Therefore, each synapse will have to have its own weight update circuit and the large footprint of the overall solution severely limits their usefulness. However, there is a trade-off between the accuracy of the STDP implemented and that of the additional circuitry. If STDP is to be biologically accurate, for example by implementing ion channel functionality then complex mathematical modelling and dedicated processors are required, as is used in SpiNNaker, the Blue Brain Project and the Facets project, [21-22], [39-41], [51-58]. However if biologically plausible circuits are used then more compact CMOS circuits can be achieved, [23-38].

When scaling is considered, it is clear to see that the additional circuitry required to implement STDP will take up the majority of the silicon. Therefore an alternative method of implementing synaptic plasticity within HNNs needs to be considered. These new methods must allow for the update in synaptic weight while maintaining a compact and low power circuit design, such that more complex neural networks can be constructed. Future work will look at alternative biologically synaptic weight update methods which can be implemented in HNN using compact circuits. The proposed methods will be assessed both for their feasibility when implementing a neural network and their biological plausibility.

2.6 References

[1] D. H. Goldberg, G. Cauwenberghs, and A. G. Andreou, “Probabilistic synaptic weighting in a reconfigurable network of VLSI integrate and-fire neurons,” *Neural Networks,* vol. 14, pp. 781-793, 2001.

[2] E. Chicca, D. Badoni, V. Dante, et al., “A VLSI recurrent network of integrate-and-fire neurons connected by plastic synapses with long term memory,” *IEEE Transactions on Neural Networks,* vol. 14, no. 5, pp. 1297–1307, Sep. 2003.

[3] A. Bofill-i-Petit and A. F. Murray, “Synchrony detection and amplification by silicon neurons with STDP synapses,” *IEEE Transactions on Neural Networks*, vol. 15, no. 5, pp. 1296–1304, Sep. 2004.

[4] G. Indiveri, E. Chicca, and R. Douglas, “A VLSI array of low-power spiking neurons and bistable synapses with spike-timing dependent plasticity,” *Neural Networks, IEEE Transactions on,* vol. 17, pp. 211-221, 2006.

[5] C. Mead, *Analog VLSI and Neural Systems*. Addison-Wesley, 1989.

[6] Y. Chen, L. Mcdaid, S. Hall and P. Kelly, “A programmable facilitating synapse device.”, 2008 International Joint Conference on Neural Networks, IJCNN 2008, Institute of Electrical and Electronics Engineers Inc pp1615-1620, 2008.

[7] C. Diorio, P. Hasler, B. A. Minch, and C. A. Mead, “A floating-gate MOS learning array with locally computed weight updates,” *IEEE Transactions on Electron Devices*, vol. 44, no. 12, pp. 2281–2289, Dec. 1997.

[8] Y. Chen, S. Hall, L. McDaid, O. Buiu, and P. Kelly, “A silicon synapse based on a charge transfer device for spiking neural network application,” in *Lecture Notes in Computer Science*, vol. 3973, J. Wang, Eds. Germany: Springer-Verlag, 2006.

[9] T. Shibata and T. Ohmi, “A functional MOS transistor featuring gate level weighted sum and threshold operations,” *IEEE Transactions on Electron Devices*, vol. 39, no. 6, pp. 1444–1455, June 1992.

[10] C. Diorio, P. Hasler, B. A. Minch, and C. A. Mead, “Single transistor learning synapses with long term storage,” *IEEE Int. Symp. On Circuits and Systems*, vol. 3, pp. 1660-1663, 1995.

[11] D.O. Hebb. *The Organisation of Behaviour.* Wiley 1949.

[12] A. Carling *Introducing Neural Networks.* Sigma Press 1992.

[13] W.B. Levy and O. Steward, “Temporal contiguity requirements for long-term associative potentiation/depression in the hippocampus,” *Neurosience*, vol. 8, no. 4, pp. 791-797, 1983.

[14] G.Q. Bi and M.M Poo, “Synaptic modification in cultured hipocampl neurons: Dependence on spike timing, synaptic strength and postsynaptic cell type,” *J. Neuroscience*, vol. 18, pp. 10462-10472, 1993.

[15] M.Nishiyama, K. Hong, K. Mikoshiba, M.M. Poo and K. Kato, “ Calcium stores regulate the polarity and input specificity of synaptic modification,” *Nature*, vol. 408, pp. 584-588, 2000.

[16] M. Tsukada, T. Aihara, Y. Kobayashi and H. Shimazaki, “ Spatial analysis of spike-timing-dependent ltp and ltd in the ca1 area of hipocample slices using optical imaging,” *Hippocampus*, vol. 15, no. 1, pp. 104-109, 2005.

[17] H. Tanaka, T. Morie, and K. Aihara, “A CMOS spiking neural network with symmetric/asymmetric STDP function,” *IEICE Transcations on Fundamentals,* vol E92-A, no. 7, pp. 1690-1698, 2009.

[18] G.Q. Bi and M.M Poo, “Synaptic modification of correlated activity: Hebbs postulate revisited,” *Annu. Rev. Neurosci*, vol. 24, pp. 139-166, 2001

[19] N. Caporale and Y. Dan, “Spike timing-dependent plasticity: A Hebbian learning rule,” *Annu. Rev. Neurosci*, vol. 31, pp. 25-46, 2008

[20] L. F. Abbott and S. B. Nelson, “Synaptic plasticity: taming the beast,” *Nature Neuroscience Supplment*, vol 3, pp. 1178-1183, 2000.

[21] J. Schemmel, K. Meier and E. Mueller, “ A new VLSI model of neural microcircuits including spike timing dependent plasticity,” *IEEE International Joint Conference on Neural Networks 2004*, vol. 3, pp. 1711-1716, 2004.

[22] J. Schemmel, K. Meier and E. Mueller, “ Implementing synaptic plasticity in a VLSI spiking neural network model,” *IEEE International Joint Conference on Neural Networks 2006*, pp. 1-6, 2006.

[23] S. Mitra, S. Fusi and G. Indiveri, “A VLSI spike-driven dynamic synapse which learns only when necessary,” *ISCAS 2006*, pp. 2777-2780, 2006.

[24] G. Indiveri, “Circuits for bistable spike-timing-dependent plasticity neuromorphic VLSI synapses,” *Advances in Neural Information Processing System*, 2002.

[25] G. Indiveri, “Neuromorphic bistable VLSI synapses with spike-timing-dependent plasticity,” *Advances in Neural Information Processing System,* 2003.

[26] G. Indiveri, E. Chicca and R. Douglas, “A VLSI array of low-power spiking neurons and bistable synapses with spike-timing-dependent plasticity,” *IEEE Transactions on Neural Networks,* vol. 17, no. 1, pp. 211-220, 2006.

[27] K. Cameron, V. Boonsobhak, A. Murray and D. Renshaw, “Spike timing dependent plasticity (STDP) can ameliorate process variations in neuromorphic VLSI,” *IEEE Transactions on Neural Networks,* vol. 16, no. 6, pp. 1626-1637, 2005.

[28] J. V. Arthur and K. Boahen, “Learning in silicon; Timing is everything,” *Advances in Neural Information Processing System 17,* 2006.

[29] A. Bofill-i-Petit and A. F. Murray, “Synchrony detection by analogue VLSI neurons with bimodal STDP synapse,” *Advances in Neural Information Processing System,* 2003.

[30] A. Bofill-i-Petit and A. F. Murray, “Synchrony detection and amplification by silicon neurons with STDP synapse,” *IEEE Transactions on Neural Networks,* vol. 15, no. 5, pp. 1296-1304, 2004

[31] J. Huo and A. F. Murray, “The role of membrane threshold and rate in STDP silicon neuron circuit simulations,” *ICANN 2005, Lecture Notes In Computer Science*, vol. 3697/2005, pp. 1009-1014, 2005.

[32] Y. Hayashi, K. Saeki, and Y. Sekine, “A synaptic circuit of a pulse-type hardware neuron model with STDP,” *International Congress Series*, vol. 1301, pp. 132-135,

2007.

[33] K. Saeki, R. Shimizu and Y. Sekine, “Pulse-type hardware nerual network with two time window STDP,” *ICONIP 2008, Lecture Notes In Computer Science*, vol. 5507/2009, pp. 877-884, 2009.

[34] G. M. Tovar, E. S. Fukuda,T. Asai, T. Hirose and Y. Amemiya, “Neuromorphic CMOS circuits implementing a novel neural segmentation model based upon symmetirc STDP learning,” *Proceedings of International Joint Congress on Neural Networks,* pp. 897-901, 2007

[35] S. C. Lui and R. Mockel, “Temporally learning floating-gate VLSI synapses,” *ISCAS 2008,* pp. 2154 – 2157, 2008.

[36] H. Tanaka, T. Morie, and K. Aihara, “A CMOS for STDP with a symmetric time window,” *International Congress Series,* vol 1301, pp. 152-155, 2007.

[37] S. Fusi, M. Annunziato, D. Badoni, A. Salamon and D. J. Amit, “Spike-driven synaptic plasticity: Theory, simulation, VLSI implementation,” *Nerual Computation*, vol. 12, pp. 2227-2258, 2000.

[38] T. J. Koickal, L. C. Gouveia and A. Hamilton, “ A programmable spike-timing based circuit block for reconfiguable neuromorphic computing,” *Neurocomputing,* vol. 72, pp. 3609-3616, 2009.

[39] M. M. Khan, D. R. Lester, L. A. Plana, A. Rast, X. Jin, E. Painkras and S. B. Furber, “SpiNNaker: Mapping nerual networks onto a massively-parallel chip multiprocessor,” *Internation Joint Conference on Neural Networks 2008,* pp.2850-2857, 2008.

[40] X. Jin, M. Lujan, L. A. Plana, S. Davies, S. Temple and S. B. Furber, “Modeling spiking neural networks on SpiNNaker,” *Computing In Science and Engineering,* vol. 12, no. 5, pp. 91-97, 2010.

[41] X. Jin, A. Rast, G. Galluppi, S. Davies, and S. B. Furber, “Implementing spike-timing-dependent plasticity on SpiNNaker neuromorphic hardware,” *World Congress on Computational Intelligence 2010,* pp. 2302-2309, 2010

[42] B. Linares-Barranco and T. Serrano-Gotarredona, “Memristance can explain spike-timing-dependent-plasticity in neural synapses,” *Nature Proceedings,* 2009.

[43] A. M. Haas, T. Datta, A. Abashire and M. C. Peckerar, “Two transistor synapse with spike timing dependent plasticity,” http://hdl.handle.net/1903/8650, 2009

[44] Y. Chen, L. McDaid, S. Hall and P. Kelly, “A programmable facilitating synapse device.”, 2008 *International Joint Conference on Neural Networks, IJCNN 2008,*

Institute of Electrical and Electronics Engineers Inc pp1615-1620, 2008

[45] Y. Chen, S. Hall, L. McDaid, O. Buiu, and P. Kelly, “A silicon synapse based on a charge transfer device for spiking neural network application,” *Lecture Notes in Computer Science*, vol. 3973, J. Wang, Eds. Germany: Springer-Verlag, 2006.

[46] T. Dowrick, L. McDaid, S. Hall, O. Buiu and P. Kelly, “A biologically plausible neuron circuit.”, *International Joint Conference on Neural Networks, IJCNN 2007,* 2007

[47] Y. Chen, S. Hall, L. McDaid, O. Buiu, and P. Kelly, “On the design of a low power compact spiking neuron cell based on charge-coupled synapses,” *International Joint Conference on Neural Networks, IJCNN 2006*, 2006

[48] C. Diorio, S. Mahajan, P. Hasler, B. Minch and C. Mead, “ A high-resolution nonvolatile analog memory cell,” *IEEE International Symposium on Circuits and Systems*, vol. 3, pp. 2233-2236, 1995.

[49] M. E. Izhikevich, “Simple model of spiking neurons,” *IEEE Transactions on Neural Networks,* vol. 14, no. 6, pp. 1569-1572, 2003

[50] S. Furber and S. Temple, “ Neural systems engineering,” *J. R. Soc. Interface,* vol. 4, pp. 193-206, 2006.

[51] H. Markram, “The blue brain project,” *Nat Rev Neurosci. Vol. 7, pp. 153-160,* 2006.

[52] S. Druckmann, Y. Banitt, A. Gidon, F. Schurmann, H. Markam and I. Segey, “A Novel Multiple Objective Optimization Framework for Constraining Conductance-Based Neuron Models by Experimental Data,” *Frontiers in Neuroscience, vol. 1, no. 1, 2007*

[53] J. Kozloski, K. Sfyrakis, S. Hill, F. Schurmann, C. Peck and H. Markam, “Identifying, tabulating, and analyzing contacts between branched neuron morphologies,” *IBM Journal of Research and Development, Vol 52, Number ½, 2008*

[54] M. Hines, H. Eichner and F. Schurmann, “Neuron splitting in compute-bound parallel network simulations enables runtime scaling with twice as many processors,” *J. Comput. Neurosci., vol. 25, no. 1, pp. 203-10, 2008*

[55] M. Hines, F. Schurmann, and H. Markam, “Fully Implicit Parallel Simulation of Single Neurons,” *J. Comput. Neurosci., vol. 25, no. 3, pp. 439-48, 2008*

[56] S. Druckmann, T. K. Berger, S. Hill, F. Schurmann, H. Markam and I. Segey, “Evaluating automated parameter constraining procedures of neuron models by experimental and surrogate data,” *Biol Cybern.*, vol. 99, no. 4-5, pp. 371-9, 2008

[57] H. Anwar, I. Riachi, S. Hill, F. Schurmann and H. Markram, "Capturing neuron mophological diversity" *Computational modeling methods for neuroscientists*, *2009*

[58] J. King, M. Hinesm S. Hill, P. H. Goodman, H. Markram and F. Schurmann, "A Component-Based Extension Framework for Large-Scale Parallel Simulations in NEURON", *Frontiers in Neuroinformatics, available online, doi:10.3389/neuro.11.010.2009*

[59] E. L Bienenstock, L. N. Cooper, and P. W. Munro, "Theory for the development of neuron selectivity: Orientation specificity and binocular interaction in visual cortex",The Journal of Neuroscience, vol. 2, no. 1, pp. 32-48, 1982.

[60] P. Dayan and L. Abbott, *Theoretical Neuroscience*, MIT Press, 2001

[61] J. Harkin, F. Morgan, L. McDaid, S. Hall, B. McGinley, and S. Cawley, "A Reconfigurable and Biologically Inspired Paradigm for Computation Using Network-On-Chip and Spiking Neural Networks", International Journal of Reconfigurable Computing, vol. 2009, pp. 1-13, 2009.

[62] A. DeHon, R. Rubin, "Design of FPGA Interconnect for Multilevel Metallization", IEEE Transactions VLSI Systems, Vol. 12, No. 10, pp. 1038-1050, 2004

[63] G. Lemieux, D. Lewis, "Circuit Design of Routing Switches", ACM FPGA Conference, 2002

[64] G. Lemieux, D. Lewis, "Using Sparse Crossbars within LUT Clusters", ACM FPGA Conference, 2001

[65] A. Ye., J. Rose, "Using Bus-Based Connections to Improve Field-Programmable Gate-Array Density for Implementing Datapath Circuits", IEEE Transactions VLSI Systems, Vol. 14, No. 5, pp. 462-473, 2006

[66] C. Hilton, B. Nelson, "PnoC: A Flexible Circuit-Switched NoC for FPGA-based Systems", IEEE Computers and Digital Techniques, Vol. 153, Iss. 3, pp 181-188, 2006

[67] B. Gojman, R. Rubin, C. Pilotto, T. Tanamoto, "3D Nanowire-Based Programmable Logic", International Conference on Nano-Networks (Nanonets), September 14–16, 2006

[68] P. Merolla, J. Arthur, F. Akopyan1, N. Imam, R. Manohar, D. S. Modha1 "A Digital Neurosynaptic Core Using Embedded Crossbar Memory with 45pJ per Spike in 45nm" IEEE Custom Integrated Circuits Conference, 2011.

[69] J.A. Bailey, R. Wilcock, P.R. Wilson, J.E. Chad, "Behavioral simulation and

synthesis of biological neuron systems using synthesizable VHDL", Neurocomputing, vol. 74, pp. 2392–2406, 2011

[70] X. Jin, M. Luján, L.A. Plana, S. Davies, S. Temple, and S.B. Furber "Modeling Spiking Neural Networks on SpiNNaker" IEEE *Computing in Science Engineering,* vol. 12, no. 5, pp. 91-97, 2010

[71] D.B. Thomas and W. Luk, "PGA Accelerated Simulation of Biologically Plausible Spiking Neural Networks", 17th IEEE Symp. On Field Programmable Custom Computing Machines, 2009.

[72] J.M.J Murre, R. Griffioen, and I. H. Robertson, "Selfreparing Neural Networks: A Model for Recovery from Brain Damage" KES 2003, LNAI 2774, pp. 1164-1171, 2003.

[73] T. Schoenauer, S. Atasoy, N. Mehrtash, and H. Klar, "NeuroPipe-Chip: A Digital Neuro-Processor for Spiking Neural Networks" IEEE Transactions on Neural Networks, vol. 13, no. 1, pp. 205-213, 2002

[74] J.J Wade, L.J. McDaid, J.A. Santo and H.M. Sayers, "SWAT: An Unsupervised SNN Training Algorithm for Classification Problem", IJCNN, pp. 2648-2655, 2008.

[75] A. Upegui, Y. Thoma, H. F. Satizábal, F. Mondada, P. Rétornaz, Y. Graf, A. Perez-Uribe, and E. Sanchez, "Ubichip, Ubidule, and MarXbot: A Hardware Platform for the Simulation of Complex Systems", Evolvable Systems: From Biology to Hardware Lecture Notes in Computer Science, vol. 6274, pp. 286-298, 2010

[76] A. Upegui, Y. Thoma, E. Sanchez, A. Perez-Uribe, J. M. Moreno, J. Madrenas and G. Sassatelli , "The Perplexus Bio-Inspired Hardware Platform: A Flexible and Modular Approach", International Journal of Knowledge-based and Intelligent Engineering Systems - Adaptive Hardware / Evolvable Hardware, vol. 12, no. 3, pp. 201-212, 2008

Chapter 3 – Fundamentals of Semiconductor Physics

3.1 Introduction

This chapter is intended to outline the fundamentals of semiconductor physics which are relevant to the work presented within this thesis. The operation of the Metal-Oxide-Semiconductor capacitor, MOS-C, is described and the extraction of key device and process parameters from theoretical and experimental results are presented in section 3.2. Test devices were incorporated on 3 chip runs; each chip was fabricated in a 0.35µm process from Austria MicroSystems (AMS) and used for parameter extraction, in accordance with the theoretical analysis presented. Section 3.3 contains the theory for the dominant leakage current mechanism seen in SiO_2, namely Fowler-Nordheim (FN) tunnelling. The operation of the MOS transistor, including sub-threshold theory, is described in section 3.4. Conclusions of this chapter are presented in section 3.5.

3.2 MOS Capacitor

A fundamental building block of semiconductor devices is the MOS-C. It is fabricated with a thin layer of silicon oxide grown on top of a doped semiconductor substrate, with a metal contact made to the top. The metal contact can be replaced with a polysilicon electrode, referred to as the gate, as shown in Fig 3.1.

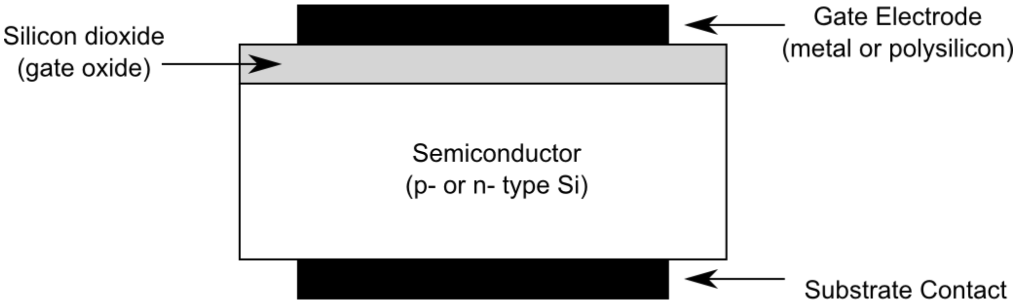


Fig 3.1 – Cross section of an MOS-C

3.2.1 Operation

The energy band diagram of a p-type MOS-C is presented in Fig. 3.2. For an ideal MOS-C the metal work function, Φ_m, is equal to the semiconductor work function Φ_{Si}, such that the work function difference Φ_{ms} is zero. This means that the Fermi level of the semiconductor, E_F, is aligned with that of the gate. There is no band bending within the device under this condition if it is assumed that the gate dielectric is free of any charge and the semiconductor is uniformly doped.

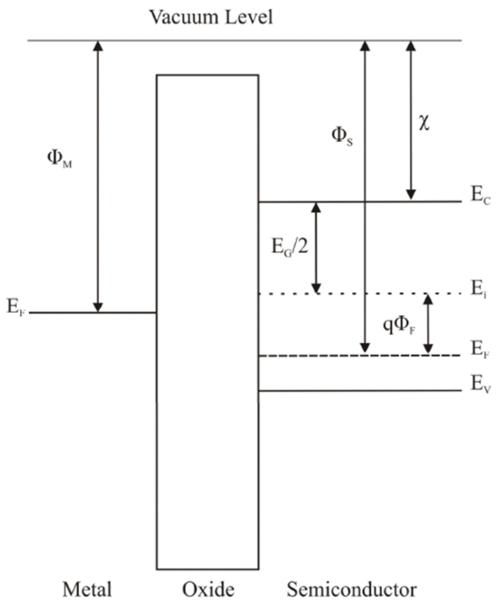


Fig. 3.2 – Energy band diagram of an ideal MOS-C

For an ideal MOS-C there are three distinct modes of operation; accumulation, depletion and inversion. These are represented by the charge distribution and energy band diagrams presented in Fig. 3.3 and Fig. 3.4 respectively.

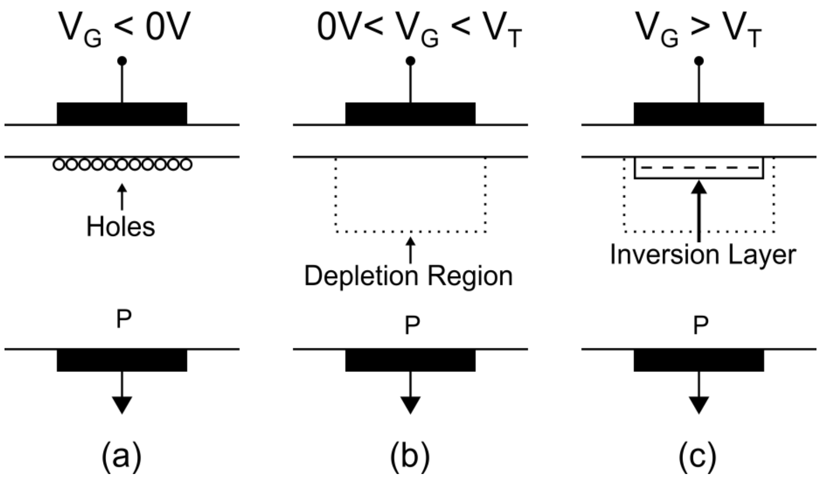
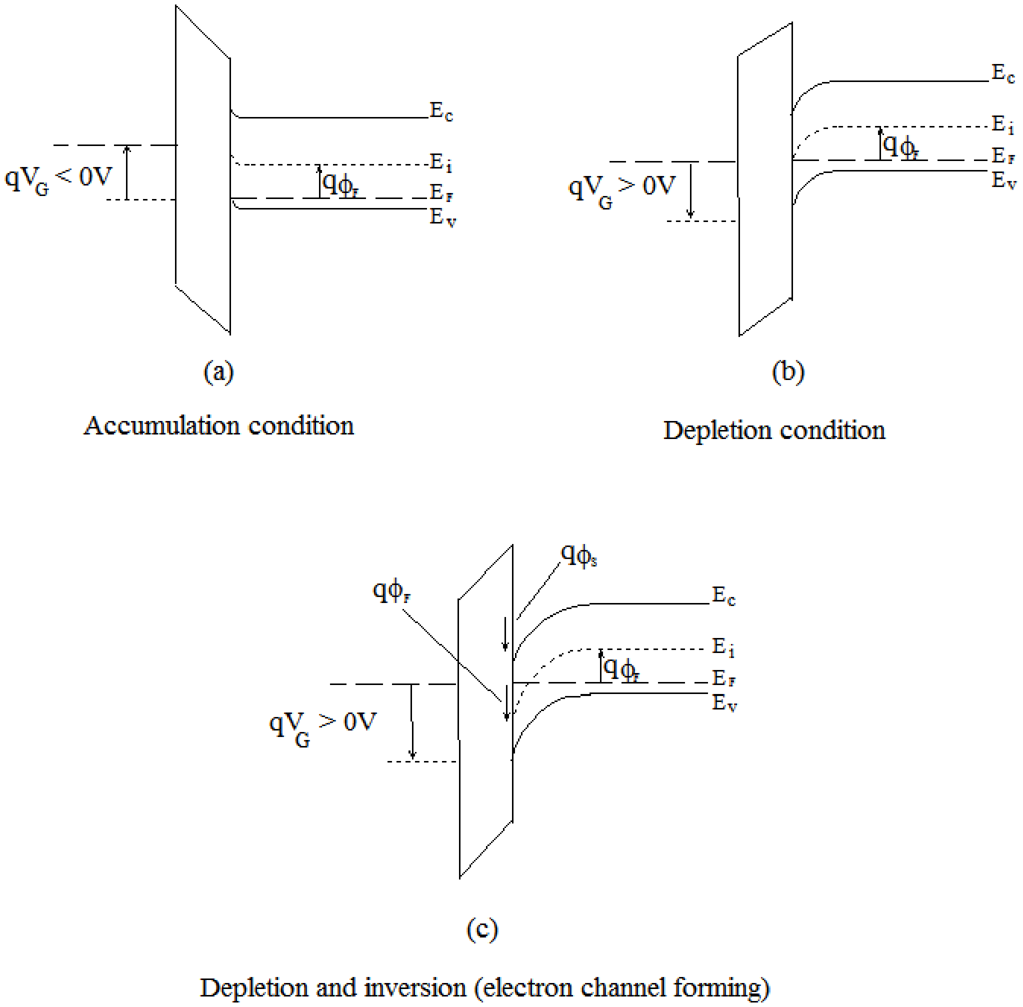


Fig. 3.3 – Charge distribution of p-type capacitor in (a) Accumulation (b) Depletion (c) Inversion



(a)

Accumulation condition

(b)

Depletion condition

(c)

Depletion and inversion (electron channel forming)

Fig. 3.4 – Energy band diagrams for the different operating modes of a p-type MOS-C

The device operates in accumulation when $V_G < 0V$. Due to this negative gate charge, the energy bands near to the semiconductor surface are bent upwards, Fig. 3.4(a), with positively charged holes forming an accumulation layer at the Si-SiO$_2$ interface, Fig. 3.3(a). Since the carrier density within the accumulation layer varies exponentially with the energy difference of the intrinsic Fermi energy and the Fermi energy, $E_i - E_F$ [1,2], the density of holes for the p-type MOS-C is given as;

$$p = n_i exp\left[\frac{E_i - E_F}{kT}\right] \tag{3.1}$$

If a small positive voltage, in the range 0 < V_G < V_T, is applied to the gate, the holes at the Si-SiO$_2$ interface are pushed away such that the accumulation layer is removed and a depletion region is formed, Fig. 3.3(b). The semiconductor provides the necessary negative charge via negative acceptor atoms to balance the positive charge on the gate. With the small positive gate bias the energy bands are bent downwards, Fig. 3.4(b), which results in a positive surface potential, ϕ_s;

$$\phi_s = \phi_F = \frac{1}{q}(E_i - E_F) = V_t ln\left(\frac{N_A}{n_i}\right) \tag{3.2}$$

V_t is the thermal voltage (25mV at 300K), N_A is the acceptor density in the substrate and n_i is the intrinsic concentration. The depletion charge per unit area, Q_d, can be given by;

$$Q_d = -qN_A W_d \tag{3.3}$$

Where W_d is the width of the depletion region underneath the gate;

$$W_d = \sqrt{\frac{2\varepsilon_{si}\varepsilon_0\phi_s}{qN_A}},\ 0 \leq \phi_s \leq \phi_F \tag{3.4}$$

The relationship between the gate voltage and the potential across the depletion region is given by equation 3.5.

$$V_G = V_{FB} + \phi_s - \frac{Q_d}{C_o} \tag{3.5}$$

Substituting 3.3 and 3.4 into 3.5 gives;

$$V_G = V_{FB} + \phi_s + \frac{\sqrt{2\varepsilon_{si}\varepsilon_0\phi_s}}{C_o} \tag{3.6}$$

Increasing the gate voltage causes the energy bands to bend further downwards such that the intrinsic Fermi level crosses over the semiconductor Fermi level, Fig.3.4(c). The electron concentration at the Si-SiO$_2$ interface increases and the surface begins to become n-type; an inversion layer is formed. The negative charge in the semiconductor now comprises of the ionized acceptor atoms in the depletion region and free electrons forming the inversion layer. At this point, $\phi_S = \phi_F$ the device is in weak inversion as the electron concentration is less than the hole concentration in the neutral bulk. As the gate voltage is increased, the energy bands are bent further, the depletion region increases up to its maximum width and the electron concentration increases in the inversion layer such it is greater than the hole concentration Fig. 3.3(c). E_i – E_F is now positive and the electron concentration at the surface is given by;

$$n = n_i exp\left[\frac{E_i - E_F}{kT}\right] \tag{3.7}$$

The onset of strong inversion is defined as the point when the surface potential is;

$$\phi_s = 2V_t ln\left(\frac{N_A}{n_i}\right) \tag{3.8}$$

Additionally the maximum width of the depletion region is

$$W_{dm} = \sqrt{\frac{4\varepsilon_{si}\varepsilon_0\phi_F}{qN_A}} \tag{3.9}$$

The applied voltage falls partially across the oxide and partially across the semiconductor;

$$V_G = V_o + \phi_s \tag{3.10}$$

Where V_o is the voltage across the oxide given by

$$V_o = \frac{|Q_d|}{C_o} \tag{3.11}$$

Hence the voltage at which inversion occurs, the threshold voltage, V_T is given by;

$$V_T = 2\phi_B + \frac{\sqrt{4\varepsilon_{si}\varepsilon_0\phi_F}}{C_o} \tag{3.12}$$

However there are additional factors which affect this ideal V_T; these include work function differences, oxide and interface charges. The amount by which V_T is shifted from the ideal value is defined as the flat band voltage, V_{FB}.

The work function of the gate, Φ_{gate}, will vary depending upon the material used; aluminum and n+ polysilicon have work functions of 4.1eV and 4.14eV respectively. The work function of the semiconductor, Φ_S, is dependent upon the doping concentration of the material, [1-3].

$$\Phi_S = \chi_s + \frac{E_g}{2} + V_t \ln\left(\frac{\mathrm{N_A}}{\mathrm{n_i}}\right) \tag{3.13}$$

χ_s is the electron affinity and is defined as the difference between the conduction band edge and the valence level, typically $\chi_s =$ 4.14eV. E_g is the semiconductor band gap; E_g = 1.1eV. For silicon the work function difference, $\Phi_{MS} = \Phi_{gate} - \Phi_S$ and is always negative, indicating that there is a reduction in V_T.

The presence of charge within the oxide and at the interface can also affect V_T. These charges occur due to intrinsic imperfections and also those that arise during fabrication. The oxide charge comprises of fixed charge, Q_f, near to the Si-SiO$_2$ interface, trapped charge within the oxide, Q_t, and mobile charge, Q_m.

Taking these two main factors into account gives the flat band voltage;

$$V_{FB} = \Phi_{MS} + \frac{Q_f + Q_t + Q_m}{C_o} \tag{3.14}$$

Hence the threshold voltage is (3.15) assuming there are no interface states (discussed later);

$$V_T = 2\phi_B + \frac{\sqrt{4\varepsilon_{si}\varepsilon_0\phi_F}}{C_o} + \Phi_{MS} + \frac{Q_f + Q_t + Q_m}{C_o} \tag{3.15}$$

These non-ideal effects in the oxide are considered in more detail in section 3.2.3, following a description of the CV method.

3.2.2 C-V Characterization

The capacitance-voltage, C-V, plot of an MOS-C can be used to extract a number of physical parameters for the device. A 100µm x 100µm p-type MOS-C was fabricated in 0.35µm AMS process. A high frequency CV analysis is undertaken with a small ac signal superimposed on to the swept, dc gate bias. The gate voltage, V_g is swept from -4V to 4V at a rate of 0.05V/sec with a superimposed ac signal of frequency 1MHz. The measured C-V plot is presented in Fig. 3.5 where the maximum and minimum values for the gate oxide capacitances are 44.66pF and 11.1pF. A return sweep form 4V to -4V was undertaken and no hysteresis was observed, indicating negligible mobile charge in the oxide.

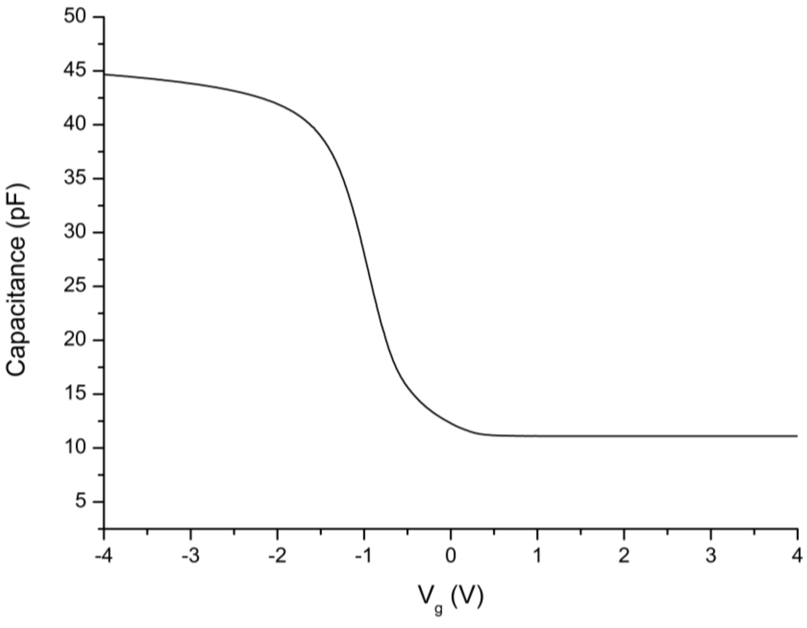


Fig. 3.5 – CV Plot for p-type MOS-C

While the device is in accumulation, the charge within the accumulation layer is modulated by the applied signal. Since the charge is formed by majority carriers (holes) these can react easily to the signal. Hence the MOS-C can be represented by two capacitors in series, one for the oxide capacitance, C_{ox}, and one for the accumulation layer capacitance, C_{acc}.

$$\frac{1}{C_{measured}} = \frac{1}{C_{ox}} + \frac{1}{C_{acc}} \tag{3.16}$$

$$C_{acc} = \frac{dQ_{acc}}{d\phi_s} \tag{3.17}$$

Since C_{acc} is large, the value of C_{ox} is dominant, therefore $C_{max} \approx C_{ox}$, and from this, the oxide thickness can be calculated from equation 3.18.

$$t_{ox} = \frac{\varepsilon_{ox}\varepsilon_0 A_C}{C_{max}} \tag{3.18}$$

where A_C is the area of the capacitor (100µm x 100µm). As V_g is reduced, the accumulation layer is reduced until the device reaches the flat-band condition, ideally when $V_g = 0$. Under this condition, the capacitance of the semiconductor is defined by the Debye length, equations 3.19, 3.20.

$$C_D = \frac{\varepsilon_{si}\varepsilon_0}{L_D} \tag{3.19}$$

$$L_D = \sqrt{\frac{\varepsilon_{si}\varepsilon_0 V_t}{qN_A}} \tag{3.20}$$

In the inversion region, the measured minimum capacitance is due to the oxide capacitance and the capacitance of the depletion region. This is because the depletion width will no longer increase as the dc component of the charge present on the gate is balanced by the charge in the inversion layer.

$$\frac{1}{C_{min}} = \frac{1}{C_{ox}} + \frac{1}{A_c C_d} \tag{3.21}$$

Where C_d is the depletion capacitance per unit area:

$$C_d = \frac{\varepsilon_{si}\varepsilon_0}{W_{dm}} = \sqrt{\frac{\varepsilon_{si}\varepsilon_0 q N_A}{2\phi_s}} \tag{3.22}$$

Hence the doping density can be found with the transcendental equation 3.23, with an initial guess for N_A in the natural log term;

$$N_A = \frac{4}{{A_c}^2}\frac{kT}{q^2}\frac{1}{\varepsilon_{si}\varepsilon_0}\left[\frac{1}{C_{min}} - \frac{1}{C_{ox}}\right]^{-2} \ln\left(\frac{\mathrm{N_A}}{\mathrm{n_i}}\right) \tag{3.23}$$

Alternatively N_A can also be found using [3];

$$\log N_A = 30.38759 + 1.68278 \log C_1 - 0.03177[\log C_1]^2 \qquad (3.24)$$

Where;

$$C_1 = \frac{RC_{ox}}{A_C(1-R)} \qquad (3.25)$$

$$R = \frac{C_{min}}{C_{ox}} \qquad (3.26)$$

3.2.3 Non-ideal Effects

In practice, the oxide will contain defects which cause unwanted behaviour in operation. These defects include; hole and electron traps, mobile ions, positive fixed charge and interface states, as depicted in Fig 3.6.

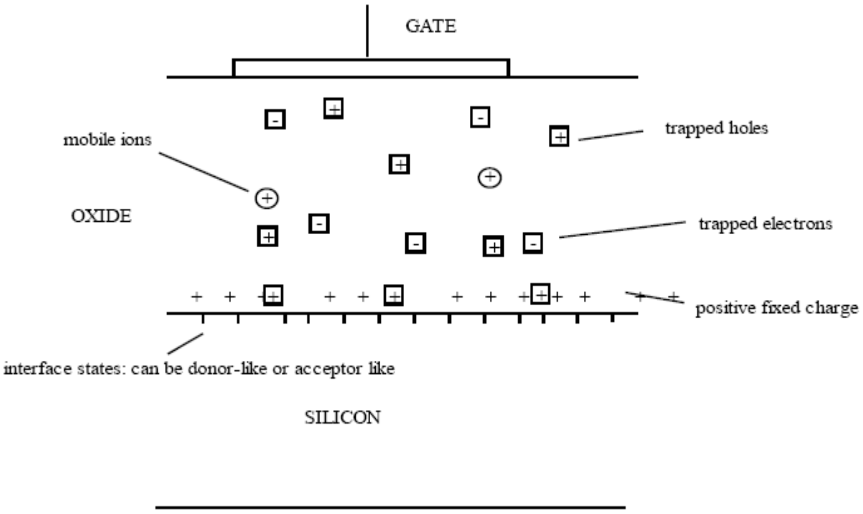


Fig. 3.6 MOS-C oxide charge components

The oxide charge induces image charges in the silicon and on the gate, which affect the threshold voltage of the device. The oxide charges are not dependent upon gate bias and as such cause a parallel shift of the C-V plot, depending upon the polarity of the charge. No distortion of the C-V curve occurs. The closer the charge is to the oxide-semiconductor surface the greater the shift will be. Positive charge is equivalent to increasing the gate bias, shifting the C-V curve to the left of the ideal C-V curve. Negative charge is equivalent to a reduction of the gate bias and will shift the C-V curve right, Fig. 3.7.

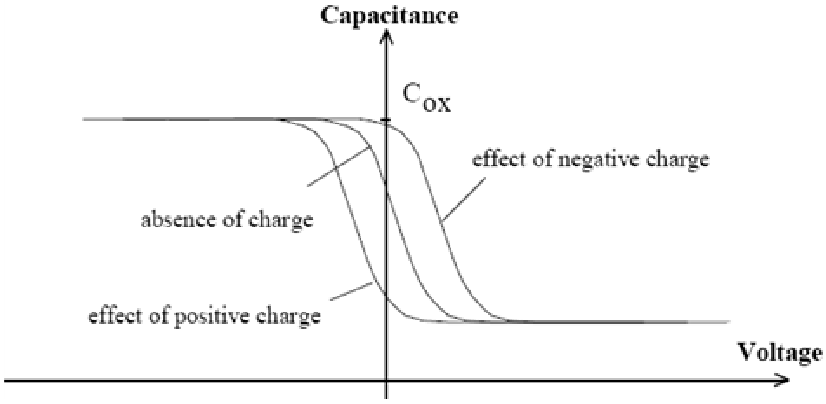


Fig. 3.7 Effects of Oxide Charge on p-type MOS-C

The positive fixed oxide charge, Q_f, is positioned very close to the oxide-semiconductor interface and is generated during the fabrication process and as such the density of Q_f is not affected by oxide thickness t_{ox} or silicon impurity concentration, but rather is dependent on the oxidation, annealing conditions and silicon surface orientation.

The fixed oxide surface charge can be defined as $Q_f = qN_f$, where N_f is the number of charges per unit area. N_f is given as;

$$N_f = \frac{C_{ox}\Delta V}{q} = \frac{C_{max}\Delta V}{qA_c} \tag{3.27}$$

ΔV is the shift of the C-V plot with respect to the mid-gap.

Mobile ionic charges in the oxide can move back and forth with applied voltage. They become more mobile at elevated temperatures. These charges, typically due to sodium or potassium (positive) ions, occur because of poor quality control of chemicals in the manufacturing process. The voltage shift, ΔV_m caused by the mobile charges is;

$$\Delta V_m = -\frac{Q_m}{C_{ox}} \tag{3.28}$$

Mobile ionic charges cause a hysteresis (the d.c. voltage is swept negative to positive and then positive to negative), in the C-V plot. The hysteresis occurs as Q_m moves slower within the structure as its transport is dependent upon the applied electric field and temperature. When a negative voltage is applied the ions are drawn towards the gate. As V_g is made positive, the ions are pushed towards the interface, reducing V_T. The return sweep, positive to negative, shows that a lower V_T exists and causes a hysteresis, Fig. 3.8.

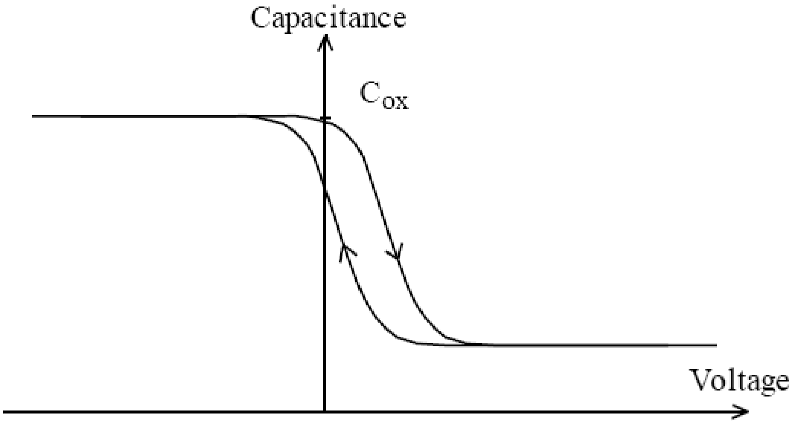


Fig. 3.8 Hysteresis C-V plot for p-type MOS-C

Hysteresis in the opposite direction, (a positive to negative sweep followed by negative to positive sweep) is not due to the mobile ion charge, but rather it is due to slow trapping. Trapped oxide charge are made up of both electron and hole traps within the oxide. The traps are initially neutral and are charged by the introduction of either holes or electrons into the oxide. They are caused by current passing through the oxide, hot carrier injection or by photon excitation. Trapped oxide charges also cause a shift, ΔV_t in the C-V plot.

$$\Delta V_t = -\frac{Q_t}{C_{ox}} \tag{3.29}$$

Interface states (also known as interface traps, fast states or surface states) occur as a result of the transition from single crystal semiconductor to the amorphous oxide, they are energy levels within the energy band gap, (they have the ability to trap charge, Q_{it},). The occupancy of these states with electrons depends upon the Fermi level and thus is bias dependent in the capacitor. These states can be;

Donor like: neutral when occupied with electrons and positive when empty

Acceptor like: negative when occupied with electrons and neutral when empty

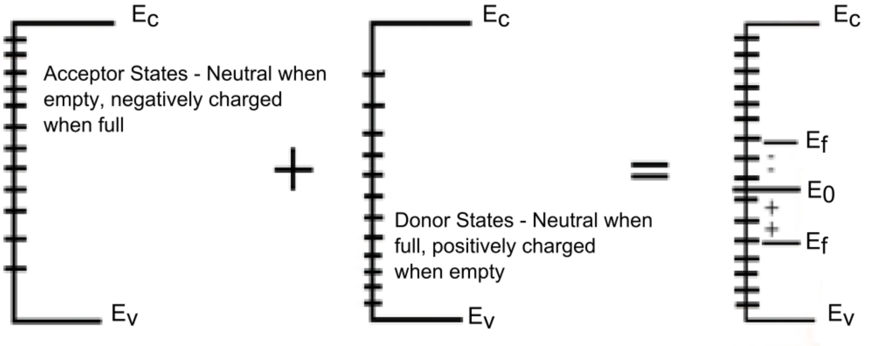

Ec
Acceptor States - Neutral when empty, negatively charged when full
Ev
+
Ec
Donor States - Neutral when full, positively charged when empty
Ev
=
Ec
Ef
E0
Ef
Ev

Fig. 3.9 MOS-C Interface States [2]

Fig. 3.9 presents a representation of the interface states within the band gap of an MOS structure. The generally accepted model is that the upper half of the energy band gap is acceptor-like while the bottom half is donor-like [2]. When in accumulation (assuming a p-type device), the interface states are below the Fermi level and are occupied by electrons. As V_g is increased and the semiconductor becomes depleted, the Fermi level will begin to move downwards through the energy band gap. This has the effect of emptying the interface states of electrons, making them positively charged, causing stretching/smearing in the C-V plot, Fig. 3.11.

It can become difficult to distinguish between the effects of the interface states and the positive fixed oxide charge on the C-V plot, as both cause similar types of shifts. To aid in the analysis of the interface states, the ac equivalent circuits are considered at low frequency. At low frequency there is enough time for the occupancy of the states to follow the ac signal which is applied to the structure. In addition to this, the shift in mid-gap and flat band voltages (from their ideal values) can be calculated.

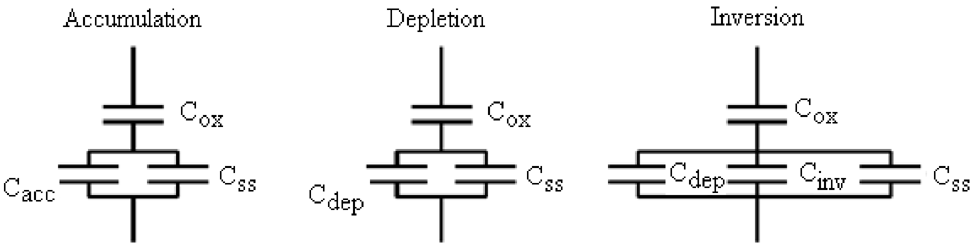


Fig. 3.10 AC equivalent circuits for each region showing the capacitive effects of the surface states

The capacitance of the MOS capacitor, including the surface state capacitance, C_{SS}, is;

$$\frac{1}{C} = \frac{1}{C_{ox}} + \frac{1}{C_d + C_{SS}} \tag{3.30}$$

The number of surface states per unit area per eV is;

$$N_{SS} = \frac{C_{SS}}{q} m^{-2} eV^{-1} \tag{3.31}$$

Fig. 3.11 shows the effects of the surfaces states on the C-V plot compared to the ideal C-V plot.

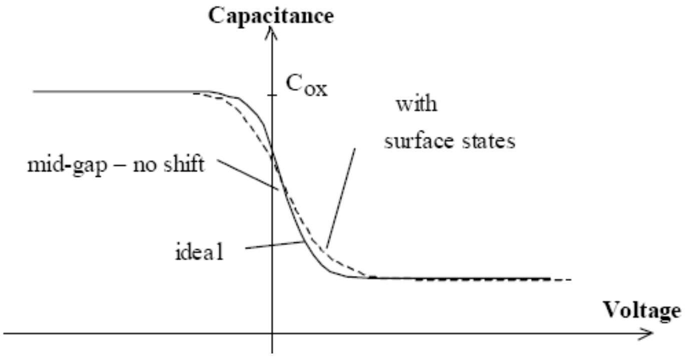


Fig. 3.11 Effect of surface states; donor like in lower half of the band gap and acceptor like in the top half

3.2.4 Experimental Results

Table 3.1 presents typical extracted values from the C-V plot of Figure 3.5; the oxide thickness, t_{ox}, gate oxide capacitance per unit area, C_o, and semiconductor doping density, N_A. The measured values are within 1% for t_{ox} and C_o and within 20% for N_A, which is consistent with process variations described by maximum and minimum values in the AMS process documentation [4]. Table 3.2 presents additional measured device characteristics using the equations presented earlier in this chapter.

	Measured	**Typical AMS**
t_{ox}	7.7nm	7.6nm
C_o	4.47 x 10^{-7} Fcm^{-2}	4.54 x 10^{-7} Fcm^{-2}
N_A	2.58 x 10^{17} cm^{-3}	2.12 x 10^{17} cm^{-3}

Table 3.1 – Extracted values and equivalent typical AMS values (where applicable) for p-type MOS-C

	Measured	**Units**
C_{max}	44.7	pF
C_{min} (HF plot)	11.1	pF
ϕ_S	0.83	V
W_d	64	nm
W_{dm}	91	nm
$\lvert Q_d \rvert$	2.64 x10^{-7}	Ccm^{-2}
Φ_S	5.11	eV
Φ_{gate}	4.14	eV
Φ_{MS}	-0.97	eV
V_T (ideal)	0.83	V
V_{FB}	2.09	V
C_{FB}	33.12	pF
L_d	7.9	nm
$V_{mg\ (ideal)}$	-0.167	V
$V_{mg\ (measured)}$	-0.141	V
ΔV_{mg}	0.026	V
N_f	7.26 x10^{10}	cm^{-2}
N_{ss}	1.56 x10^{11}	cm^{-2}eV^{-1}

Table 3.2 – Extracted values from C-V plot for p-type MOS-C

3.2.4.1 Deep Depletion

Deep depletion occurs in an MOSC when the gate voltage is swept quickly, say 0.5V/sec such that the device is driven out of thermal equilibrium whilst measuring the high frequency capacitance. Deep depletion occurs because the inversion layer does not have time to form as the gate voltage is ramped up past the threshold voltage. In this situation the depletion charge must satisfy the neutrality condition on both plates of the MOSC. The depletion region increases past its maximum thermal equilibrium value and as a result, the measured capacitance continues to reduce with increasing gate voltage. The condition is illustrated in Fig. 3.12. The MOS-C is restored to equilibrium through thermal generation of electron-hole pairs in the depletion region, since the supply of minority carriers from the substrate is very small.

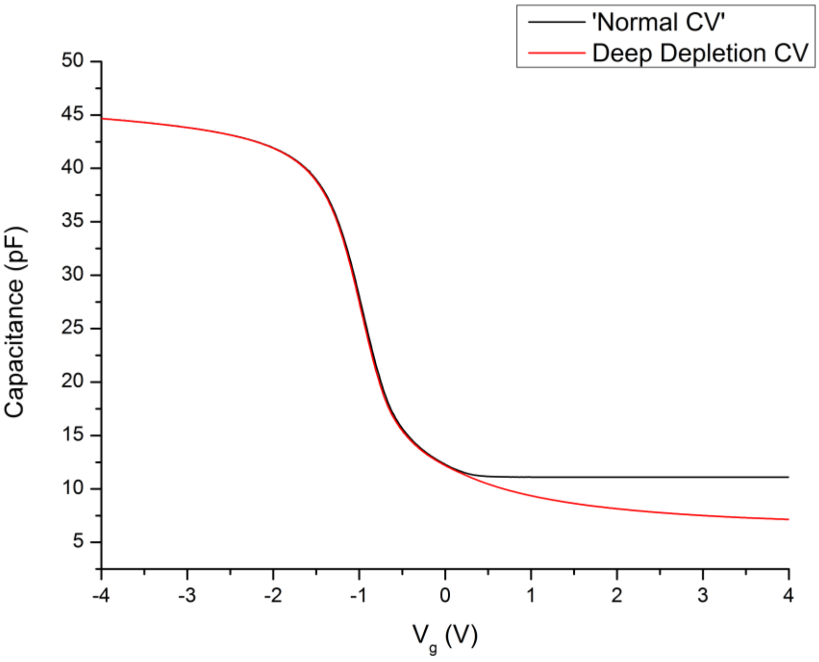


Fig. 3.12 – CV Plot showing deep depletion for p-type MOS-C

The capacitance of the device while it is in deep depletion can be given by equation 3.32, where C_S is the change in capacitance as the device is forced into deep depletion.

$$C = \frac{C_{ox}C_s}{C_{ox}+C_s} \tag{3.32}$$

where;

$$C_s(t) = \frac{C_{ox}}{\sqrt{\left[1+2\left(V_G - V_{FB} + \left(\frac{Q_{inv}(t)}{C_{ox}}\right)\right)/V_0\right]}} \tag{3.33}$$

$$V_0 = \frac{q\varepsilon_{si}\varepsilon_0 N_A}{C_{ox}} \tag{3.34}$$

Solving and differentiating equation 3.32 with respect to time, t gives;

$$\frac{dV_g}{dt} = -\frac{1}{C_{ox}}\frac{dQ_{inv}}{dt} - \frac{q\varepsilon_{si}\varepsilon_0 N_A}{{C_s}^3}\frac{dC_s}{dt} \tag{3.35}$$

The thermal generation rate can be calculated by equation 3.36 since V_g is constant for the pulse MOS-C, hence $\frac{dV_g}{dt} = 0$.

$$\frac{dQ_{inv}}{dt} = \frac{q\varepsilon_{si}\varepsilon_0 N_A}{{C_s}^3}\frac{dC_s}{dt} \tag{3.36}$$

3.3 Fowler-Nordheim Tunnelling

FN tunnelling leakage is observed when there is a sufficiently high oxide electric field, typically ≥5MV/cm. Electrons, depending on whether the applied voltage to the gate is positive or negative respectively, can tunnel from the semiconductor through the oxide onto the gate or from the gate back into the semiconductor. The process is shown in Fig.3.13 for the case of tunnelling from an accumulation layer; ϕ represents the potential barrier height for electron emission

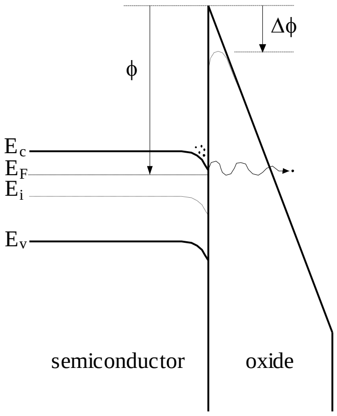


Fig. 3.13 – FN Tunnelling

Considering the case for tunnelling from a metal electrode, if the potential barrier is triangular, then the FN current density, J_{FN} is given by [5];

$$J_{FN} = CE_{ox}^2 exp\left(\frac{-\beta}{E_{ox}}\right) \tag{3.37}$$

Constant C and β are given by equations (3.38) and (3.39) respectively:

$$C = \frac{q^3 m_o}{8\pi h m_{ox}\phi} = 1.54x10^{-6}\frac{m_o}{m_{ox}}\frac{1}{\phi}\ A/V^2 \tag{3.38}$$

$$\beta = \frac{8\pi}{3}\frac{(2m_{ox})^{1/2}}{qh}\phi^{3/2} = 6.83x10^7\sqrt{\frac{m_{ox}}{m_o}}\phi^3\ V/cm \tag{3.39}$$

Where q is the electronic charge, m_o is the mass of an electron at rest, m_{ox} is the effective mass of an electron in the insulator and h is Plank’s constant.

Typically, $\ln(J_{FN}/E_{ox}^2)$ is plotted versus $(1/E_{ox}^2)$, and the barrier height can be extracted for the gradient. Fig. 3.14 shows such a FN plot for a barrier height of ϕ =3.1eV and m_{ox}~0.5 m_o. E_{ox} is given by;

$$E_{ox} = \frac{V_G - V_{FB} - \phi_S}{t_{ox}} \tag{3.40}$$

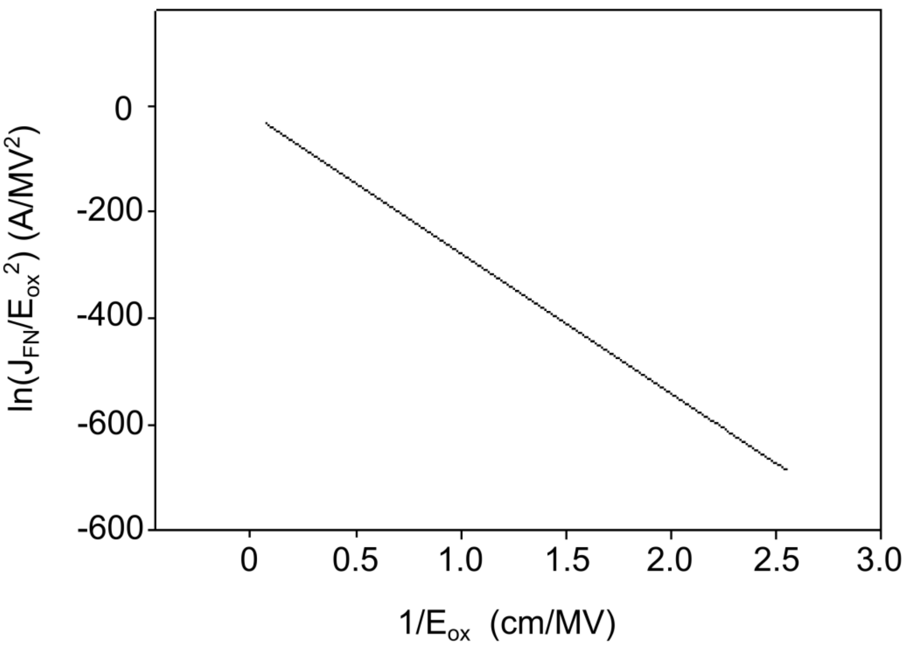


Fig. 3.14 – Theoretical F-N Plot

A linear relationship between $\ln(J_{FN}/E_{ox}^{2})$ and $1/E_{ox}$, and the expected value for barrier height, indicates the presence of FN tunnelling. The gradient of the line is B and the intercept on the y-axis is A. From Fig. 3.14 A and B are 9.94×10^{-7} A/V^2 and 2.64×10^{8} V/cm respectively. For FN tunnelling to occur, an electric field in the region of 6MV/cm to 10MV/cm is required, above this range and the silicon dioxide may break down. Extraction of a barrier height of the right order, namely 3.1eV for the SiO_2 system provides a strong indication for FN-tunnelling. A polarity dependence is also expected, in line with the barrier heights expected from energy band diagram for the system. FN tunnelling is an undesirable effect in MOS transistors as it gives rise to leakage current in the gate oxide, however, it is the basis for the operation of semiconductor memory devices. This will be looked at in more detail in chapter 4.

3.4 MOS Transistor

The structure of a n-channel, NMOS, is presented in Fig. 3.15. The structure of the NMOS differs from that of the MOS-C in that two n^+ regions, the source and drain are implanted into the p-substrate. If $V_{gs} > V_T$, then an inversion layer is formed between the drain and source within the p-type semiconductor. This is the channel and allows a current to flow, provided there is a positive voltage difference between the source and drain. If the gate to source voltage, V_{gs}, is 0, and the drain voltage is V_{ds}, then no channel exists. In this situation V_{ds} falls across the drain junction which is reverse biased by this voltage.

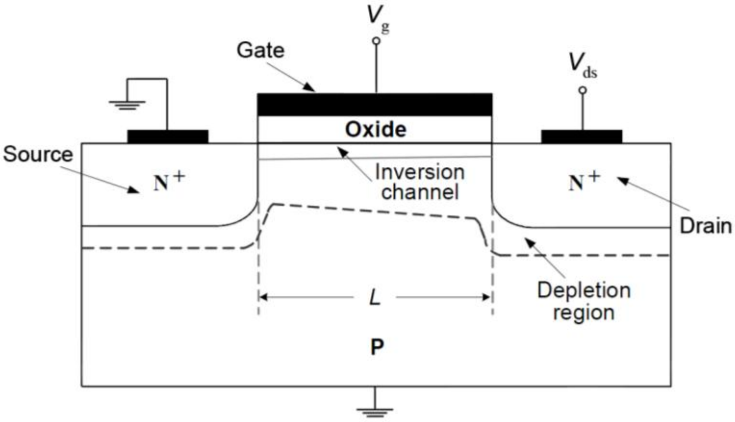


Fig. 3.15 – Cross-section of NMOS operating in the linear region

3.4.1 MOST Operation

Prior to deriving the operating characteristics of the NMOS several assumptions are made; the gate structure is an ideal MOS-C, there are no interface states, no oxide charge. The carrier mobility is constant in the channel and the substrate doping is uniform. The current is dominated by drift with the transverse electric field larger than the longitudinal electric field.

To drive an equation for the drain current it is assumed that initially the inversion layer thickness can be represented as a delta function. The surface potential is given by equation 3.41, where V(y) is the electron quasi-Fermi potential along the channel with respect to the Fermi potential of the source.

$$\phi_s = 2\phi_B + V(y) \tag{3.41}$$

The depletion charge is thus;

$$Q_d = -qN_AW_{dm} = -\sqrt{2q\varepsilon_{si}\varepsilon_0N_A\left(2\phi_B + V(y)\right)} \tag{3.42}$$

The total charge in the inversion layer is;

$$Q_{inv} = Q_{si} - Q_d$$

$$Q_{inv} = -C_0\left(V_{gs} - 2\phi_B - V(y)\right)q + \sqrt{2q\varepsilon_{si}\varepsilon_0 N_A\left(2\phi_B + V(y)\right)} \qquad (3.43)$$

The drain current, I$_D$, with respect to the inversion layer charge, channel length, L, and width, W, is;

$$\int_0^L I_D = \frac{W}{L}\mu\int_0^{V_{ds}} -Q_{inv}\left(V(y)\right)dy \qquad (3.44)$$

Where µ is the average electron mobility in the channel. Substituting equation 3.43 into 3.44 and integrating gives the drain current.

$$I_D = \frac{\mu C_0 W}{L}\left[\left(V_{gs} - 2\phi_B - \frac{V_{ds}}{2} - V_{FB}\right)V_{ds} - 2\,\frac{\sqrt{2q\varepsilon_{si}\varepsilon_0 N_A}}{3C_0}\left[\left(2\phi_B + V_{ds}\right)^{\frac{3}{2}} - \left(2\phi_B\right)^{\frac{3}{2}}\right]\right] \qquad (3.45)$$

Consider the case when V$_{ds}$ is small, V$_{ds}$ < (V$_{gs}$ - V$_T$). In this case the NMOS is operating in the unsaturated, linear region, and I$_D$ can be simplified to;

$$I_D = \frac{\mu C_0 W}{L}\left[\left(V_{gs} - 2\phi_B - \frac{V_{ds}}{2} - V_{FB}\right)V_{ds}\right] \qquad (3.46)$$

If V$_{ds}$ << (V$_{gs}$ - V$_T$) then I$_D$ simplifies further to;

$$I_D \sim \frac{\mu C_0 W}{L}\left[\left(V_{gs} - V_T\right)V_{ds}\right] \qquad (3.47)$$

Additionally $\frac{\mu C_0 W}{L}$ is referred to as β.

If V_{ds} = (V_{gs} - V_T) then the charge in the inversion layer is zero at y = L_{eff}. L_{eff} is defined as the effective electrical channel length, as known as the pinch-off point. At this point the NMOS is operating in saturation and the drain current is can be given as;

$$I_D = \frac{\mu C_0 W}{2L_{eff}}\left[\left(V_{gs} - V_T\right)^2(1 + \lambda V_{ds})\right] \tag{3.48}$$

Above the pinch-off point the drain current will saturate with any further increase in V_{ds} having little effect. In short channel devices I_D does have a dependence on V_{ds} as V_{ds} has a dependence on L_{eff}. In this situation the depletion region which is associated with the n+ region will expand into the channel with increasing V_{ds}. The right hand term in equation 3.58 models the reduction in L_{eff} above pinch-off, where λ is the channel-length modulation factor which is typically a constant value.

3.4.2 I-V Characterisation

The I-V characteristics of the NMOS (or PMOS) transistor can be used to extract various device parameters. Figs. 3.16 show the I_D v V_{GS} characteristics of an NMOS device for a chosen V_{DS} of 3V. The threshold voltage of the NMOS transistor can be estimated from Fig. 3.16 while the device is operating in the linear region.

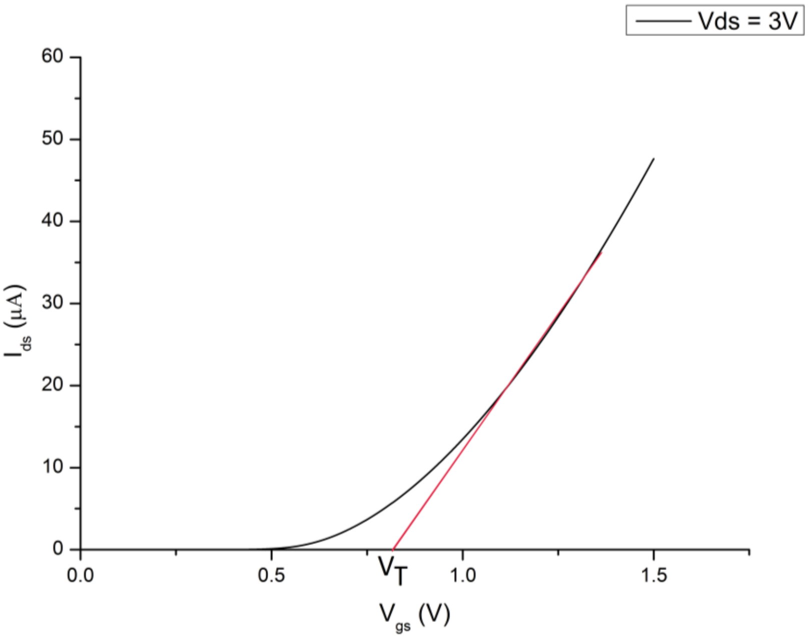


Fig. 3.16 – NMOS I_D v V_{gs} Plot

Fig. 3.17 shows that the slope of the I_d v V_{gs} curve, while the MOSFET is in saturation, can be used to find β. From this the low field value of μ can be calculated from;

$$\mu = \frac{\beta L}{W C_0} \tag{3.49}$$

The high field mobility, μ_{eff}, can be found from [6], where $U_a = 4.7 \times 10^{-10}$m/V and $U_b = 1.47 \times 10^{-18}$m/V^2. The minimum mobility value when V_{gs} is $0.83\mu_0$.

$$\mu_{eff} = \frac{\mu_0}{1+U_a\left(\frac{Vgs+V_T}{tox}\right)+U_b\left(\frac{Vgs+V_T}{tox}\right)^2} \tag{3.50}$$

Fig. 3.17 presents the output characteristics for an NMOS transistor for various values of V_{GS}.

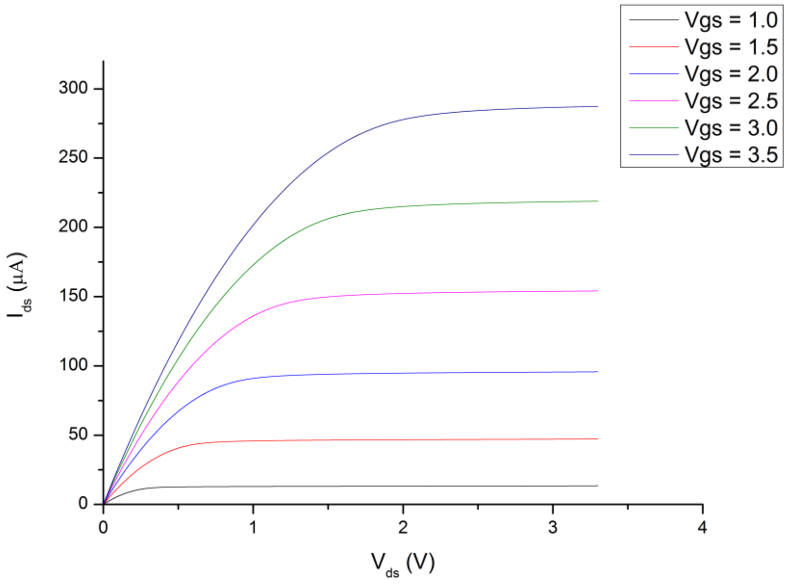


Fig. 3.17 – NMOS Output Characteristics, I_D v V_{ds} for V_{gs} = 1.0V, 1.5V, 2.0V, 2.5V, 3.0V and 3.5V. W = 100µm, L= 100µm.

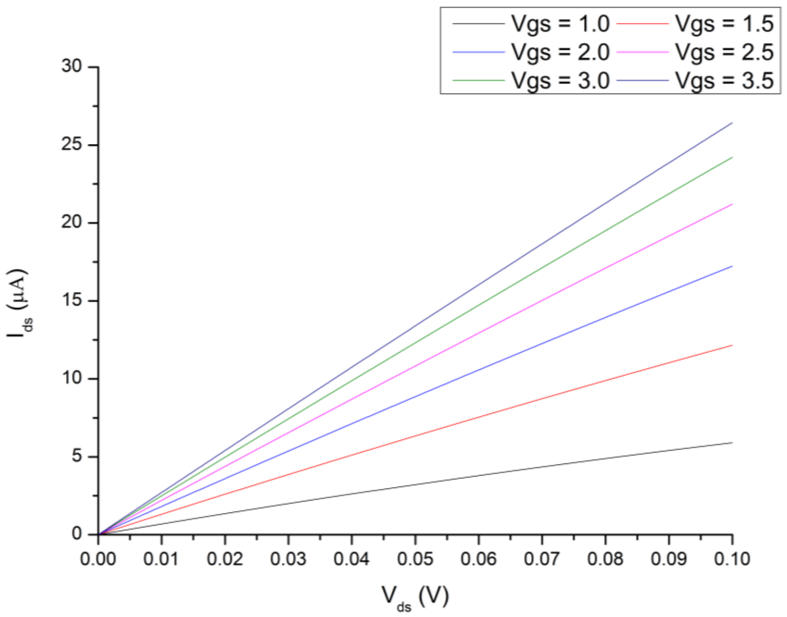


Fig. 3.18 – NMOST Output Characteristics, I_D v V_{ds} showing linear region for Vgs = 1.0V, 1.5V, 2.0V, 2.5V, 3.0V and 3.5V

Since these are devices are long channel devices, L = 100µm, the effect of λ is negligible. The total source to drain resistance, R_0 and channel resistance, R_{CH} can be found from equations 3.51 and 3.52 respectively.

$$R_0 = R_S + R_D + R_{CH} \tag{3.51}$$

$$R_{CH} = \frac{1}{\beta(V_{gs} - V_T)} \tag{3.52}$$

The source-drain resistances, R_S+R_D can be determined from the plot of R_0 against $\frac{1}{(V_{gs} - V_T)}$, where the intercept on the y-axis gives a value for the source and drain resistances, R_S+R_D, and β is the gradient of the line.

3.4.3 Sub-threshold Operation

When V_{gs} is below V_T the NMOS is said to be operating in subthreshold. In this situation it is the diffusion current rather than the drift current which dominates, causing the drain current to be exponentially dependent upon V_{gs}.

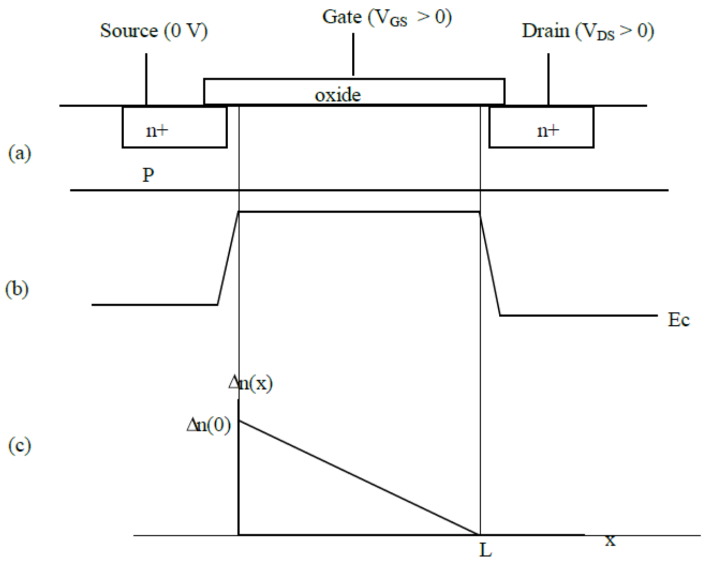


Fig. 3.19 – (a) Cross section of NMOS (b) conduction band from source to drain (c) concentration of electrons along the channel

The gate voltage induces a small positive surface potential, ϕ_s, along the length of the channel. This causes the n^+-p junction at the source to become forward biased, hence injecting electrons to form a weak channel. The channel charge has an insignificant effect on the electrostatics under the gate, which is determined predominantly by depletion charge. The injected electrons diffuse to the reverse biased drain where they are collected and give rise to a current given as:

$$I_{DsubVT} = I_0 exp\left(\frac{qV_{gs}}{mkT}\right)\left[1 - exp\left(\frac{qV_{ds}}{kT}\right)\right] \tag{3.53}$$

I_0 is the off-current and the gate-channel coupling coefficient is m:

$$m = 1 + \frac{C_d}{C_0} \tag{3.54}$$

$$I_0 = \mu_{eff} C_0 \frac{W}{L}(m-1)\left[\frac{kT}{q}\right]^2 \tag{3.55}$$

For V_{ds} > 3kT/q equation 3.53 reduces to

$$I_{DsubVT} = I_0 exp\left(\frac{qV_{gs}}{mkT}\right) \tag{3.56}$$

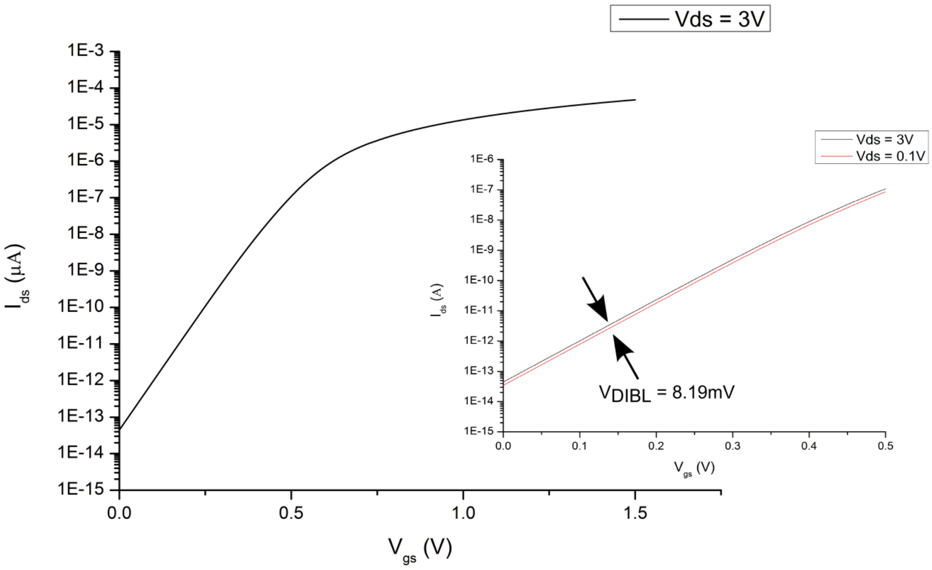


Fig. 3.20 – NMOS I_D v V_{GS} Plot. Insert showing DIBL effect

Fig. 3.20 presents the sub-threshold plot for an NMOS device. I_o can be found by extrapolating to the y axis, I_o = 44.6fA. The slope of the subthreshold plot, S, can be used to find m using equation 2.57. S = 89.9 mV/decade, m = 1.56.

$$S = m\frac{kT}{q}\ln(10)\, mV/decade \tag{3.57}$$

N_{SS} can be estimated since the presence of surface states will increase the value of m based upon equations 3.57, 3.58. Typically N_{SS} are low, around 10^{10} cm^{-2} [2], from equation 3.59 $N_{SS} = 4.3\text{x}10^{11}$ cm^{-2}.

$$m = 1 + \frac{C_d}{C_0} + \frac{C_{SS}}{C_0} \tag{3.58}$$

$$N_{SS} = \frac{C_{SS}}{q} \tag{3.59}$$

The insert in Fig. 3.20 shows the effect of drain induced barrier lowering (DIBL) for V_{ds} = 0.1V and V_{ds} = 3V, a shift of approximately 8.19mV occurs. DIBL occurs as the effective channel length decreases with increasing drain voltage. Thus less charge is supported by the gate voltage and this has the effect of reducing the threshold voltage. Additionally DIBL causes an increase in I_0 and static power consumption.

3.5 Conclusions

An overview of MOS semiconductor physics has been presented with particular focus on the MOS capacitor and transistors. The basic operation and extraction of device parameters from typical CV and IV characteristics has been undertaken using fabricated devices from 3 chip runs, and simulations. Simulation results of the back annotated layout (including the effects of parasitic capacitances) were obtained using Cadence SpectreS SPICE simulator. While the characteristics of the ideal device has been the main focus of this chapter, non-ideal effects and particularly FN tunnelling has also been considered. FN tunnelling is one of the main methods used to store charge on a floating gate non-volatile memory, FGNVM device.

3.6 References

[1] A. S. Grove, *Physics and Technology of Semiconductor Devices*, Wiley International, 1967

[2] S. M. Sze, *Semiconductor Devices: Physics and Technology 2nd Edition*, John Wiley & Sons, 2001

[3] D. K. Schroder, *Semiconductor Material and Device Characterization 2nd Edition*, John Wiley & Sons, 1998

[4] Austria MicroSystems, “0.35µm CMOS C35 Process Parameters”, 1997

[5] Z. A. Wienberg, *“On tunneling in metal-oxide-silicon structures”*, Journal of Applied Physics, vol. 53, no. 7, pp. 5052-5056, 1962

[6] M. C. Yuhua Cheng, Kelvin Hui, Min-chie Jeng, J. H. Zhinhong Liu, Kai Chen, James Chen, Robert Tu and C. H. Ping K. Ko, *BSIMS3v3 Manual*, University of California, Berkeley, 1996

Chapter 4 – Synaptic Weight Storage in Analogue Hardware Neural Networks

4.1 Introduction

The dominant processing element within the brain is the synapse. A synapse forms an electrochemical junction between the axon of a presynaptic neuron and the dendrite of a postsynaptic neuron. It is responsible for information processing, learning and adaption within a neural network. This is done through the storage and modification of the synaptic weight, known as synaptic plasticity. The synaptic weight dictates the effect that the presynaptic neuron has upon the adjoining postsynaptic neuron. In hardware neural networks there has been considerable focus on how to represent and store synaptic weight for use with silicon synapses.

In this chapter a floating gate device is presented which is to be integrated with the charge coupled synapse proposed in [1-4]. The floating gate device is designed using polysilicon and MOS capacitors; the gate of the MOS capacitor and lower plate of the polysilicon capacitor forms the electrically isolated floating gate. Negative charge is stored and removed from the floating gate via Fowler-Nordheim tunnelling. Electrons from the induced inversion region in semiconductor tunnel through the gate oxide and onto the floating gate when a high positive voltage is applied to the control gate. This increases the number of electrons stored on the floating gate causing a reduction of the potential of the floating gate. The application of a large negative voltage to the control gate causes electrons to tunnel back through the gate oxide into the semiconductor. This serves to reduce the number of electrons on the floating gate and increases its potential. The theoretical operation and design equations are derived from MOS physics to produce design guidelines and a theoretical model of the device characteristics. Simulation results of the charge storage characteristics are presented and discussed. Experimental results using CV and pulsed CV measurements taken from the 2nd AMS chip run are also presented. Where required, MOSFET and MOSC parameters extracted in chapter 3 are used for calculations. For convenience these values are presented in Table 4.1 and 4.2.

	Measured	**Typical AMS**
t_{ox}	7.7nm	7.6nm
C_o	4.47 x 10^{-7} Fcm^{-2}	4.54 x 10^{-7} Fcm^{-2}
N_A	2.58 x 10^{17} cm^{-3}	2.12 x 10^{17} cm^{-3}

Table 4.1 – Extracted values and equivalent typical AMS values (where applicable) for p-type MOS-C

	Definition	**Measured**	**Units**
C_{max}	Maximum MOS Capacitance	44.7	pF
C_{min} (HF plot)	Minimum MOS Capacitance	11.1	pF
$\|Q_d\|$	Depletion region charge	2.64 x10^{-7}	Ccm^{-2}
Φ_S	Semiconductor Work Function	5.11	eV
Φ_{gate}	Gate Function difference	4.14	eV
Φ_{MS}	Work Function difference	-0.97	eV
V_T (ideal)	Ideal Threshold Voltage	0.83	V
V_{FB}	Flatband Voltage	2.09	V
C_{FB}	Flatband Capacitance	33.12	pF
L_d	Debye Length	7.9	nm
$V_{mg\ (ideal)}$	Ideal Mid-gap Voltage	-0.167	V
$V_{mg\ (measured)}$	Measured Mid-gap Voltage	-0.141	V
ΔV_{mg}	Mid-gap Voltage Shift	0.0026	V
N_f	Oxide Charge Density	7.26 x10^{10}	cm^{-2}
N_{ss}	Surface States Per Unit Area	1.56 x10^{11}	$cm^{-2}eV^{-1}$

Table 4.2 – Extracted values from C-V plot for MOS-C

The remainder of this chapter is organized as follows; the design and theoretical operation of the floating gate device is presented in section 4.2. A model of the charge storage characteristics is presented in 4.3 together with simulation results and analysis. Experimental results and analysis are presented in section 4.4 with conclusions and discussion in section 4.5

4.2 FGNVM Device

A non-volatile memory device utilizing a floating gate, FG, is designed and implemented using a mixed signal CMOS fabrication process. The FG device is used to store the synaptic weight locally at the associated synapse as negative charge upon the FG. Charge will be added or removed by to and from the FG via a tunnelling mechanism. One of the constraints for the design of the FG device is that it must be compact (see chapter 1) as it is to be integrated with all of the charge transfer synapses used within the neural network [1-3]. In addition to this it will be shown within section 4.3.1 that the coupling coefficient, α, will determine the optimum width and length of the control gate within the device.

The operation and relevant theoretical equations of the FG device is outlined in section 4.2.1. In section 4.2.2 an overview of the design and design equations for the FG device are presented. Specifically the design equations focus upon the determination of C_{poly} and as such width and length of C_{poly} based upon the choice of C_{ox} and α. A minimum sized device is first established, this is then expanded such that the optimum, compact, sized device for a maximum α value is found.

4.2.1 FGNVM Device – Operation

The non-volatile memory device consists of a floating polysilicon gate, as shown in Fig. 4.1. Traditionally in non-volatile memory devices charge is stored and removed from a FG using either Hot Electron Injection, HEI, or Fowler-Nordheim Tunnelling, FN tunnelling. In the scope of this work and due to the design of the FG device FN tunnelling is the preferred method used to modify the charge upon the FG. The FN mechanism is given by equation 4.1 [5].

$$J_{FN} = A{E_{ox}}^{2} exp\left(\frac{-B}{E_{ox}}\right) \tag{4.1}$$

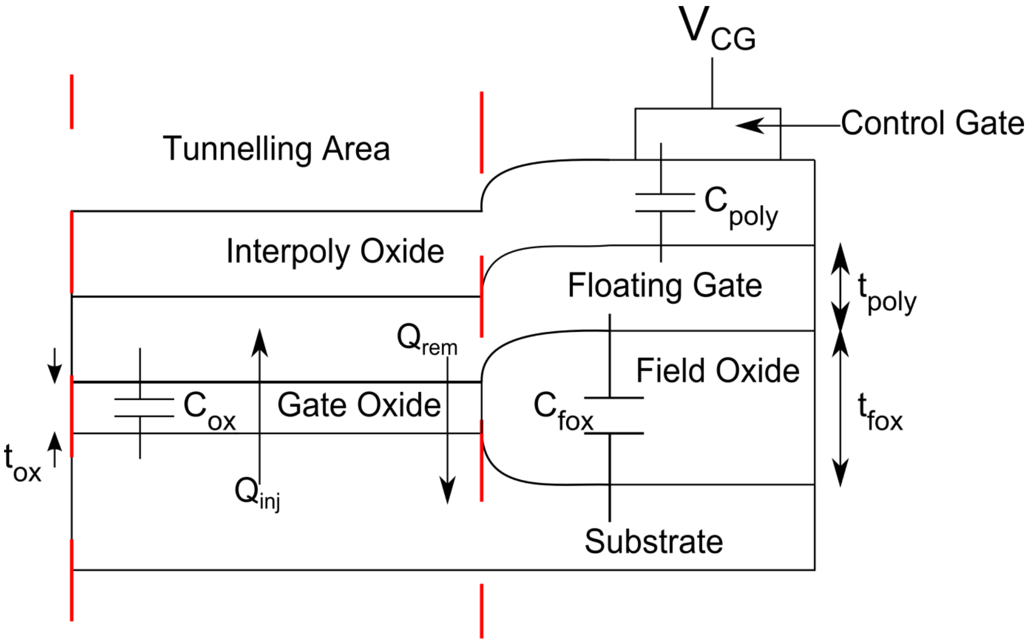


Fig. 4.1 – Cross-section of FG deice, constructed using poly-silicon and MOS capacitors. C_{ox} is the oxide capacitance, C_{poly} is the polysilicon capacitance, and C_p is the field oxide capacitance. t_{ox} is the oxide thickness, t_{poly} is the polysilicon oxide thickness, and t_{fox} is the field oxide thickness. Q_{inj} represents the charge stored and Q_{rem} the charge removed from the FG, both due to FN tunnelling.

To increase the charge stored on the FG a large positive voltage must be applied to the control gate, V_{CG}. This voltage is capacitively coupled on to the FG. The associated field must be sufficiently large, 6-10 MV/cm, for significant FN tunnelling to take place, as shown in Fig. 4.2. In the absence of V_{CG} the charge will remain on the FG, Fig. 4.3.

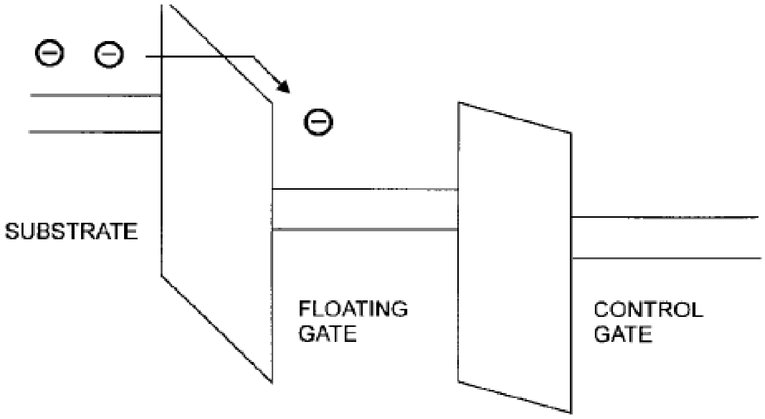


Fig. 4.2 – Band diagram of device showing charge moving in to the potential well of the floating gate [6] (band bending in the semiconductor is omitted)

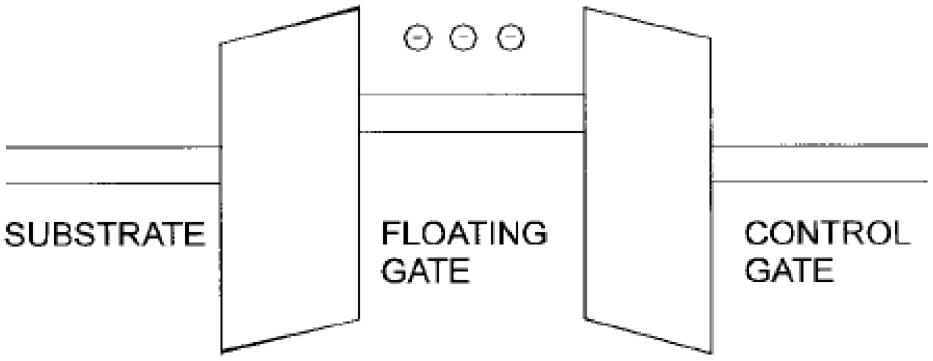


Fig. 4.3 – Band diagram of FGMOS showing trapped charge in the potential well of a floating gate [3] (band bending in the semiconductor is omitted)

Fig. 4.4 shows the arrangement to remove charge stored on the FG, whereby a large negative voltage is applied to V_{CG} and FN tunnelling causes electrons stored on the FG to tunnel back through the gate oxide into the substrate. In either case the duration, pulse width, of V_{CG} determines the magnitude of the charge injected/removed to/from the FG.

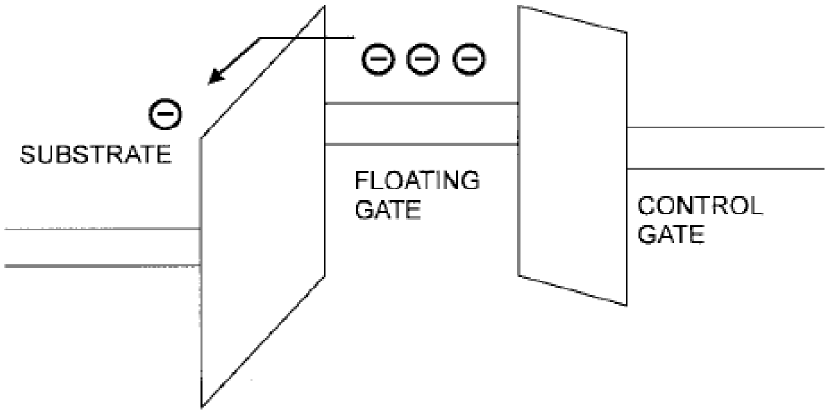


Fig. 4.4 – Band diagram of FGMOS showing charge moving from the potential well of the floating gate

A common non-ideal effect which occurs in FG devices is that initial charge is located on the floating gate. This can occur during fabrication and can be difficult to quantify. This initial charge serves to alter the threshold voltage and charge storage characteristics of the device and must be considered in practice.

4.2.2 FGNVM Device Design

Floating gate devices are traditionally fabricated using a specialized process, however due to cost constraints associated with this work, a standard CMOS process provided by AMS is to be used. A floating gate device can be designed and fabricated within the chosen process using an MOS capacitor, MOS-C, and polysilicon capacitor, as depicted in Figs. 4.5 and 4.6. The FG is created by combining the gate of the MOS-C with the bottom plate of the polysilicon capacitor, (poly1 layer, Fig. 4.5). The FG is electrically isolated from the rest of the device as it is completely enclosed by the gate and interpoly oxides. The control gate is formed by creating a contact to the top plate of the polysilicon capacitor, (poly2 layer) Figs. 4.5 and 4.1. Although the FG is electrically isolated from the rest of the device, it is capacitively coupled to the control gate(s), Figs. 4.1 and 4.6.

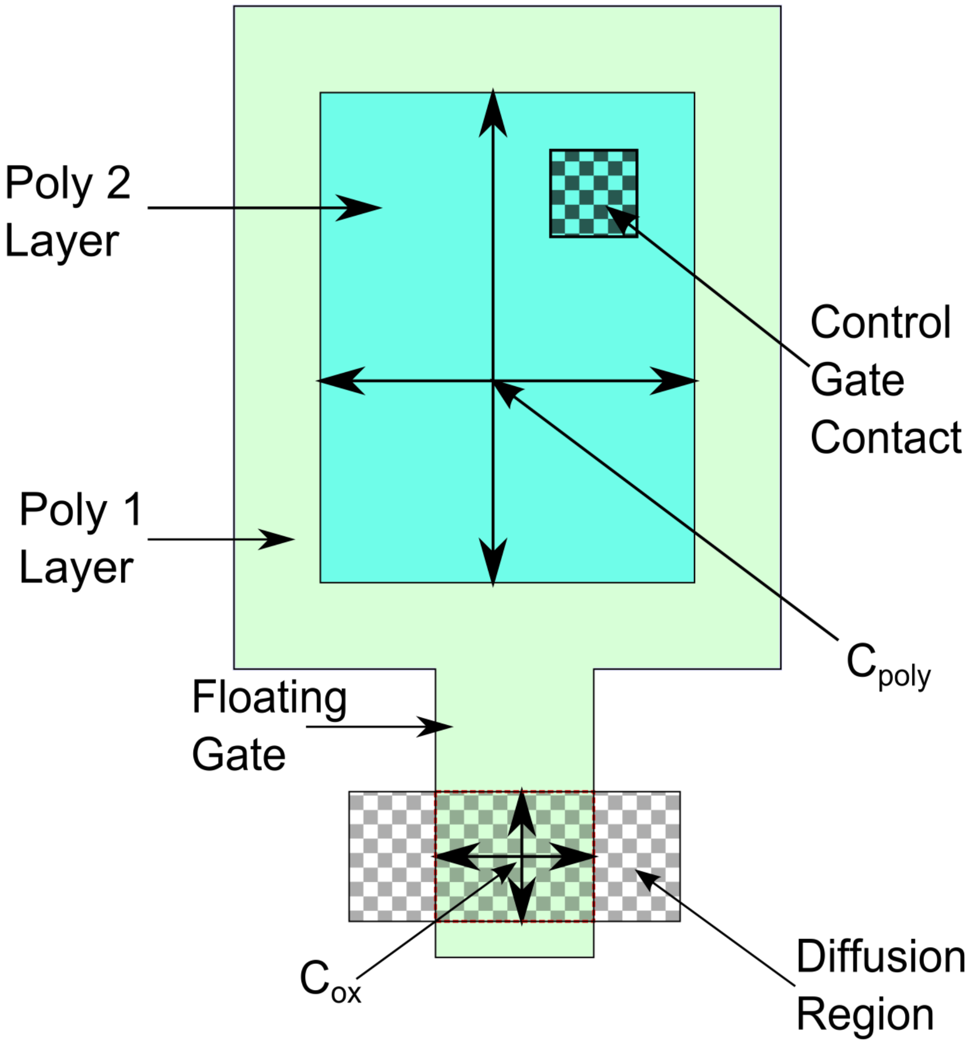


Fig 4.5 – Synaptic weight storage device; fabricated using a polysilicon capacitor and MOS capacitor. Dimensions for each FG device are presented in Table 4.5.

The DC behaviour of the FG device is modelled as follows. [7]. A relationship between the control gate voltages, V_i, and V_{FG} is obtained in order to ascertain the capacitively coupled voltage on the FG, V_{FG}. A coupling coefficient α is introduced to define how much of the control gate voltage(s) is capacitively coupled onto the FG. This subsequently determines the level of FN current. The capacitance of both the FG and control gate(s) determines the magnitude of V_{FG}, as shown in equation 4.2, 4.3. The parameter α must be chosen such that V_{FG} is sufficient to allow the desired level of FN tunnelling to take place and hence control the sensitivity of charge storage, as shown in Fig. 4.6.

$$V_{FG} = \frac{\sum_{i=0}^{i=n} C_i V_i}{C_T} \tag{4.2}$$

$$C_T = C_{ox} + \sum_{i=0}^{i=n} C_i \tag{4.3}$$

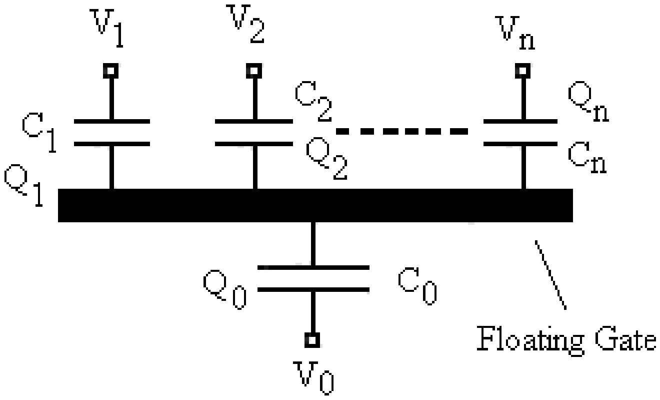


Fig 4.6 – FG capacitive coupling

A FG device with a single control gate is now considered. For a single control gate FG device, V_{FG}, is given by equation 4.4, where C_T is the capacitance seen by the FG, and V_{CG} is the voltage applied to the control gate.

$$V_{FG} = \frac{C_{poly} V_{CG}}{C_T} \tag{4.4}$$

$$C_T = C_{ox} + C_{poly} \tag{4.5}$$

From equation 4.2, the coupling coefficient, α, can be given as;

$$\alpha = \frac{C_{poly}}{C_T} \tag{4.6}$$

Equation 4.6 indicates that α is dependent upon the magnitude of C_{poly} and C_{ox}. C_{poly} is dependent upon the size/area of the control gate; equation 4.7 is used to determine the capacitance of C_{poly} for the control gate.

$$C_{poly} = \left(\frac{\varepsilon_{ox}}{t_{poly}}\right) A_{poly} = C_p A_{poly} \tag{4.7}$$

Where A_{poly} is the area of the control gate, t_{poly} is the interpoly oxide thickness and C_p is the capacitance of the interpoly oxide per μm^2 and has range between 0.78 < 0.86 < 0.96 $fF/\mu m^2$.

The capacitance of the gate oxide, C_{ox}, is dependent upon the area of the floating gate, A_{tun};

$$C_{ox} = \left(\frac{\varepsilon_{ox}}{t_{ox}}\right) A_{tun} = C_o A_{tun} \tag{4.8}$$

Where t_{ox} is the gate oxide thickness and C_o is the capacitance of the gate oxide per unit area (μm^2). C_o has a range of 4.26 < 4.54 < 4.86 $fF/\mu m^2$. For a compact design C_{ox} and C_{poly} must be as small as possible, yet still maintain the design rules provided by AMS [8]. The AMS process document indicates that the smallest possible width and the length for FG is 0.35μm x 0.40μm, Fig. 4.7.

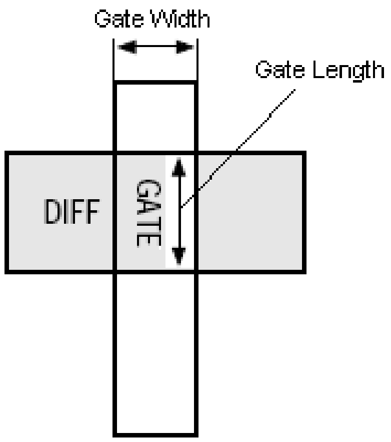


Fig 4.7 – MOS-C polysilicon gate width and length

By contrast several design constraints affect the minimum size of C_{poly}; specifically the minimum size of C_{poly} is dependent upon the size of the contact which must be made to the top polysilicon layer, poly2, as well as the minimum spacing which must be observed, Fig. 4.8.

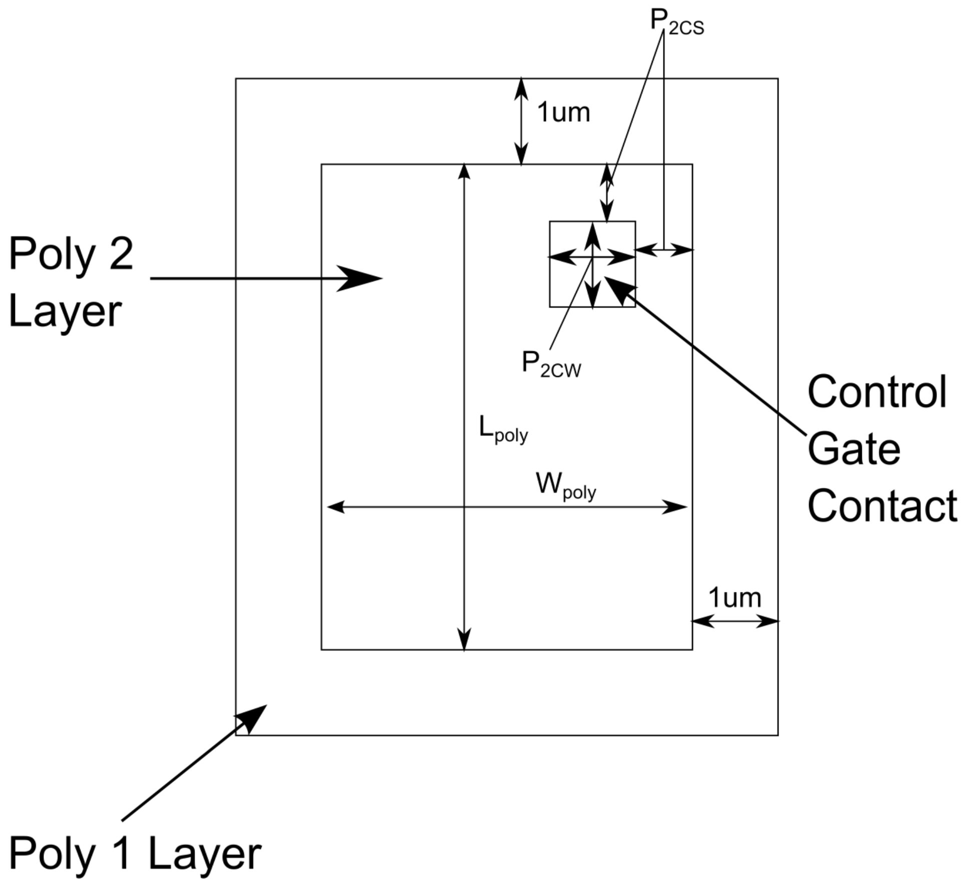


Fig 4.8 – Polysilicon capacitor dimensions

Figure 4.8 indicates that the minimum values of W_{poly} and L_{poly} can be given as equation 4.9;

$$W_{poly} = L_{poly} = P_{2CW} + 2P_{2CS} \qquad (4.9)$$

where P_{2CW} = 0.4μm and is the width of the contact made to poly2, while P_{2CS} = 0.6μm and is the spacing between the edge of poly2 and the contact; W_{poly} = 1.6μm and the minimum area for C_{poly}, Ap_{oly} = 2.56μm. This is the minimum area of C_{poly}. Use the minimum values of the widths and lengths for both C_{ox} and C_{poly}, gives C_{ox} = 0.68 fF and C_{poly} = 2.46 fF with α = 0.78.

Equations 4.6 - 4.8 indicate that α is dependent upon A_{poly} and A_{tun}; any variation in these will change the overall value of α, and consequently V_{FG}. By rearranging equation 4.6 it is possible to determine a value for C_{poly} in terms of C_{ox} and α. C_{ox} and α are values which can be chosen by the designer, equation 4.10.

$$C_{poly} = \frac{\alpha}{1-\alpha} C_{ox} \qquad (4.10)$$

From equation 4.10, $\frac{\alpha}{1-\alpha}$ can be defined as the ratio of C_{poly} to C_{ox}. Table 4.3 shows the ratio of C_{poly} to C_{ox} based upon the chosen value of α. It should be noted that α ≥ 1 or 0 is not possible.

α	$\frac{\alpha}{1-\alpha}$	α	$\frac{\alpha}{1-\alpha}$
0.1	0.11	0.6	1.50
0.2	0.25	0.7	2.30
0.3	0.43	0.8	4.00
0.4	0.67	0.9	9.00
0.5	1.00		

Table 4.3 –C_{poly} to C_{ox} ratio required to ensure desired coupling coefficient

A value of α = 0.9 ensures that the majority of V_{CG} is capacitively coupled onto the FG. Therefore C_{poly} must be 9 times larger than C_{ox}. For the minimum value of C_{ox}, C_{poly} = 6.12fF. Assuming that C_{poly} has unity aspect ratio, W_{poly} and L_{poly} can be found using Equation 4.11. Table 4.4 presents the dimensions C_{ox} and C_{poly} for the minimum sized FG device with an α = 0.9.

$$W_{poly} = L_{poly} = \sqrt{\frac{C_{poly}}{C'_{poly}}} \tag{4.11}$$

W_{Cox} (µm)	L_{Cox} (µm)	A_{tun} (µm^2)	C_{ox} (fF)	C_{poly} (fF)	W_{poly} (µm)	L_{poly} (µm)	A_{poly} (µm^2)	A_{total} (µm^2)
0.35	0.4	0.14	0.68	6.12	2.53	2.53	6.38	24.37

Table 4.4 – Minimum FG device specifications

4.3 Charge Storage Model

4.3.1 Introduction

Section 4.3.3 presents a model of the charge storage characteristics of the FG device shown in Fig. 4.1 and outlined in the previous section. The model uses an iterative procedure to generate characteristics for the device, including; charge vs. time (t), charge vs. floating gate potential, charge vs. oxide electric field (E_{ox}). In section 4.3.4, simulation results using the model equations are undertaken using Matlab. Section 4.3.2 presents a note on areal effects and the convention used with FG devices.

4.3.2 Areal Effects and convention

In memory devices it is conventional to quote charge as cm^{-2}. This is referred to as the number density, N. Charge density, Q, is expressed in C/cm^{-2}.

$$N_{inj} = \frac{Q_{inj}}{q} \tag{4.12}$$

The areal form, Q, is convenient as it works better when combining with C_{ox}, which is expressed in Fcm^{-2}, so as to find potential when the model is used with a 1-D system. The total charge concentrations associated with N and Q can be expressed as follows;

$$Q_{injT} = Q_{inj}A_{tun} \tag{4.13}$$

$$N_{injT} = N_{inj}A_{tun} \tag{4.14}$$

Additionally, the injector area, $A_{tun} = W_{Cox}L_{Cox}$, is smaller than that of the FG, $A_{FG} = W_{FG}L_{FG}$, Fig. 4.10. Following the previous convention the FG charge is related to the injected charge as;

$$Q_{inj}A_{tun} = Q_{FG}A_{FG} \tag{4.15}$$

So

$$Q_{FG} = Q_{inj}\frac{W_{Cox}L_{Cox}}{W_{FG}L_{FG}} \tag{4.16}$$

It then follows that:

$$V_A = \frac{Q_{FG}}{C_0} \tag{4.17}$$

That is,

$$V_A = \frac{Q_{inj}}{C_0}\frac{W_{Cox}L_{Cox}}{W_{FG}L_{FG}} \tag{4.18}$$

Fig. 4.11(b) shows the total area of the FG, A_{FG}, can be given as;

$$A_{FG} = A_{s1} + A_{s2} + A_{s3} + A_{tun} \tag{4.19}$$

Where A_{s1} A_{s2}, and A_{s3} are;

$$A_{s1} = W_{s1}L_{s1} \tag{4.20}$$

$$A_{s2} = W_{s2}L_{s2} \tag{4.21}$$

$$A_{s3} = W_{s3}L_{s3} \tag{4.22}$$

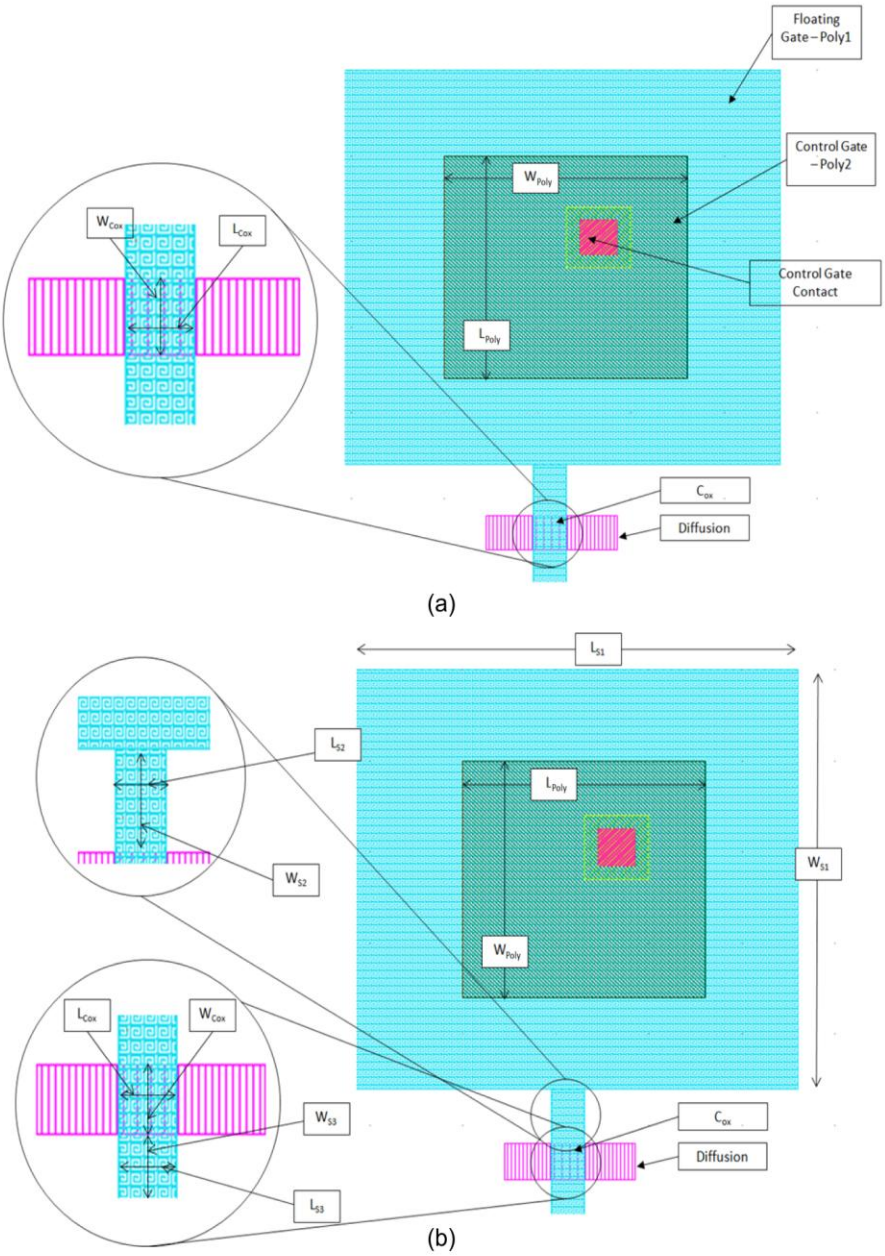


Fig 4.9 – Layout of FG device showing (a) Poly2 layer width and length, control gate contact and tunnelling region, C_{ox}, width and length (b) FG dimensions as given by equations 4.19-4.22

4.3.3 Charge Storage Characteristic Equations

The aim of the modelling is to characterise both the transient charge storage capabilities and the effect of varying the tunnelling area within the FGNVM device shown in Fig. 4.1. It is assumed that the FGNVM device has a capacitive coupling coefficient α, of 0.9.The model is based on a set of equations which are iterated to allow determination of the charge upon the gate with respect to time, voltage, FN current density and oxide electric field. The surface potential at inversion is calculated from experimental results presented in Table 4.2.

The capacitance of the gate oxide is given as

$$C_{ox} = C_0 W_{Cox} L_{Cox} \tag{4.23}$$

The interpoly capacitance C_{poly} is calculated as:

$$C_{poly} = \frac{\alpha}{1-\alpha} C_{ox} \tag{4.24}$$

Where α = 0.9, the maximum possible value for a FGNVM device. Assuming that C_{poly} has unity aspect ratio, W_{poly} and L_{poly} can be found using Equation 4.25;

$$W_{poly} = L_{poly} = \sqrt{\frac{C_{poly}}{C_p}} \tag{4.25}$$

The charge injected onto the FG, Q_{inj} represents the change in the associated weight; Q_{inj} α Δw. The charge is injected by the Fowler-Nordheim mechanism;

$$J_{FN} = A{E_{ox}}^2 exp\left(\frac{-B}{E_{ox}}\right) \tag{4.26}$$

Constants A and B are given by equations 4.27 and 4.28 respectively:

$$A = 1.54x10^{-6}\frac{m_o}{m_{ox}}\frac{1}{\phi_B}\ A/V^2 \tag{4.27}$$

$$B = 6.83x10^{7}\sqrt{\frac{m_{ox}}{m_o}}\phi_B{}^{3/2}\ V/cm \tag{4.28}$$

where m_o is the mass of an electron at rest, m_{ox} is the effective mass of an electron in the insulator and ϕ_B is the barrier height for injection from semiconductor to oxide. It should be noted that the constants A, B are strictly for tunnelling from a metal contact but are similar to the case of injection from a semiconductor [5] and serve the purpose for illustrating the model and method.

Fig. 4.10 presents the cross-section of a FG device constructed using a poly-silicon capacitor. The charge injected onto the FG, Q_{inj}, can be found from consideration of the current in the thin tunnelling oxide, t_{ox} over a time step, Δt. The following equations 4.29, 4.30, 4.38– 4.40 are used to determine Q_{inj} (Δw) and the associated potential of charge stored on the FG, ΔV_w.

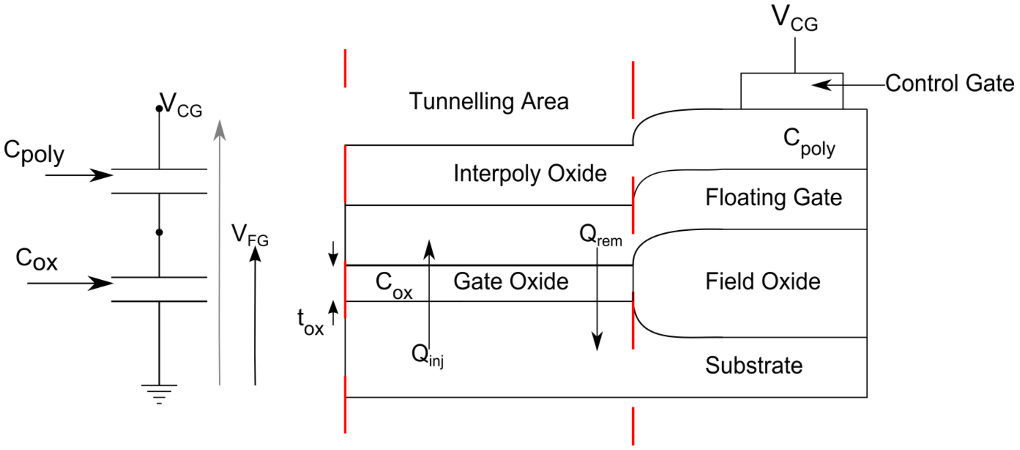


Fig. 4.10 – Equivalent capacitor diagram and cross section of FG device, constructed using poly-silicon and MOS capacitors. C_{poly} is the capacitance of the interpoly oxide, C_{ox} is the capacitance of the tunnelling oxide, V_{CG} and V_{FG} are the voltages applied to the control gate and coupled onto the FG respectively Q_{inj} represents the charge stored on the FG and Q_{rem} represents the charge removed from the FG, both due to FN tunnelling.

The capacitively coupled voltage, V_{FG} which falls across t_{ox} is shown in Fig.4.12, and given by equation 4.29.

$$V_{FG} = \alpha V_{CG} \tag{4.29}$$

Where α is the capacitive coupling coefficient, defined as $\alpha = \frac{C_{poly}}{C_{ox}+C_{poly}}$. The electric field in the oxide, E_{ox} is given by (4.30), assuming that there is no charge in the oxide or initially stored on the FG

$$E_{ox} = \frac{V_{FG}-\phi_S}{t_{ox}} \tag{4.30}$$

where V_{FG} is the potential of the FG and ϕ_S is the surface potential at the oxide-semiconductor interface. The field at successful time steps, Δt, can be found from equation 4.38. Equation 4.38 is derived as follows;

Starting with the time dependent FN Equation:

$$J_{FN} = C_0 \frac{dV_{ox}}{dt} = AE_{ox}{}^2 exp\left(\frac{-B}{E_{ox}}\right) \tag{4.31}$$

Take the time derivative of electric field:

$$\frac{dE_{ox}}{dt} = \frac{1}{dt_{ox}} \frac{dV_{ox}}{dt} \tag{4.32}$$

hence

$$J_{FN}(E_{ox}) = C_0 t_{ox} \frac{dE_{ox}}{dt} \tag{4.33}$$

Separate variables:

$$J_{FN}(E_{ox})dt = C_0 t_{ox} dE_{ox} = A{E_{ox}}^2 exp\left(\frac{-B}{E_{ox}}\right) dt \tag{4.34}$$

$$C_0 t_{ox} \frac{1}{A{E_{ox}}^2 exp\left(\frac{-B}{E_{ox}}\right)} dE_{ox} = dt \tag{4.35}$$

$$\frac{C_0 t_{ox}}{A} \int_{E_{ox}(i)}^{E_{ox(i+1)}} \left[{E_{ox}}^{-2} exp\left(\frac{B}{E_{ox}}\right)\right] dE_{ox} = \int_{t(i)}^{t(i+1)} dt \tag{4.36}$$

Where $t_{(i+1)} - t_{(i)} = \Delta t$, the time step. Integrating, putting in limits and re-arranging gives

$$ln\left[\Delta t \frac{AB}{C_0 t_{ox}} + exp\left(\frac{B}{E_{ox}(i)}\right)\right] = \left(\frac{B}{E_{ox(i+1)}}\right) \tag{4.37}$$

Finally,

$$E_{ox(i+1)} = B\left[ln\left(\Delta t \frac{AB}{t_{ox} C_0} + exp\left(\frac{B}{E_{ox}(i)}\right)\right)\right]^{-1} \tag{4.38}$$

The associated change in potential is calculated by finding the difference between successive steps of field:

$$\Delta V_w = t_{ox}(E_{ox}(i) - E_{ox}(i+1)) \tag{4.39}$$

The charge per unit area injected onto the FG for the duration of the pulse width Δt is then found as

$$\Delta w \propto Q_{inj} = C_0 V_{\Delta w} \tag{4.40}$$

With the areal value given by equation 4.41;

$$N_{inj} = \frac{Q_{inj}}{q} \tag{4.41}$$

The charge per unit area as seen by the FG and the areal density are;

$$Q_{FG} = Q_{inj} \frac{A_{tun}}{A_{fg}} \tag{4.42}$$

$$N_{FG} = \frac{Q_{FG}}{q} \tag{4.43}$$

The total charge, Q_w and total potential of Q_w, V_w, stored on the FG are given as;

$$V_w = V_{FG} - \phi_S - V_{Q_{init}} - \left(E_{ox(i)} * t_{ox}\right) \tag{4.44}$$

$$Q_w = V_w C_0 \tag{4.45}$$

4.3.4 Simulation Results

Simulations of the charge storage characteristics of the FG device theoretically outlined are undertaken using Matlab. The simulations focus on the effect of varying the gate oxide area of the MOS capacitor, known as the tunnelling area, upon the charge storage capabilities of the device. Throughout the simulations V_{CG} = 10V, α = 0.9, Q_{init} = 0, C_{poly}, and W_{poly} = L_{poly} are calculated using equations 4.24 and 4.25 respectively. Table 4.5 presents C_{poly}, W_{poly} and L_{poly} and A_{total}, the total area of the device, for the chosen tunnelling areas used within the simulations. Table 4.6 presents definition of variables used in the model.

W_{Cox} (μm)	L_{Cox} (μm)	A_{tun} (μm^2)	C_{ox} (fF)	C_{poly} (fF)	W_{poly} (μm)	L_{poly} (μm)	A_{poly} (μm^2)	A_{total} (μm^2)
0.35	0.4	0.14	0.68	6.12	2.53	2.53	6.38	24.37
0.5	0.5	0.25	1.22	10.94	3.38	3.38	11.4	33.52
1	1	1	4.86	43.74	6.75	6.75	45.57	84.00
1.5	1.5	2.25	10.94	98.42	10.13	10.13	102.52	157.45
2	2	4	19.44	174.96	13.50	13.50	182.25	253.43
2.5	2.5	6.25	30.38	273.40	16.88	16.88	285	372.50

Table 4.5 – FGNVM device specifications

Variable	Definition	Units
E_{ox}	Oxide Electric field	MVcm^{-1}
Q_{inj}	Charge injected	Ccm^{-2}
N_{inj}	Number density of Q_{inj}	cm^{-2}
ΔV_w	Change in V_W	V
Q_{FG}	FG Charge per unit area	Ccm^{-2}
N_{FG}	Number density of Q_{FG}	cm^{-2}
Q_W	Weight Charge	C
N_{elec}	Number of electrons	-
V_W	Potential of stored Q_w	V
V_B	Overall potential of FG	V

Table 4.6 – Definition of variables used in model

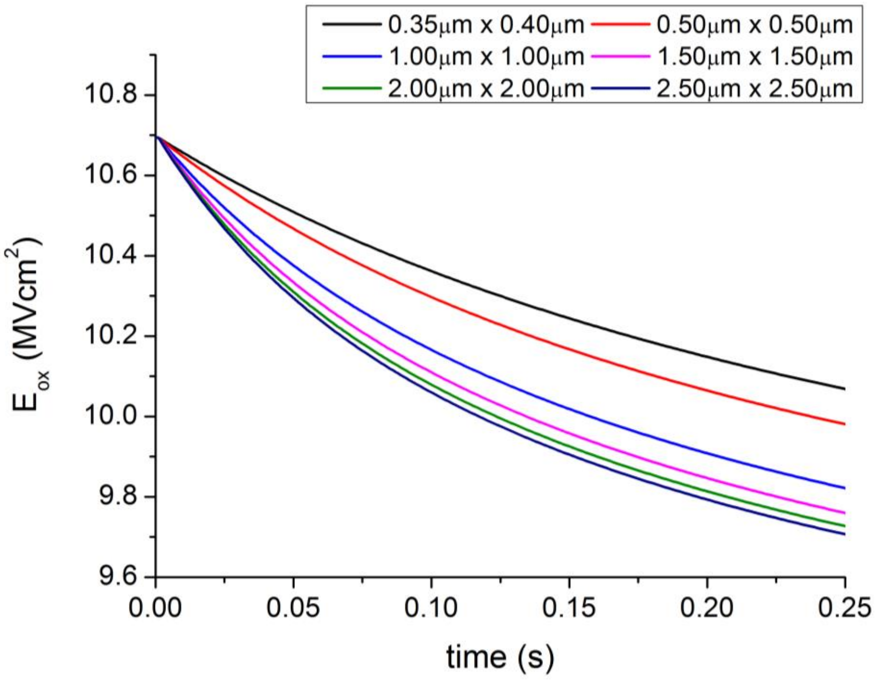


Fig 4.11 – Variation of the gate oxide electric field, E_{ox}, with respect to time and A_{tun}

Figure 4.11 presents the variation of E_{ox} with injection time for the various tunnelling areas presented in Table 4.5. The results indicate that the tunnelling area, A_{tun}, has a dramatic effect upon the charge storage characteristics. A larger A_{tun} allows for a larger flux of electrons to tunnel through C_{ox} causing more charge to be stored on the FG for the same control gate voltage, V_{CG}. Figure 4.11 also indicates that E_{ox} decreases with time, causing a decrease in J_{FN}. The oxide field E_{ox} decreases from its initial value of 10MV/cm to approximately 9.7MV/cm for the minimum tunnelling area used. This decrease in E_{ox} is due to the increase in charge stored on the FG. Since J_{FN} decreases, the flux of electrons onto the FG also decreases. Charge injected and stored on the FG also begins to decrease. Therefore E_{ox} causes a decrease in the charge injected on the FG, Q_{inj}, Fig. 4.12. Q_{inj} decrease from its initial value of approximately 225nCcm^{-2} to approximately 50nCcm^{-2} (again this is for the smallest tunnelling of 0.35µm x 0.40µm). As with E_{ox}, Q_{inj} is dependent upon the tunnelling area; the greater the tunnelling area the more charge is injected for the same initial conditions and time period. It should be noted that the decrease in Q_{inj} is not only due to the decreasing E_{ox}, but also due to the increase in Q_w. The areal density of Q_{inj}, N_{inj}, over the same period of time is presented in Fig. 4.12. N_{inj} decreases with respect to E_{ox} and at the same rate as Q_{inj}.

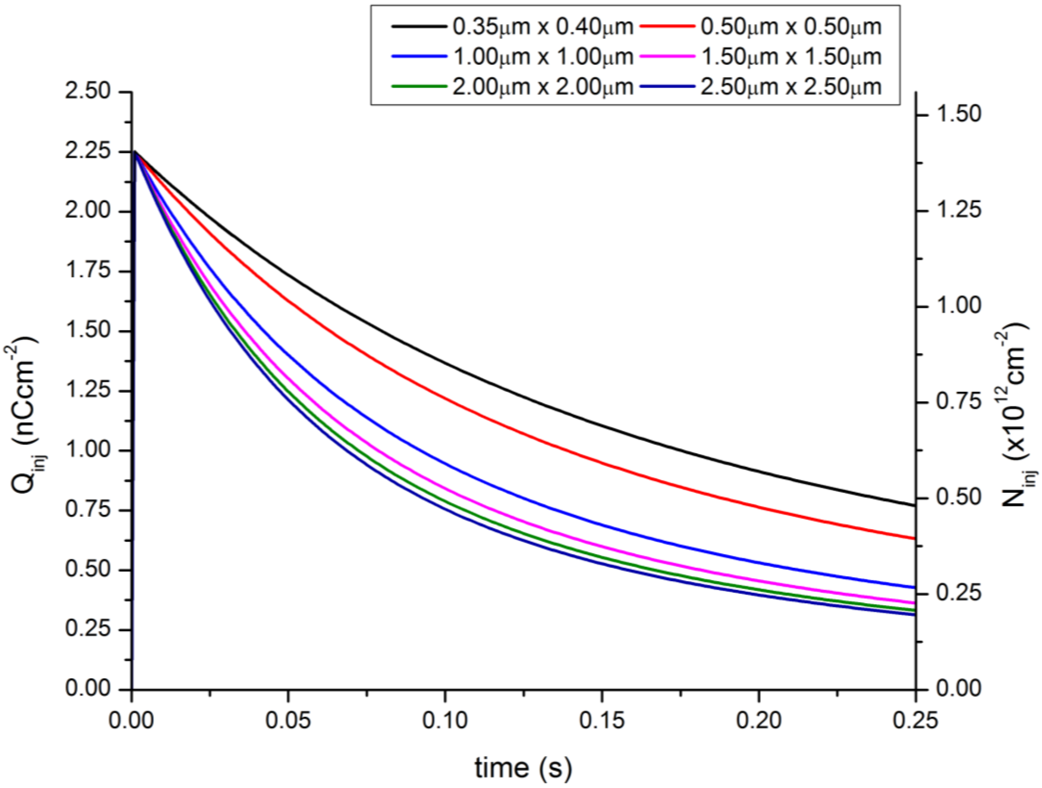


Fig 4.12 – Variation of Q_{inj} and $N_{inj,}$ with respect to time and A_{tun}

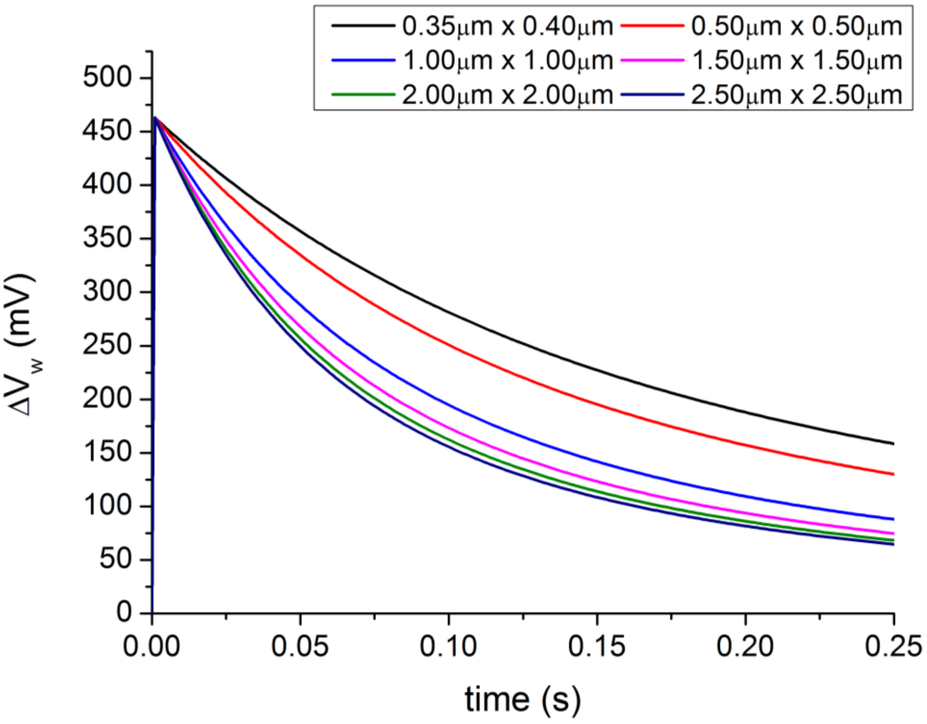


Fig 4.13 – Variation of ΔV_w with respect to time and A_{tun}

The variation in the potential associated with Q_{inj}, ΔV_w, for the various tunnelling areas is presented in Fig. 4.13. ΔV_w decreases from an initial value of 465mV to approximately 60mV (for a tunnelling area of 0.35µm x 0.40µm). Figure 4.14 presents the charge injected and areal density with respect to the area of the FG, Q_{FG} and N_{FG}, given by equations 4.42 and 4.46 respectively. Fig. 4.14 indicates that Q_{FG} varies with A_{tun}. The decrease in Q_{FG} with respect to time is due to two main factors. Firstly increasing Q_w results in decreasing E_{ox}, which in turn causes a decrease in FN current.

$$N_{FG} = \frac{Q_{FG}}{q} \tag{4.46}$$

Figure 4.15 presents the total charge stored on the FG, Q_w, and the total number of electrons physical stored on the FG. As with previous plots, Q_w is also dependent upon the tunnelling area with the FG device. Specifically by increasing the tunnelling area within the device a great flow of electrons can tunnel from the semiconductor and onto the FG. This causes a larger amount of charge to be stored over the same time period.

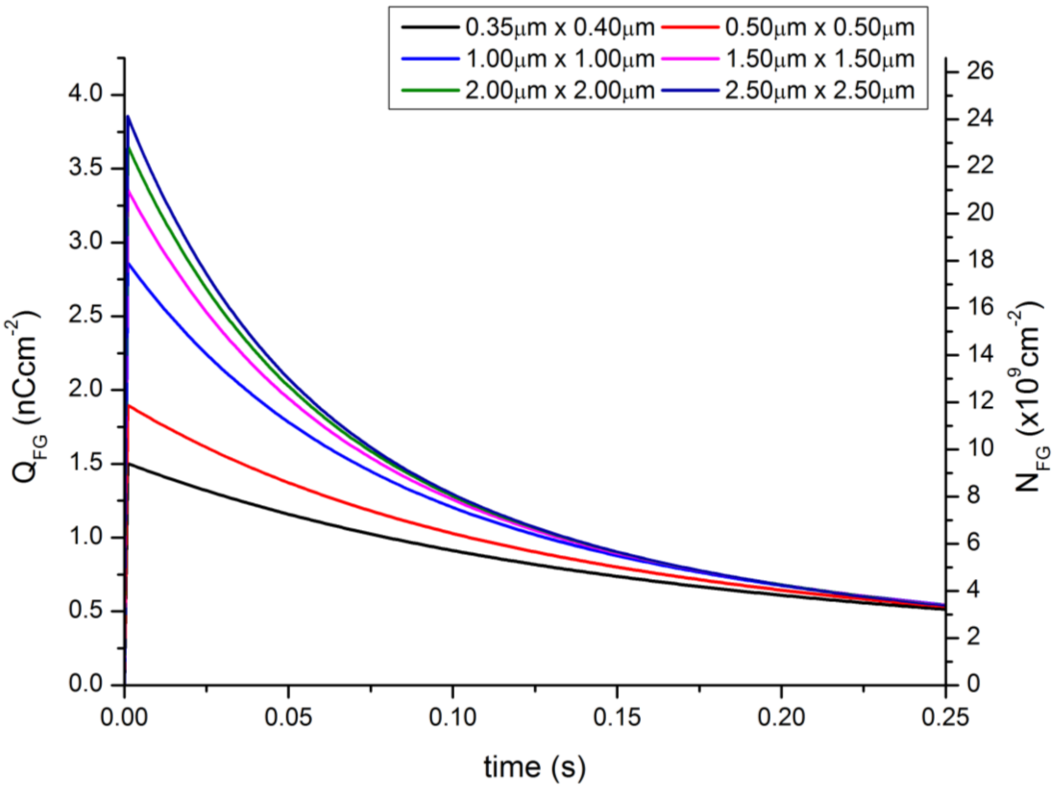


Fig 4.14 – Variation of Q_{FG} and N_{FG} with respect to time and A_{tun}

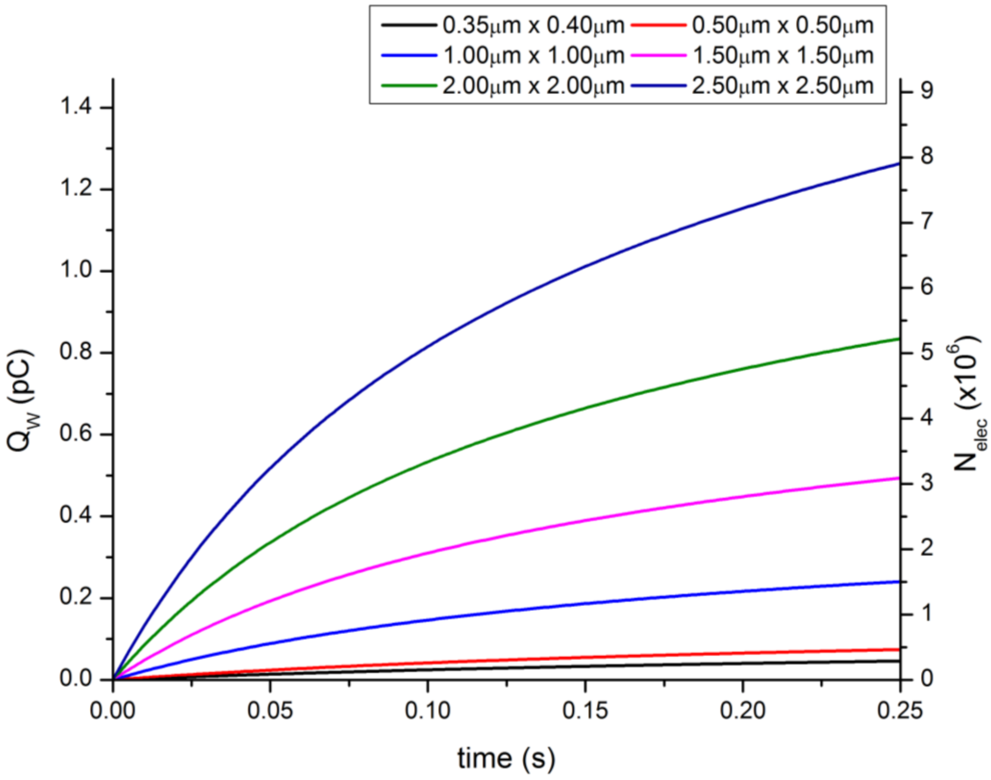
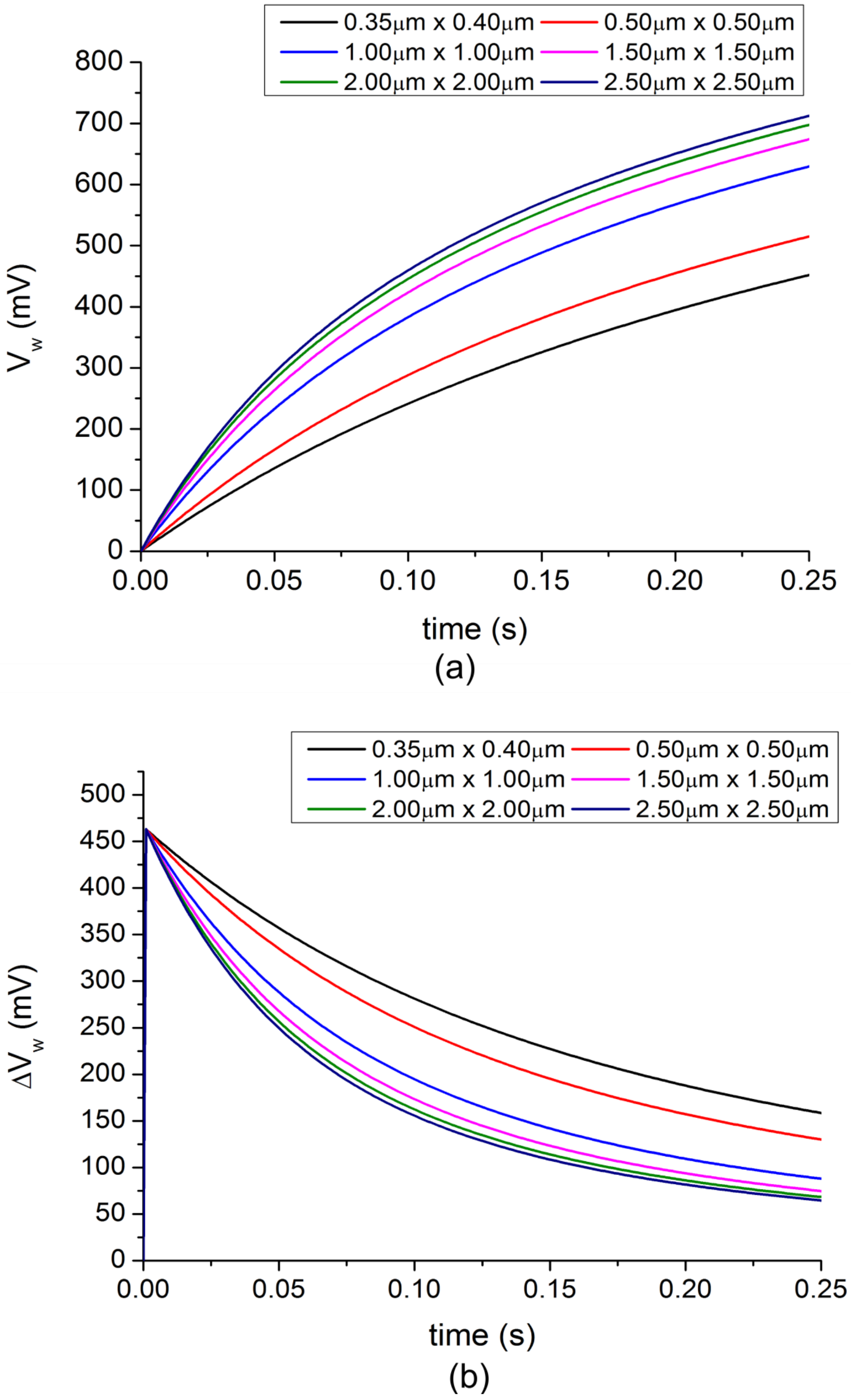


Fig 4.15 – Variation of Q_w and N_{elec} with respect to time and A_{tun}



(a)

(b)

Fig 4.16 – Variation of (a) V_w and (b) ΔV_w with respect to time and A_{tun}

Figure 4.16(a) presents the potential associated with Q_w, V_w while (b) presents the potential $V_{\Delta w}$ associated with Q_{inj}. From Fig. 4.17 and Fig. 4.16(a) it is clear that eventually Q_w (and V_w) will saturate. However as Q_{inj} is dependent upon the tunnelling area, Q_{wmax} (and V_{wmax}) will vary; the larger the tunnelling area, the more charge can be injected and stored on the floating gate to represent the synaptic weight of the associated synapse. The weight charge Q_w increases from an initial value of 0C to approximately 1.6pCcm^{-2} (≈720mV) for a tunnelling area of 2.00µm x 2.00µm.

Finally Fig. 4.17 presents the overall potential of the floating gate, V_b, given by equation 4.47. V_b decreases at the same rate V_w increases. This is due to the increasing negative charge stored on the FG. The decreasing V_b with respect to time causes a decrease in E_{ox} and consequently J_{FN}, Q_{inj} and as a result causes the saturation in Q_w.

$$V_b = V_{FG} - \phi_s - V_{Q_{init}} - \left(E_{ox(i)} * t_{ox}\right) - V_w \tag{4.47}$$

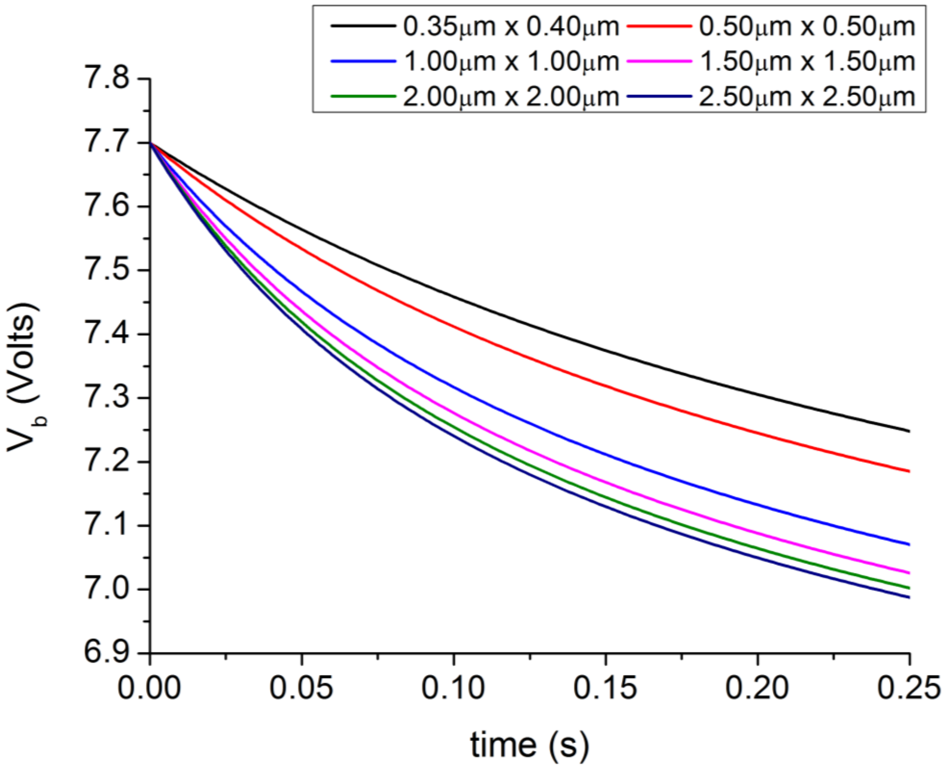


Fig 4.17 – Variation of FG potential, V_B, with respect to time and A_{tun}

4.4 Experimental

4.4.1 CV Plots

In this section High Frequency CV, HFCV, experimental results are presented which confirm that the operation of the FG device outlined earlier in this chapter is consistent with theory. The experimental results are taken from devices fabricated during the 2nd chip run using an AMS 0.35µm CMOS process. Unless otherwise stated, V_{CG} is set to 5V, $\alpha = 0.3$, W_{Cox}, L_{Cox}, and $W_{poly} = L_{poly}$ are calculated using equations 4.24 and 4.25 respectively and their values along with measured values of C_{ox} and C_{poly} are presented in Table 4.7.

W_{Cox} (µm)	L_{Cox} (µm)	A_{tun} (µm^2)	C_{ox} (pF)	C_{poly} (pF)	W_{poly} (µm)	L_{poly} (µm)	A_{poly} (µm^2)
50	100	5000	24.3	9.38	100	100	10,000

Table 4.7 – Minimum FG device specifications

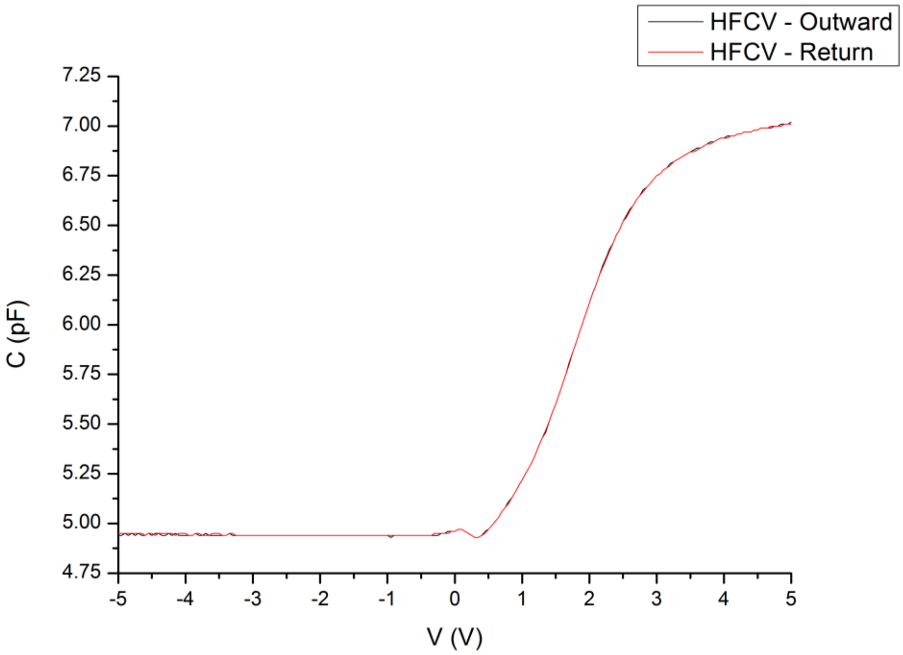


Fig 4.18 – FG device HFCV plot

The HFCV plot for the FG device is presented in Fig 4.18. The gate voltage, V_{CG} is swept from -5V to 5V at a rate of 0.05V/sec with a superimposed ac signal of frequency 1MHz.

The maximum and minimum values for the gate oxide capacitances are 7.15pF and 4.80pF respectively. A return sweep from 5V to -5V was undertaken and no hysteresis was observed, indicating negligible mobile charge in the oxide.

In order to determine whether charge has been stored on the FG, additional HFCV plots are generated where V_{CG} is swept from ±5V to ±15V in 1V steps, with two additional sweeps at ±20V and ±25V also undertaken, Results are presented in Fig. 4.19 which indicate that there is a shift in the CV curves for V_{CG} ±15V, ±20V and ±25V, compared to the initial, ±5V CV. In all cases, the CV curves are shifted to the right of the initial ±5V curve, indicating that there has been storage of negative charge on the FG.

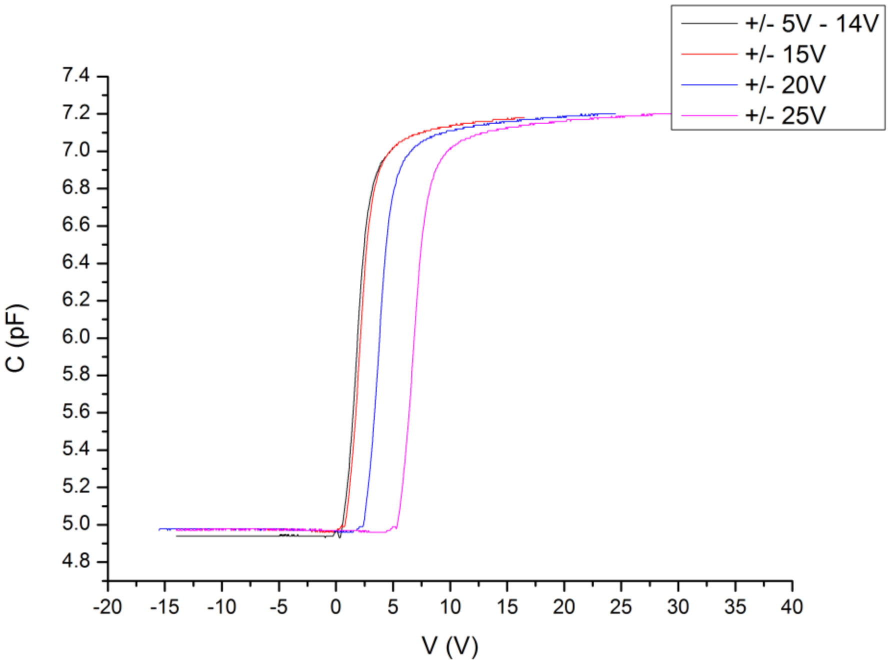


Fig 4.19 – FG device HFCV plots. Voltage swept from ±5V – ±14V, ±15V, ±20V, and ±25V

Table 4.8 presents the capacitively coupled floating gate voltage, V_{FG}, the maximum oxide electric field, E_{ox}, flat band voltage shift between successive CV curves and the initial ±5V CV curve, ΔV_{FB} and charge stored on the FG, Q_W for the given V_{CG} sweep. The weight charge, Q_w is assumed to be uniformly distributed over the poly-Si region which is of thickness t_{poly}, Fig. 4.20. Using Gauss and Poisson's equations, it is possible to show that the charge centroid of Q_W is given by equation 4.48. Note the FG, poly layer, is treated as an equivalent oxide thickness by using the permittivity ratio for t_{poly}.

$$Q_W = Q_{s/c}\left[\frac{t_1 + \frac{\varepsilon_{ox}}{\varepsilon_s}\frac{t_{poly}}{2}}{t_1 + t_2 + \frac{\varepsilon_{ox}}{\varepsilon_s}\frac{t_{poly}}{2}}\right]^{-1}$$

(4.48)

$$Q_{s/c} = \Delta V_{FB} C_{ox} \qquad (4.49)$$

where C_{ox} is the series combination of layers t_1, t_2; that is

$$C_{ox} = \frac{\frac{A_1\varepsilon}{t_1}\cdot\frac{A_2\varepsilon}{t_2}}{\frac{A_1\varepsilon}{t_1}+\frac{A_2\varepsilon}{t_2}}$$

(4.50)

$$V_w = Q_w/C_{ox} \qquad (4.51)$$

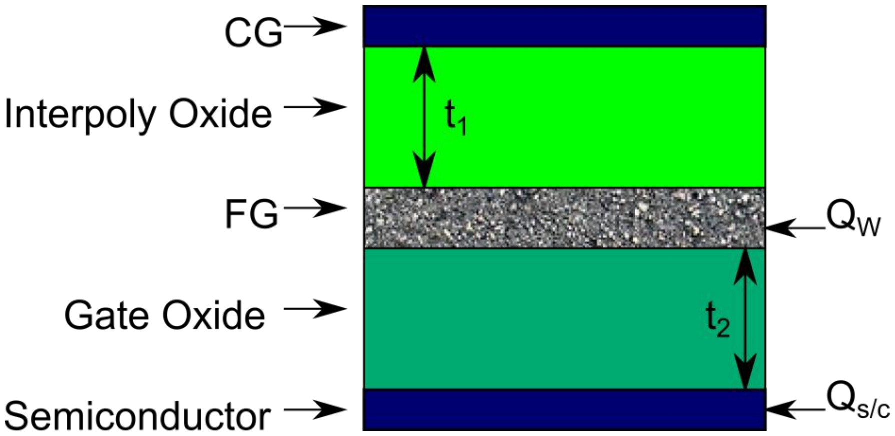


Fig 4.20 – Calculation of weight charge

The image charge in the semiconductor is $Q_{s/c}$. For V_{CG} <±15V the maximum gate oxide field, E_{ox}, is not sufficient to cause significant FN tunnelling within the device. For V_{CG}

±15V, ±20V and ±25V, E_{ox} is sufficiently high such that FN tunnelling causes charge to be stored on the FG. The results presented in table 4.8 also indicate that as V_{CG} is increased, the maximum value of E_{ox} is also increased. This causes the increase in both ΔV_{FB} and Q_W; indicating that by choosing a large value of V_{CG}, more charge can be stored on the FG.

V_{CG} (±Volts)	V_{FG} (Volts)	E_{OX} (MV/cm)	ΔV_{FB} (Volts)	$Q_{S/C}$ (pC)	Q_w (pC)	Q_w (pC/cm^2)	V_W (mV)
5	1.5	2.08	0	0	0	0	0
6	1.8	2.50	0	0	0	0	0
8	2.4	3.33	0	0	0	0	0
10	3	4.17	0	0	0	0	0
12	3.6	5.00	0	0	0	0	0
14	4.2	5.83	0	0	0	0	0
15	4.5	6.25	0.25	1.80	0.14	60.00	19
20	6	8.34	1.95	14.04	1.12	486.97	160
25	7.5	10.42	5	36.00	2.87	1247.00	400

Table 4.8 – FG device characteristics – V_{CG}, V_{FG}, E_{ox}, ΔV_{FB}, and Q_w. Note – AMS substrate breakdown voltage is typically 30V.

4.4.2. Pulsed CV Results and Analysis

In this section, pulsed CV (PCV), experimental results are presented to allow characterization of the FG device. The results are compared with the theory presented previously in this chapter. The experimental results are taken from devices fabricated during the 2nd chip run using an AMS 0.35µm CMOS process. The pulsed CV technique allows for rapid measurement and characterization of the charge storage of the FG device compared to the standard CV methods. The Pulsed CV technique is used to measure the displacement current, i_{disp}, within the device under test (DUT), during the linear ramp-up or ramp-down of an applied voltage pulse to the gate[9-11]. The measured displacement current is proportional to the differential capacitance as shown below:

$$i_{disp} = C\frac{dV}{dt} \tag{4.51}$$

The experimental setup is shown in Fig. 4.21, a voltage pulse with a V_{CG} = ±25V is applied to the DUT, with i_{disp} is amplified and converted to a voltage using a transimpedance amplifier, TA (the full schematic circuit diagram is presented in the Appendix). The output from the TA and along with V_{CG} are sampled by an oscilloscope; allowing for extraction of the CV curves. The change in V_{FB}, ΔV_{FB}, allows extraction of the charge stored on to the FG for a given PW as depicted in Fig. 4.22. Referring to Fig. 4.22, varying the PW of successive voltage pulses, establishes the duration of charge injection provided that the pulse amplitude, PA, is high enough to cause charge injection. The measured shifts in V_{FB} and equations 4.48-4.50 are used to calculate the amount of charge stored on the FG, Q_W.

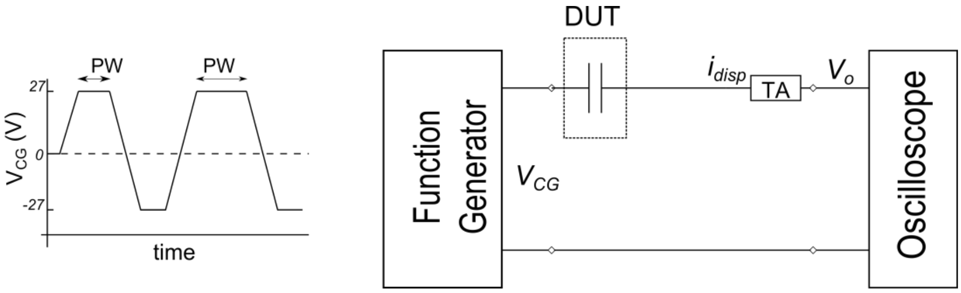


Fig 4.21 - Experimental set up and voltage pulse, V_{CG}, applied to the DUT. i_{disp} is converted to a voltage V_o using a transimpedance amplifier, TA.

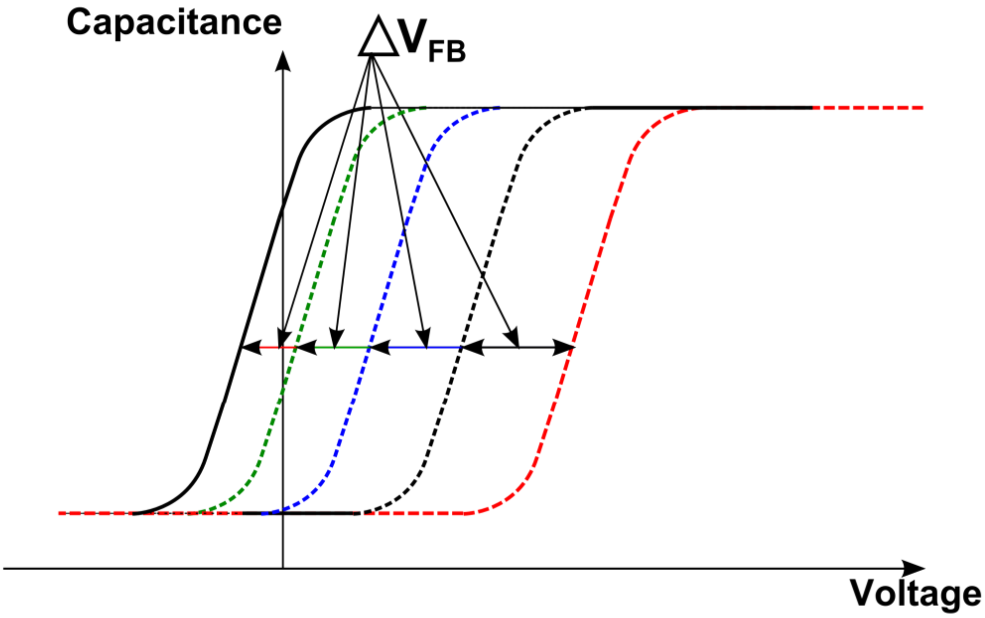


Fig 4.22 – Sequence of CV plots showing shifts due to charge storage.

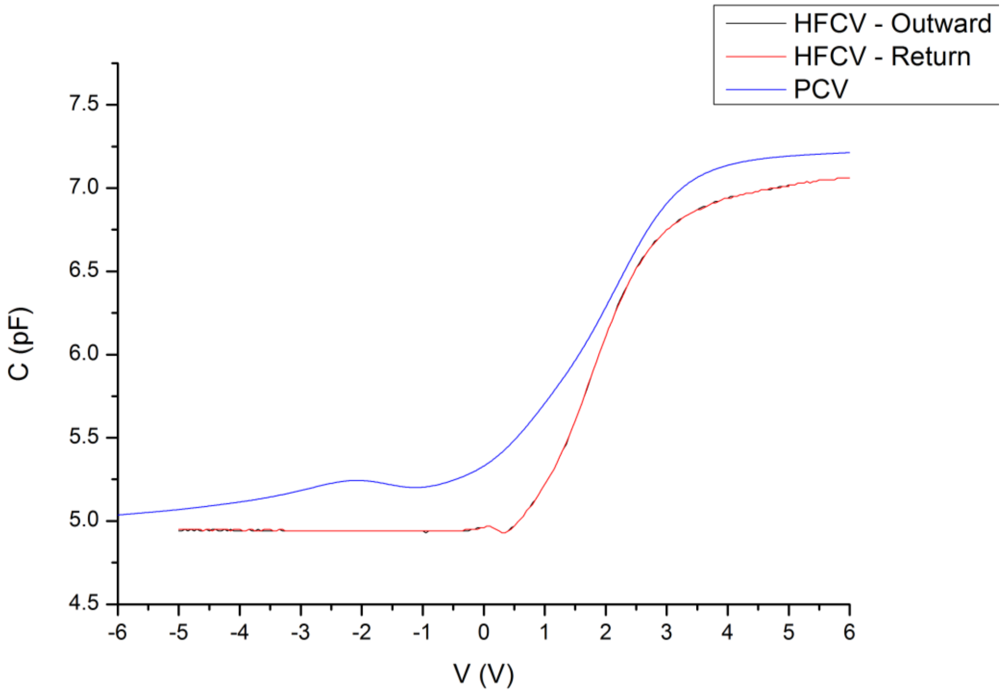


Fig 4.23 – PCV and HFCV plots

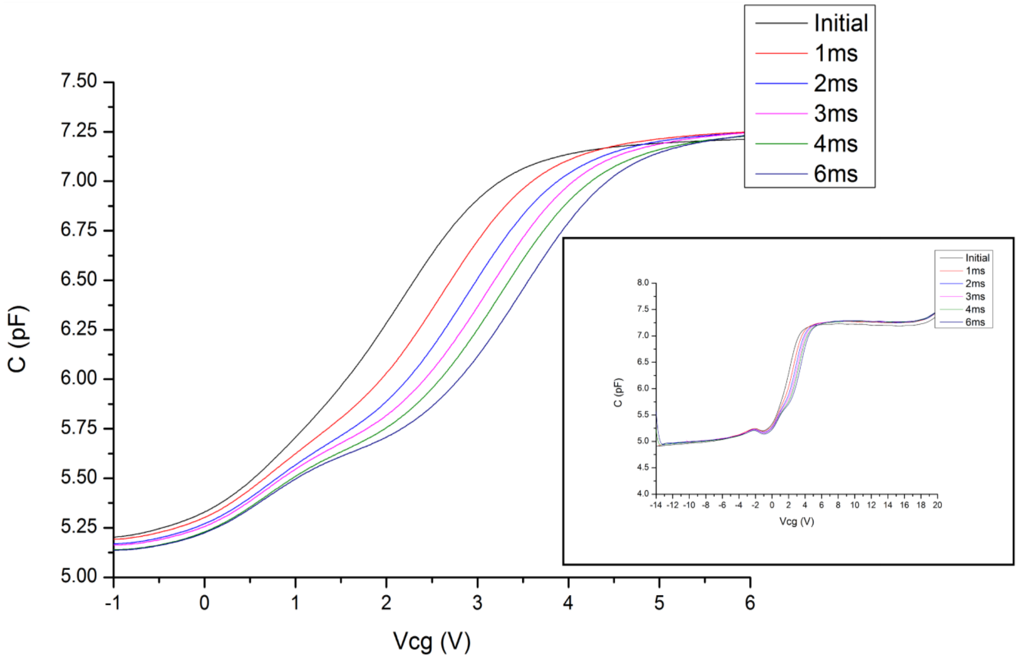


Fig 4.24 – Pulsed CV plots showing hysteresis due to charge storage. Insert showing full Pulsed CV curves

Figure 4.23 presents the CV curve obtained from pulsed CV experimental results along with the HFCV curve obtained in the previous section. The PCV results in Fig. 4.23 are obtained with V_{CG} = ±6V, the capacitance ratio $\alpha = 0.3$, and the device dimensions are given in Table 4.7. The PCV curve in Fig. 4.23 is slightly shifted to the right in comparison to the HFCV curve, in that the PCV curve has a maximum and minimum capacitance of 7.2pF and 5.05pF respectively, compared to the HFCV values of 7.15pF and 4.80pF.

Fig. 4.24 presents the experimental results of successive PCV measurements where the pulse width, PW, between CV measurements is increased from 0 to 6ms in 1ms increments; V_{CG} = ±27V, the capacitance ratio $\alpha = 0.3$, and the device dimensions are given in Table 4.7. The curves are shifted to the right of the initial ±6V consistent with negative charging. Table 4.9 presents ΔV_{FB}, charge stored, Q_W, and V_W for each given pulse width.

Pulse Width (ms)	ΔV_{FB} (Volts)	$Q_{S/C}$ (pC)	Q_w (pC)	Q_w (pC/cm^2)	V_W (mV)
1	0.38	2.74	0.22	94	30.00
2	0.72	5.18	0.41	179.59	57.37
3	0.79	5.69	0.45	197.06	62.95
4	0.85	6.12	0.49	212.02	67.73
6	1.1	9.58	0.76	331.75	105.98

Table 4.9 – Flatband voltage shift, ΔV_{FB}, charge stored, Q_W, and FG weight potential, V_W, due to varying pulse width

Fig. 4.25 presents the charge storage characteristics, (a) V_W and (b) Q_W for the FG device from experimental results and the model presented earlier in this chapter. The insert in Fig. 4.25 (a) shows $\Delta V_{FB}(t)$, $V_W(t)$ and a best fit curves for the results. Model results are generated using device dimensions as the fabricated device, Table 4.7, however t_{ox} is chosen to be the minimum value possible within the AMS process of 7.1nm [8]. The results presented in Fig 4.25 indicate that there is good agreement between the experimental and simulation results generated using the model, for both V_w and Q_W. As the PW is increased, the experimental results show that V_W varies from 30mV to 105.98mV, Q_W from 94pCcm^{-2} to 331.75 pCcm^{-2}. Simulation results indicate that V_W increases from 87mV to 108mV and Q_W from 87 pCcm^{-2} to 339 pCcm^{-2}.

For PW < 4.5ms the model accurately predicts, within 5%, both V_W and Q_W . For a PW = 1ms, V_W = 30mV (experimental) and 28mV (model) with Q_W = 94pC/cm^{-2} (experimental) and 87pC/cm^{-2} (model), there is approximately a 3% difference between experimental and simulation results. Similarly for a PW = 3ms, V_W = 62.95mV and 67mV with Q_W = 212.02pC/cm^{-2} and 205pC/cm^{-2} for the experimental and simulation results respectively; there is a 2% difference between results. When the PW becomes greater than 4.5ms; Fig. 4.29 shows that V_W and Q_W obtained from experimental results begin to diverge from the model. For PW = 5ms, V_W = 91mV (experimental) and 96mV (model) with Q_W = 287.93pC/cm^{-2} (experimental) and 299.09pC/cm^{-2} (model) the difference increases slightly to approximately 4%. The difference between further increase to approximately 6% with PW = 6ms, V_W = 105.98mV(experimental) and 108mV(model) with Q_W = 331.75pC/cm^{-2} (experimental) and 339.42pC/cm^{-2} (model). The experimental results also indicate that Q_W will saturate at a lower value than the model suggests. Overall, on average, there is approximately a 4% difference between the simulation results and experimental results for the same PW value.

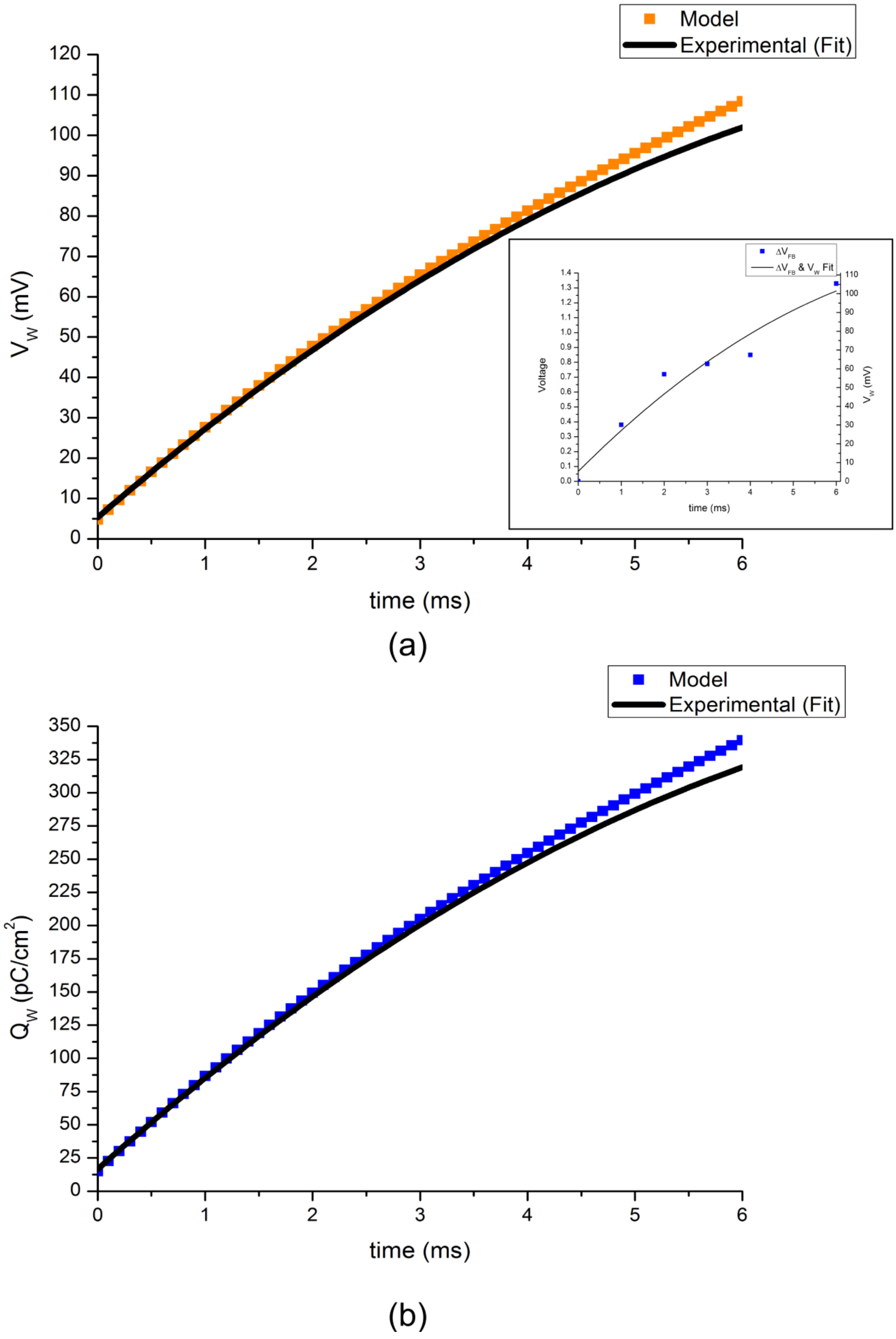


Fig 4.25 – Variation of (a)V_w and (b)Q_w with respect to time. Insert in (a) showing ΔV_{FB} against PW and fit curve

4.5 Conclusions

A floating gate device has been presented suitable for integration with the charge-coupled synapse proposed in [1-4]. The device is designed using a polysilicon and MOS capacitor; the gate of the MOS capacitor and lower plate of the polysilicon capacitor forms the electrically isolated floating gate. The theoretical operation and design equations presented are derived from MOS physics to produce design guidelines. A theoretical model has been developed and compared with experimental results obtained from fabricated devices. The obtained experimental results from both HFCV and PCV measurements indicate that negative charge stored and removed from the floating gate occurs via Fowler-Nordheim tunnelling. The application of a large negative voltage to the control gate causes electrons to tunnel back through the gate oxide into the semiconductor. This serves to reduce the number of electrons on the floating gate and increases its potential. The experimental HFCV results presented confirm that FN tunnelling does occur and causes charge to be stored on the FG. This is shown by a shifting of the HFCV plot to the right of the ideal HFCV. The transient charge storage characteristics have been modelled using physical equations. Experimental results obtained using the pulsed CV technique has been used to validate the model which can be used to predict the charge storage characteristics of the FG device.

4.6 References

[1] Y. Chen, L. McDaid, S. Hall, and P. Kelly, “A programmable facilitating synapse device.”, 2008 *International Joint Conference on Neural Networks*, IJCNN 2008, 2008, Institute of Electrical and Electronics Engineers Inc pp1615-1620, 2008.

[2] T. Dowrick, S. Hall, L. McDaid, O. Buiu. and P. Kelly, “A biologically plausible neuron circuit,” *2007 International Joint Conference on Neural Networks*, IJCNN 2007, Aug 12-17 2007, *Institute of Electrical and Electronics Engineers Inc* pp715-719, 2007

[3] T. Dowrick, S. Hall, L. McDaid, “A silicon based dynamic synapse with depressing response”, *IEEE Transactions on Neural Networks,* in press 2012.

[4] T. Dowrick, *Biologically Motivated Circuits For Third Generation Neural Networks,* PhD Thesis, University of Liverpool, 2010

[5] Z. A. Weinberg, “On tunnelling in metal-oxide-silicon structures”, *Journal of Applied Physics*, vol. 53, no. 7, pp. 5052-5056, 1962.

[6] P. Pavan, R. Bez, P. Olivo and E. Zanoni, “Flash memory cells-an overview”,

Proceedings of the IEEE, vol. 85, no. 8, pp. 1248-1271, 1997

[7] R. Rodriguez-Villegas, Low power and low voltage circuit design with the FGMOS transistor. London: Institution of Engineering and Technology, 2006.

[8] AMS, "0.35µm CMOS C35 Process Parameters," July 2007.

[9] G. Puzzilli, B. Govoreanu, F. Irrera, M. Rosmeulen, and J. Van Houdt, “Characterization of charge trapping in SiO2/Al2O3 dielectric stacks by pulsed C-V technique”, *Microelectronics Reliability*, vol. 47, pp. 508-512, 2007

[10] M. Gurfinkel, H.D. Xiong, K.P. Cheung, J.S. Suehle, J.B. Bernstein, Y. Shapira, A.J. Lelis, D. Habersat, and N. Goldsman , “Characterization of transient gate oxide trapping in SiC MOSFETs using fast I-V techniques” , *IEEE Transactions on Electron Devices*, vol. 55, no. 8, pp. 2004-2012, 2008.

[11] M. Rosmeulen, E. Sleeckx, and K. De Meyer. “Electrical characterization of silicon-rich-oxide based memory cells using pulsed current-voltage techniques”, *Proceeding of the 32nd European Solid-State Device Research Conference, ESSDERC 2002,* pp. 471-474, 2002.

Chapter 5 – Synaptic Weight Update In Analogue Hardware Neural Networks

5.1 Introduction

Chapter 2 presented the notion that Spike-Timing-Dependent-Plasticity, STDP, is believed to be the dominant form of learning that occurs within biological neural networks. STDP determines the effect that one neuron has upon another based upon pre- and post- synaptic input stimuli, Fig. 5.1. Pre-post spiking, where a postsynaptic spike occurs after a presynaptic spike, serves to increase the synaptic weight, while Post-pre spiking serves to reduce the synaptic weight [1-7].

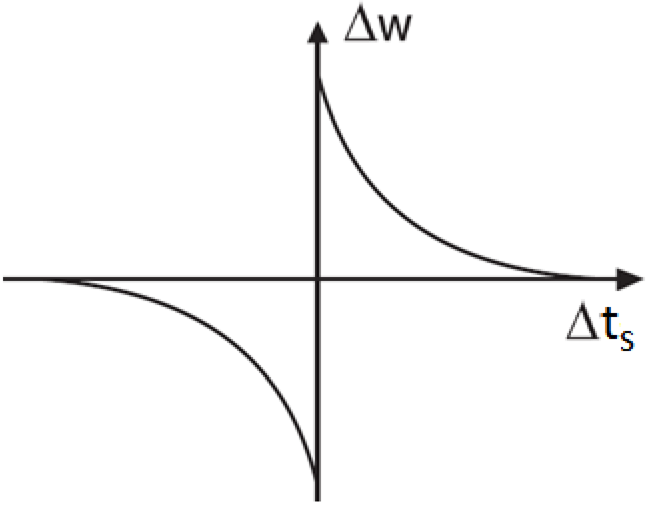


Fig 5.1 – Asymmetric STDP Curve. Δw is the incremental weight and Δt_s is the temporal difference between pre-post firing times; t_{post} is the firing time of the postsynaptic neuron while t_{pre} is the firing time of the presynaptic neuron. $\Delta t_s = t_{post}-t_{pre}$.

In [6-13] it has been established that the modification of the synaptic weight occurs during a specific timing window. Specifically the synaptic weight is only increased if the postsynaptic spike occurs within 20 - 25 milliseconds of the presynaptic spike, while a decrease only occurs when if the presynaptic spike occurs within 20 - 25 milliseconds of the postsynaptic spike. Outside of these windows no modification of the synaptic weight occurs.

From the review of STDP circuits in hardware, Chapter 2, it was concluded that while it has been shown to be possible to implement STDP within HNN, little consideration has been given as to how achievable the functionality of more complex circuits; that is, the scalability of the circuit blocks. In order for the synaptic weight to be updated the proposed synapse(s) need additional circuitry, which can be used to increase or decrease the stored synaptic weight between associated neurons. Therefore, each synapse will have to have its own weight update circuit and the large footprint of the overall solution severely limits their usefulness. However, there is a trade-off between the accuracy of the STDP implemented and that of the additional circuitry.

When scaling is considered, it is clear to see that the additional circuitry required to implement STDP will take up the majority of the silicon. Therefore STDP or an alternative method of implementing synaptic plasticity must allow for the update in synaptic weight while maintaining a compact and low power circuit design, such that more complex neural networks can be constructed. In this chapter a compact circuit is presented which implements STDP using two symmetrically designed circuit blocks, one for pre-post and the other for post-pre spiking. In addition to STDP, the compact circuit also implements a variable critical timing window which determines when and if synaptic modification takes place. The circuits reviewed in chapter 2 do not include this feature.

The remainder of this chapter is organized as follows; the compact STDP circuit is outlined in section 5.2. A theoretical model for the operation of the circuit is outlined with simulation and experimental results confirming the operation also presented. Experimental results are taken from devices fabricated during the 3nd chip run using an AMS 0.35µm CMOS process. Section 5.3 presents analysis of the effect of process variation on the proposed circuit, with particular emphasis on its effect on the critical timing window. Conclusions and discussion are presented in Section 5.4

5.2 Compact STDP Circuit

A compact STDP circuit is proposed which will work in conjunction with the NVM device presented in chapter 4. Specifically the output from the STDP circuit will serve to either increase or decrease charge (representing the synaptic weight) stored on a floating gate, FG.

To modify this charge the compact STDP circuit is split into two symmetric circuit blocks. The weight potentiation, WP, block serves to increase the synaptic weight during a pre-post synaptic spiking event; the weight depression, WD, block serves to decrease the synaptic weight during a post-pre spiking event. The amount of charge which is added to or removed from the FG is proportional to the pulse width, τ_{cg}, of the output from the WP and WD blocks, V_{CG}. Additionally τ_{cg} is proportional to the time difference, Δt_s, between pre- and post- synaptic inputs to the circuit blocks, as defined in Fig. 5.1.

5.2.1 Weight Potentiation Block

Fig. 5.2 presents the weight potentiation block. The circuit increases the synaptic weight, represented as a negative charge stored on a FG of a non-volatile (NVM) device. Weight potentiation only occurs when a presynaptic spike occurs prior to a postsynaptic spike, (pre-post spiking event). The NVM device is represented by its equivalent capacitance C_{FG}. An output buffer is required to provide the required voltage to cause Fowler Nordheim, FN, tunnelling in the NVM to increase the stored charge.

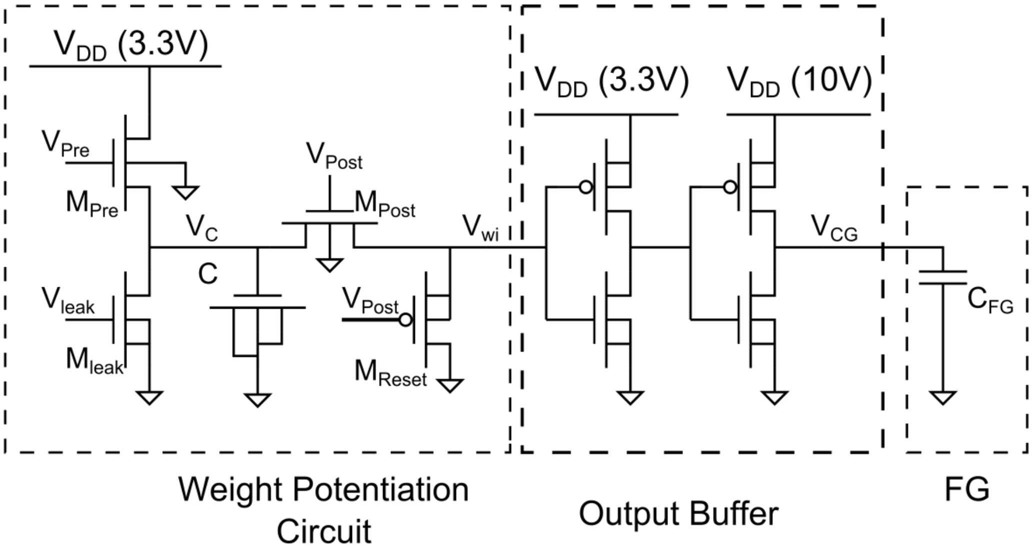


Fig 5.2 – Weight Potentiation, WP, Circuit Block, Output Buffers and NVM Device

The WP circuit comprises of 3 NMOSTs, M_{Pre}, M_{Post} and M_{leak}, 1 PMOST, M_{reset} and a PMOS capacitor, C, Transistor M_{reset} is used to ensure that when V_{post} is low, V_{wi} is pulled low. While V_{post} is high, M_{reset} is off and will not significantly affect V_{wi}.

The theoretical operation of the device is now outlined. The initial conditions when no pre- or post- synaptic spikes occur are;

- $V_{wi,}$ V_{pre} and V_{post} are low
- node V_C is pulled low by M_{leak}
- C discharged

Consider a pre-post spiking event. When a pre-synaptic spike occurs (V_{Pre}), V_C is pulled up to its maximum value, V_M = 3.3V-V_{TMpre}; where it is noted that V_{TMpre} is the threshold voltage of M_{pre}. When the pre-synaptic pulse ends, C starts to discharge via M_{leak}, and V_C decreases at a rate determined by voltage V_{leak}. Voltage V_{leak} thus controls the timing window in which a post-synaptic spike must occur in order to cause the synaptic weight to be increased. When the post-synaptic spike (V_{Post}) occurs, the nodes with voltages V_C and V_{wi}, are connected and V_{wi} is pulled up to V_C-$V_{TMpost}(V_{wi})$; $V_{TMpost}(V_{wi})$ is the threshold voltage associated with M_{post}. The synaptic weight will be increased, while V_{wi} is greater than the trigger voltage of the output buffer, V_{buf}.

Consider now a post-pre spiking event when the circuit is operating under the initial conditions defined above. When a post-synaptic spike occurs, since V_C and V_{wi} are low, no update of the synaptic weight occurs. If a pre-synaptic spike now occurs, it forms the start of a pre-post spiking event.

The output buffer served to supply the 10V required to increase the stored charge on the FG. The WP output buffer is constructed using two CMOS inverters with 3V and 10V V_{DD} rails, as shown in Fig. 5.3. The MOSFETs are sized so as to produce the following operation; if V_{wi} is greater than the trigger voltage of the first CMOS inverter then the output from the second inverter will be pulled up to 10V. If V_{wi} is below the trigger voltage of the first CMOS inverter, then the output from the second inverter is held at ground. The pulse-width, τ_{cg}, and magnitude of V_{CG} determines how much charge is injected and stored on the FG. As $\Delta t_s \rightarrow \Delta t_{s\ min}$, $\tau_{cg} \rightarrow$ max τ_{cg}. Similarly as $\Delta t_s \rightarrow \Delta t_{s\ man}$, $\tau_{cg} \rightarrow$ min τ_{cg}.

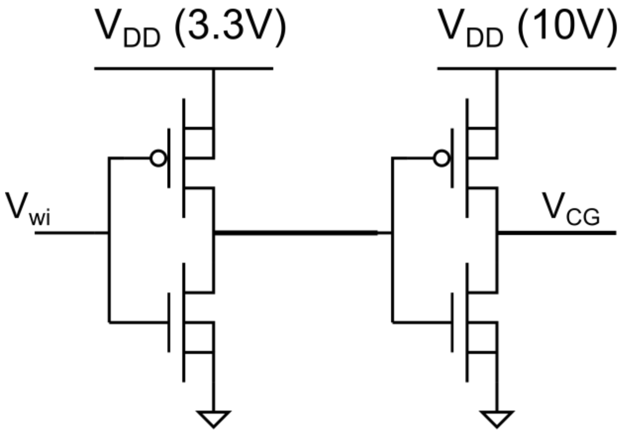


Fig 5.3 – Weight Potentiation Output Buffer

To determine the required sizes of the transistors in each inverter the following output voltages from the circuit are desired;

- V_{CG}= 9.8V when V_{wi} = 3V
- V_{CG} = 0.2V when V_{wi} = 0V

The WP output buffer can be split into two separate inverters. The second inverter needs to be sized such that the following output voltages are present;

- V_{CG} = 0.2V when $V_{in'}$ = 3V
- V_{CG} = 9.8V when $V_{in'}$ = 0V

The additional circuit parameters are as follows;

- V_{DD} = 10V
- V_{tn} = 0.6V
- V_{tp} = -0.8V

The value of $\frac{W_p}{L_p}$ for the inverter is calculated using equation 5.01.

$$\frac{W_p}{L_p} = \frac{2C_{FG}}{K_p t_f (V_{DD} - V_{tp})} \left\{ \frac{(|V_{tp}| - 0.1V_{DD})}{(V_{DD} - V_{tp})} + \ln \left[\frac{2(V_{DD} - V_{tp}) - 0.1V_{DD}}{0.1V_{DD}} \right] \right\} = 1.8 \qquad (5.01)$$

The PMOS transistor is saturated as the gate to source voltage, V_{GSp}=7, and V_{DSp}=10. The drain current in the PMOS device is;

$$I_{Dp} = K_p \frac{W_p}{L_p} \left[\frac{(V_{GSp} - |V_{tp}|)^2}{2} \right] \qquad (5.02)$$

The NMOS is saturated and with its drain current given by;

$$I_{Dn} = K_n \frac{W_n}{L_n} \left[(V_{GSn} - V_{tn})V_{DSn} - \frac{{V_{DSn}}^2}{2} \right] \qquad (5.03)$$

Equating the drain currents gives;

$$K_p \frac{W_p}{L_p} \left[\frac{(V_{GSp} - |V_{tp}|)^2}{2} \right] = K_n \frac{W_n}{L_n} \left[(V_{GSn} - V_{tn})V_{DSn} - \frac{{V_{DSn}}^2}{2} \right] \qquad (5.04)$$

Substituting in $K_n = 170\ \mu A/V^2$ and $K_p = 58\ \mu A/V^2$ gives;

$$170\mu \frac{W_n}{L_n} \left[\frac{(V_{GSp} - |V_{tp}|)^2}{2} \right] = 58\mu \frac{W_p}{L_p} \left[(V_{GSn} - V_{tn})V_{DSn} - \frac{{V_{DSn}}^2}{2} \right] \qquad (5.05)$$

Substituting $\frac{W_p}{L_p} = 1$, into 5.05 gives $\frac{W_n}{L_n} \approx 6$. L_n is chosen to be 0.5µm, such that;

$$W_n = 6L_n = 6x0.5 = 3.0\mu m \qquad (5.06)$$

The first inverter is designed using a similar method, such that the WP output buffer has the dimensions, Fig. 5.4.

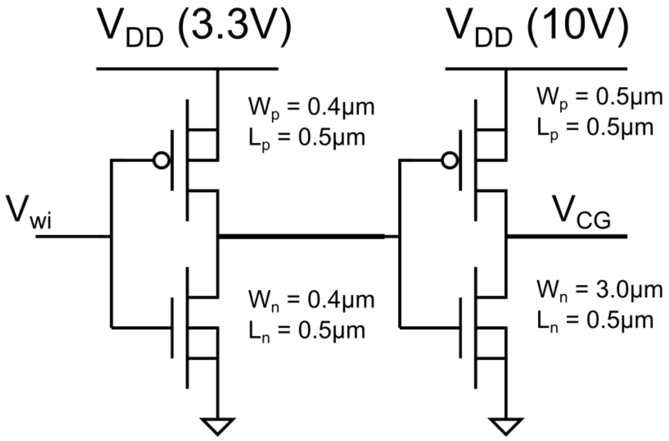


Fig 5.4 – WP Output Buffer showing device dimensions

5.2.2 Weight Depression Block

Fig. 5.5 presents the WD block; its layout is identical to that of the WP block with the exception of the application of pre- and post- synaptic inputs and the design of the output buffer.

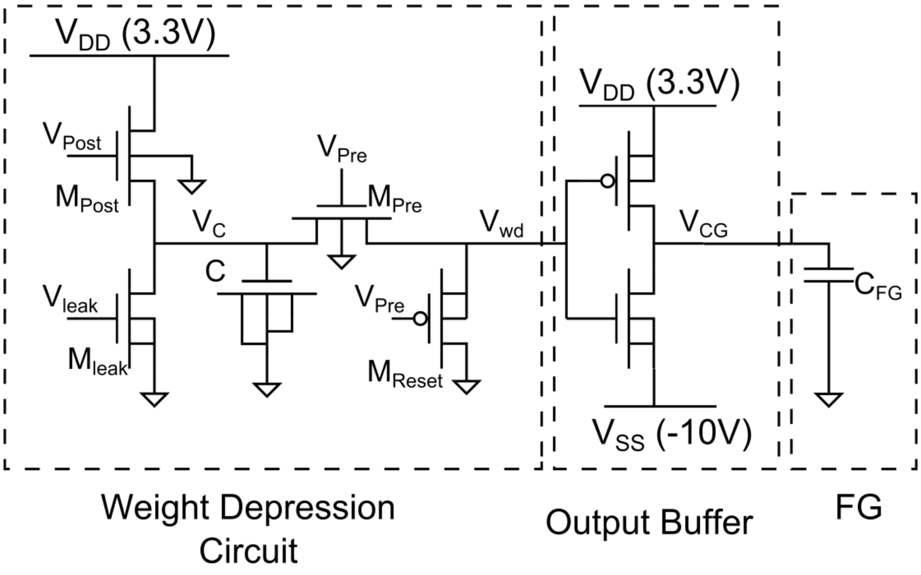


Fig 5.5 – Weight Decrease, WD, Circuit Block, Output Buffers and NVM Device

The operation of the WD block is similar to that of the WP block. However, if post-pre spiking occurs, the synaptic weight will decrease. When a post-synaptic spike (V_{Post}) occurs, V_C is pulled up to its maximum value, V_M = 3.3V-V_{TMpost} and charges capacitor C via M_{Post}. When a pre-synaptic pulse (V_{Pre}) occurs, V_{wd} is pulled up to V_C-V_{TMpost}(V_{wd}) causing the synaptic weight to be decreased over the time period when V_{wd} is greater than V_{buf}.

The WD output buffer serves to supply -10V in order to decrease the charge stored on the FG. It is constructed using a single CMOS inverter with a 3V, supply rail (V_{DD}) and a -10V rail (V_{SS}), Fig. 5.6. The MOSFETs are sized so as to produce the following operation; when V_{wd}, is greater than V_{buf}, its output voltage is pulled down to -10V. If V_{wd}, is less than the V_{buf}, then the output is held at 0V. The design methodology used to determine the transistor sizes is similar to that outlined for the WP output buffer.

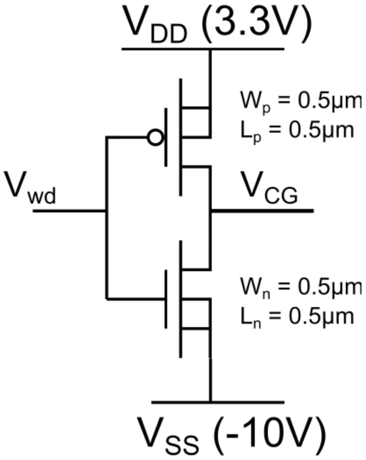


Fig 5.6 – Weight Decrease Output Buffer

For the case of pre-post spiking, when a pre-synaptic spike occurs, V_C and V_{wd} are low and there is no update of the synaptic weight. The occurrence of a post-synaptic spike forms the start of another post-pre spiking event.

If Δt_s = 0, a pre- and post-synaptic spike occurring at the same time then Δw = 0. Both the WP and WD circuits will be 'on' during this event causing node V_{CG} to be biased at 0V.

Therefore it can be shown the in reality the STDP curve which is to be modelled is more akin to that shown in Fig. 5.7 rather than that in Fig. 1(b) [48]. This is consistent with biophysical experiments where it has been reported [49] that synaptic communication between pre and postsynaptic neurons is inherently delayed by axons or dendrite latencies and thus the actual strongest and weakest synapse efficacy do not occur at the absolute temporal difference ($\Delta t_s = 0$).

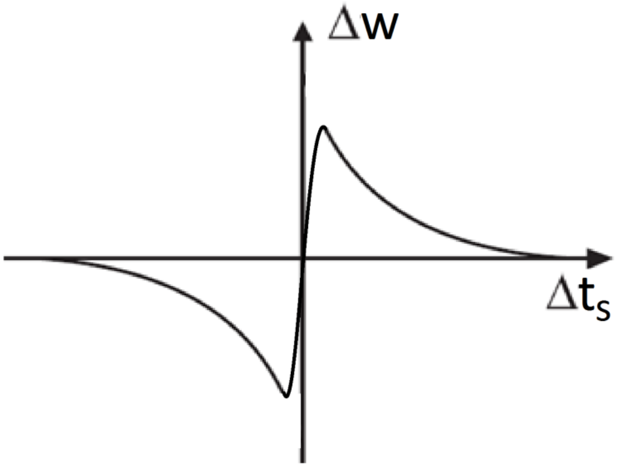


Fig 5.7 – STDP curve showing $\Delta t_s = 0$ gives $\Delta w = 0$

5.2.3 Critical Timing Window

The STDP response implies a critical timing window [7, 14-19]. This critical timing window determines whether synaptic modification will take place within the NN. Experimental results have shown [7, 14] that typically if a postsynaptic spike follows a presynaptic spike within a window of 20-25ms then potentiation takes place. Similarly if a pre-synaptic spike follows a post-synaptic spike within a window of 20-25ms then depression occurs. Outside of this window, no potentiation or depression will occur [7, 14-21]. Typically the critical timing window is between 20-25ms [7, 14].

It was shown in Chapter 2, that STDP can be implemented in hardware [22-31], however few of the implementations presented take into account the variability of the critical timing window for synaptic modification. The proposed STDP circuit outlined in this chapter aims to overcome this short-coming by implementing a variable CTW. Typically in biology the critical timing window is 25ms for potentiation and depression [7, 9]. However, in hardware the computational speed is greatly accelerated, with average spike train frequencies ranging in to the MHz range. We therefore implement an equivalent timing window of 20-25µs in this work.

The critical timing window, t_{cw}, is defined here by the time it takes for V_C to fall from V_M to V_{buf} for both the WP and WD blocks of Figs. 5.2, 5.5. The rate at which the sub-threshold current reduces V_C is set by V_{leak} and the aspect ratio of M_{leak}, S_{Mleak}. The sub-threshold current, I_{leak} is constant for V_{DS}> 3kT/q [47];

$$I_{leak} = I_0 exp\left[\frac{q(V_{leak}-V_t)}{mkT}\right] \tag{5.07}$$

where V_t is the threshold voltage of M_{leak}, q is the charge of an electron, k is the Boltzmann constant, m is…. and T is absolute temperature. I_o is defined as;

$$I_0 = \mu_{eff} C_0 S_{Mleak}(m-1)\left[\frac{kT}{q}\right]^2 \tag{5.08}$$

$$m = 1 + \frac{C_d}{C_0} \tag{5.09}$$

C_d is the depletion layer capacitance, C_o is the capacitance of the oxide in Fm^{-2}, μ_{eff} is the effective channel mobility. Now

$$dt = -\frac{C}{i} dV_c \tag{5.10}$$

Where *i* is the discharging current, I_{leak}. Integrating (5.10) with voltage limits, gives equation (5.11) which can be used to determine the critical timing window, t_{cw}. The window can be adjusted using V_{leak}, according to:

$$t_{CW} = \frac{C}{I_{leak}}[0.8V_M] \tag{5.11}$$

Substituting equation (5.07) into (5.11) and rearranging allows a value for V_{leak} to be calculated for the required t_{CW}. For t_{CW} = 25µs, a V_{leak} ≈ 400mV is required. The effects of process variation upon the critical timing window will be discussed later in this chapter.

5.2.4 Results

5.2.4.1 Output Buffers

Simulation and experimental results showing the operation of the WP output buffer are presented in Fig. 5.8. The results are obtained by applying a 3.0V, 10µs wide pulse to the input of the WP output buffer, V_{in}. The rise and fall time of the pulse are both equal to 1µs. The results confirm that while V_{in} is greater than V_{buf}, the output from the buffer, V_{CG}, is pulled high to 10V (simulation) and 9.81V (measured). In addition to this the pulse width is found to be ≈10µs for both the simulation and experimental results. The experimental results also indicate that while V_{in} = 0V, V_{CG} has a resting voltage of -0.32V.

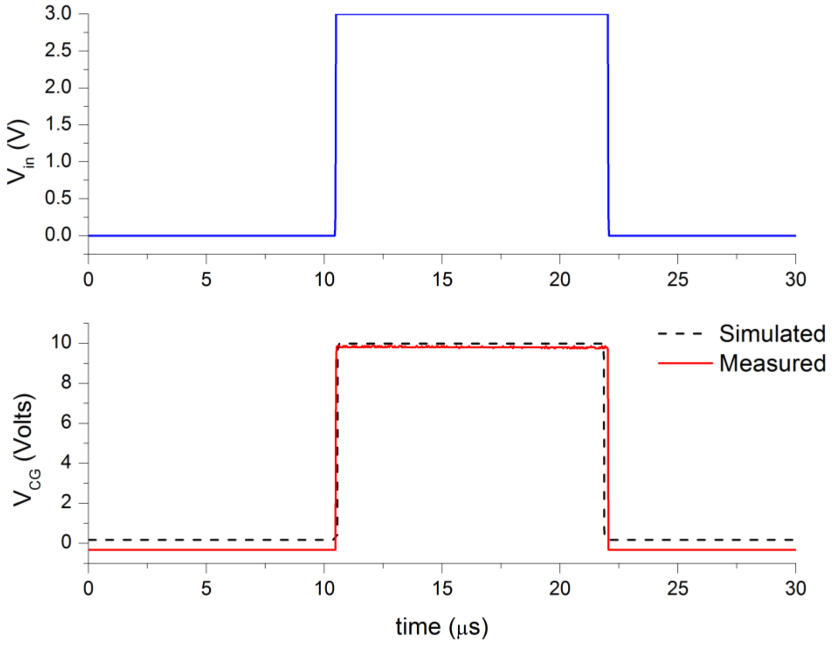


Fig 5.8 – WP Output Buffer Response

In addition to the WP output buffer, Fig. 5.9 presents the simulated response of the WD output buffer. In this case a 3.0V, 10μs pulse to the input of the WD output buffer, V_{wd}. The rise and fall time of input pulse remains at 1μs. The results confirm that while V_{wd} is greater than V_{buf}, the output from the buffer, V_{CG}, is pulled down to -10Vwith a constant pulse width of 10μs. In the absence of V_{wd}, V_{CG} remains at 0V.

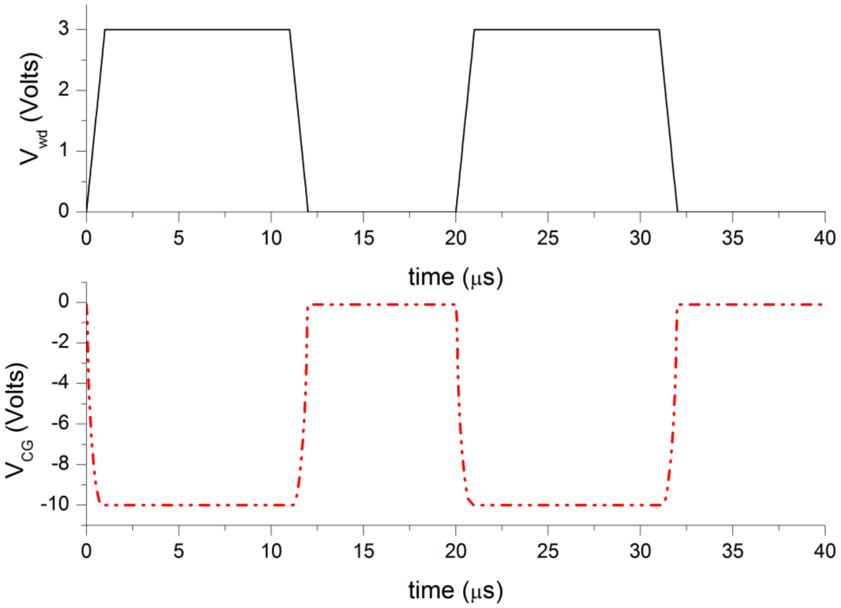


Fig 5.9 – WD Output Buffer Simulated Response

5.2.4.2 STDP Circuit

In this section, simulation and experimental results are presented which confirm that the operation of the STDP circuit is consistent with the theory presented in the previous section. Simulations of the back-annotated Virtuoso layout of the WP block, including parasitic circuit capacitances, were undertaken using Cadence IC Design Tools 5.1.41 with SpectreS SPICE. Experimental results are taken from circuits fabricated in AMS 0.35μm CMOS process. Unless otherwise stated, V_{leak} is set to 410mV, C_w is 100fF and S_{Mleak} = 1 such that the t_{CW} ≈ 20μs, equation (5.11). Additional parameters for the circuit are; W_{Mpre} = L_{Mpre} = 0.5μm, W_{Mreset} = L_{Mreset} = 0.5μm, W_{Mpost} = 0.4μm L_{Mpost} = 0.35μm.

As outlined earlier in this chapter, one of the main features of STDP circuit is that a variable critical timing window, CTW, can be implemented. The CTW is determined by two main variables, C_w and V_{leak}. However C_w must be chosen prior to fabrication. C_w is chosen such that the overall circuit is compact but also allows for a large range of values for the CTW. Therefore the CTW is solely dependent upon the user adjustable V_{leak}.

Fig. 5.10 presents a plot for the duration of the critical timing window based upon theory, simulation and experimental results. The theoretical results are obtained by using equations 5.07 and 5.11 with V_M = 2.5V and V_{tMleak} = 0.5V. The results presented in the figure show that the theoretical equation concurs with both simulation and experimental results for the decrease in V_C due to the subthreshold current through transistor M_{leak}. For V_{leak} < 400mV, the theoretical equation gives approximately the same results as both the simulation and experimental data. For V_{leak} ≥400mV a slight discrepancy between theoretical and simulation/experimental results occur. This is due to the both simulation and experimental data taking into account various non-ideal effects as discussed in chapter 3, equation 5.11 predicts the ideal variation of V_C over time.

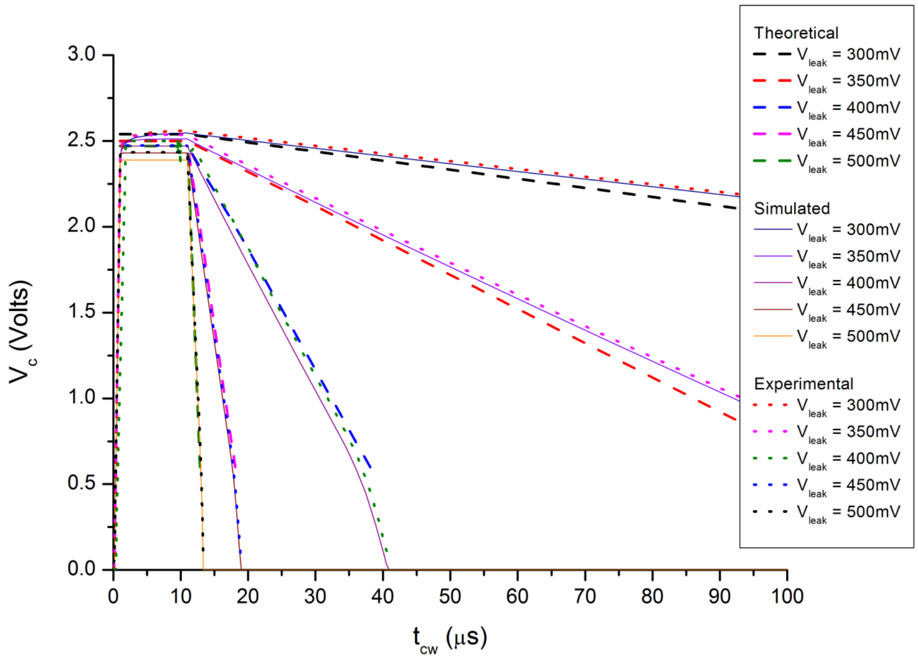


Fig 5.10 – Critical Timing Window duration from theory, simulation and experimental

As mentioned earlier, typically in a biological synapse, any change in the synaptic weight occurs within a critical timing window of t_{cw} ≈ 20-25ms. Substituting equation (5.07) into (5.11) and rearranging allows a value for V_{leak} to be calculated for the required t_{cw}. For a biologically equivalent critical timing window of t_{cw} = 20μs, V_{leak} = 410mV is required.

Fig. 5.11 presents the results of a post-pre spiking event. A pre-synaptic spike, V_{Pre}, occurs at Δt_s = 5μs, after the end of the postsynaptic spike, V_{Post}. Both the simulation and experimental results confirm the theoretical operation outlined previously. When V_{Post} occurs, C remains discharged with V_C = 0V as per the initial conditions. Thus $V_{wi} \approx 0V$ and $V_{CG} \approx 0V$, and no update of the synaptic weight takes place. V_{Pre} goes high 5μs after the end of V_{Post} causing C to charge and V_C to be pulled up to ≈ 2.4V. When V_{Pre} goes low; C discharges via subthreshold transistor M_{leak} as indicated earlier, causing V_C to discharge linearly. Since node V_C and V_{wi} remain unconnected no change in synaptic weight is seen. This satisfies the condition in the STDP rule that Post-pre spiking does not cause an increase in the synaptic weight.

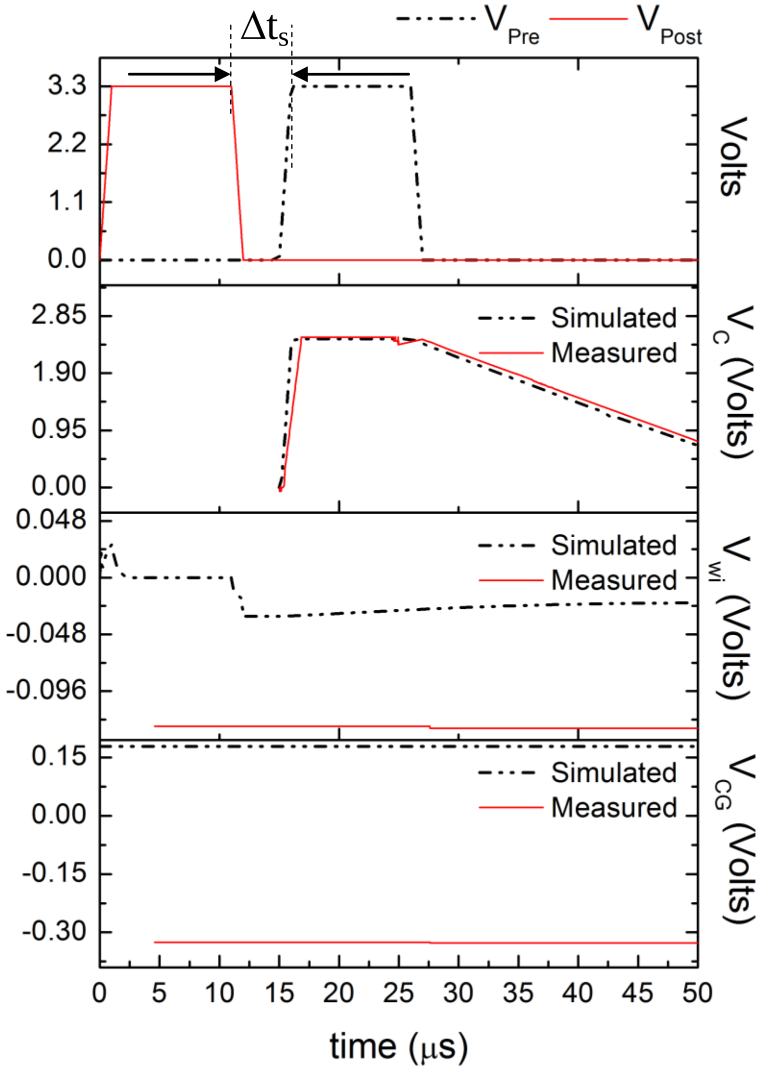
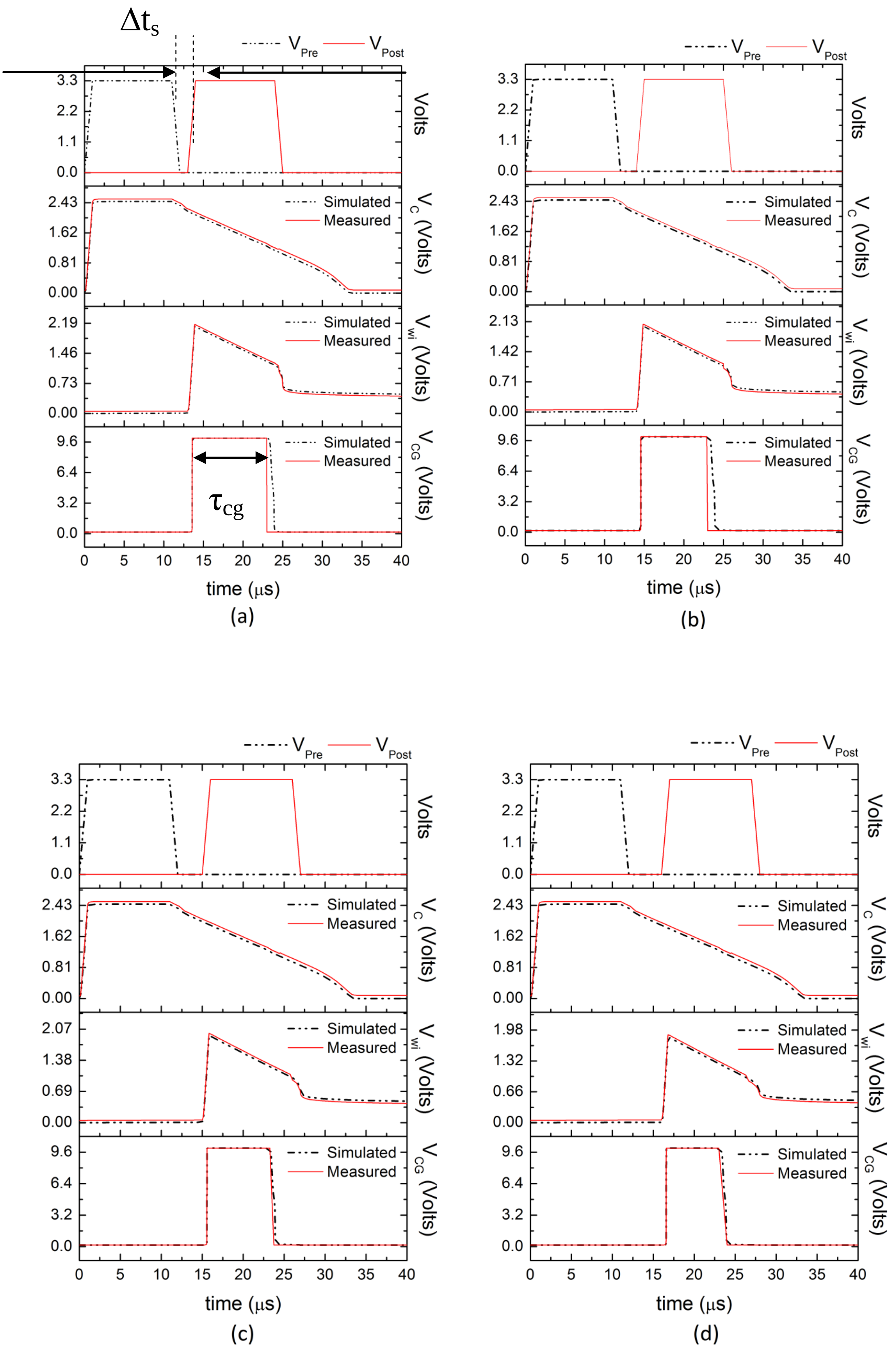


Fig 5.11 – Post-pre spiking event, Δt_s = 5μs



(a)

(b)

(c)

(d)

Fig 5.12 – Pre-post spiking events, (a) $\Delta t_s = 1\mu s$; (b) $\Delta t_s = 3\mu s$; (c) $\Delta t_s = 4\mu s$; (d) $\Delta t_s = 5\mu s$.

Consider now the effects of a series of pre-post synaptic spikes upon the WP circuit, where Δt_s is increased from 1µs to 20µs, Figs 5.12, 5.13 and 5.14.

Referring to Fig. 5.12(a), as V_{pre} is pulled high, C is charged to voltage V_M = 2.43V for both simulation and experimental results, therefore a voltage drop of 0.87V is seen across transistor M_{Pre}. When V_{pre} goes low, C discharges (initially) linearly via M_{leak} which is operating in sub-threshold. When V_{post} goes high, V_{wi} is pulled up to V_C - $V_{tMpost}(V_{wi})$, for both simulation and experimental results $V_{wi} \approx 1.70$V. Transistor M_{post} sees a voltage drop of $\approx$ 0.73V. A weight increase is triggered as V_{CG} is pulled high to 10V. It should be noted that V_{tMpost} is shifted due to substrate bias effects. Fig. 5.12(a) shows that when V_{post} goes low, both V_{wi} and V_{CG} are pulled low, ending the synaptic weight update. This is consistent with the theoretical operation outlined previously. It should be noted that V_{wi} is only pulled down to about V_t after V_{post} returns to 0V, as the pMOST, M_{reset}, has poor pull-down capability.

The pulse-width and magnitude of V_{CG} will determine how much charge is injected and stored on the FG. The time difference between pre- and post- synaptic spikes is Δt_s=1µs. For Δt_s =1µs, the maximum weight update occurs, $\Delta w = \Delta w_{max}$. This occurs as V_{wi} is above the threshold voltage of the output buffer, while V_{post} is still high. Thus V_{CG} is at its maximum pulse width, τ_{cg} = 10.91µs (simulation) and has a measured value of τ_{cg} = 10.75µs. In both cases V_{CG} has a magnitude of 10V.

In Fig. 5.12(b), Δt_s is now increased to 3µs. As with the previous results, C is charged to V_M, and when the postsynaptic spike occurs, $V_{wi} \approx 1.70$V. This again causes V_{CG} to be pulled high to 10V. However τ_{cg} is reduced compared to Δt_s =1µs, τ_{cg} is now 8.91µs (simulation) and 8.62µs (measured). The reduction in τ_{cg} occurs because V_{post} coincides with the linearly decreasing V_C. Voltage V_{wi} now tracks the decreasing V_C, until, eventually V_{wi} is pulled below the threshold voltage of the first CMOS inverter, while V_{post} is still high, Fig. 5.12(b). Further increasing Δt_s to 4µs and 5µs, Fig 5.12(c) and (d), continues to reduce τ_{cg} to 7.90µs (simulation), 7.62µs (experimental) and 6.90µs (simulation), 6.61µs (experimental) respectively.

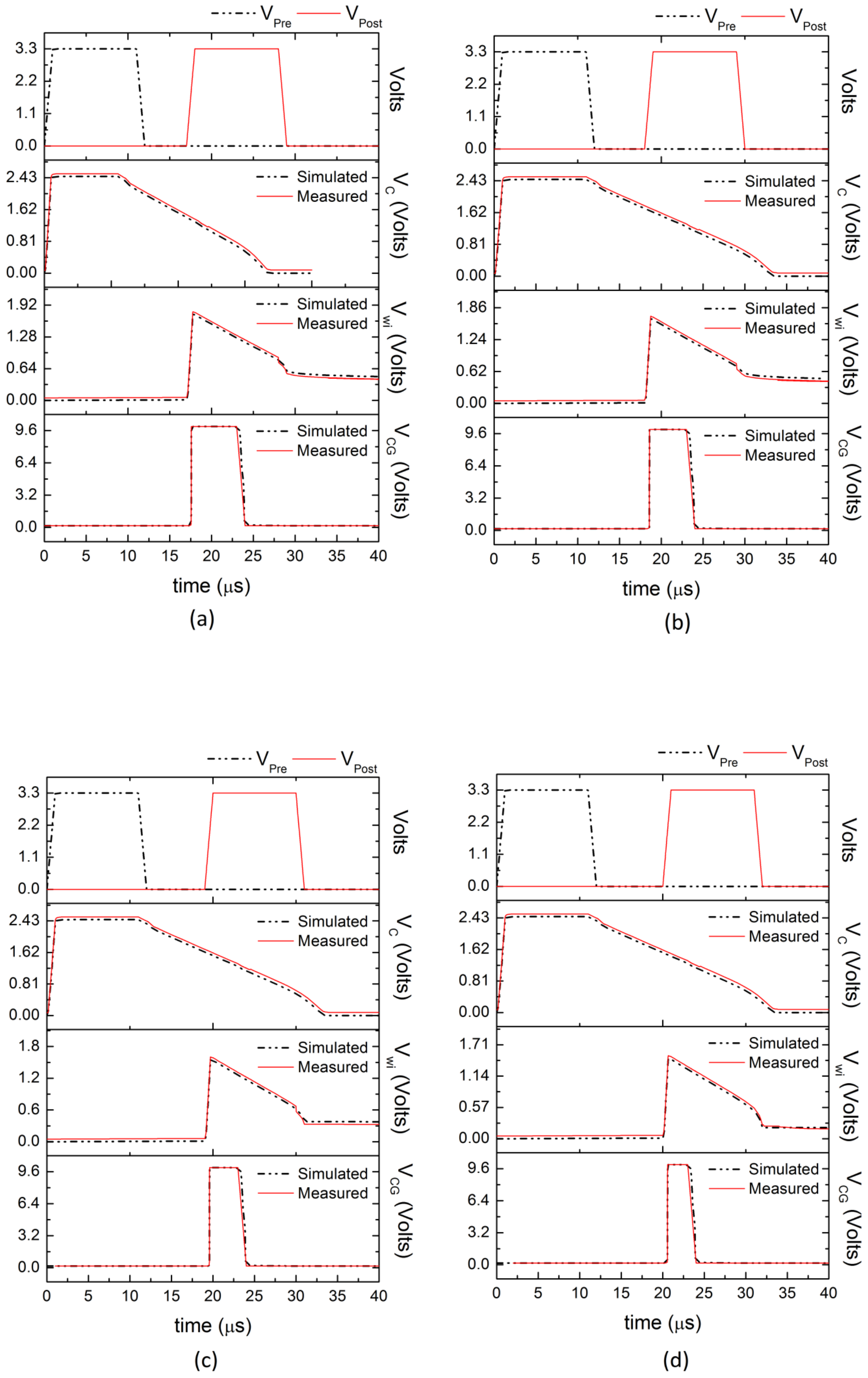


Fig 5.13 – Pre-post spiking events, (a) Δt_s = 6μs; (b) Δt_s = 7μs; (c) Δt_s = 8μs; (d) Δt_s = 9μs.

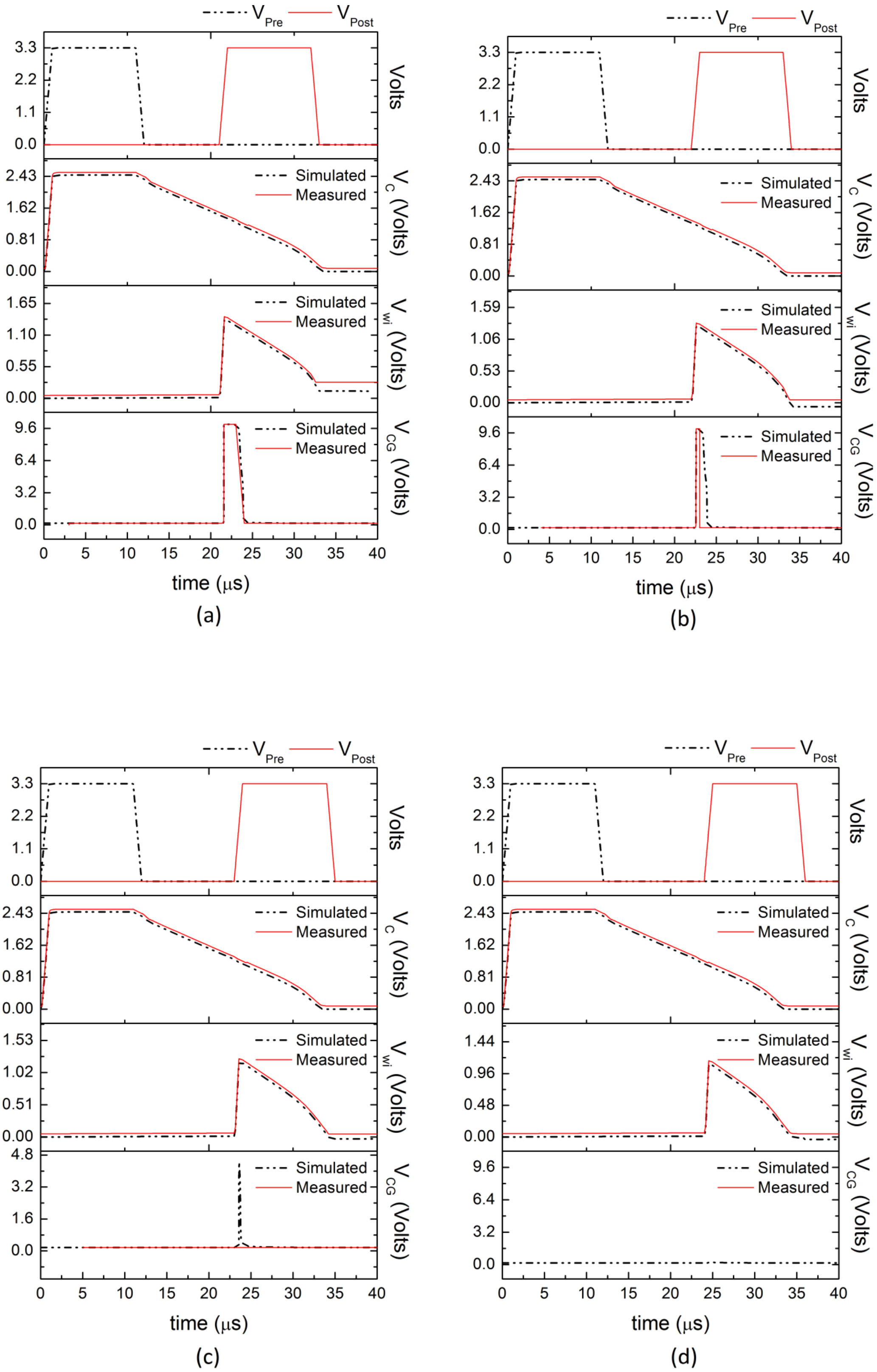


Fig 5.14 – Pre-post spiking events, (a) Δt_s = 10μs; (b) Δt_s = 11μs; (c) Δt_s = 12μs; (d) Δt_s = 13μs.

Fig. 5.13 shows the effect of further increasing Δt_s, with (a) Δt_s = 6μs, (b) Δt_s = 7μs, (c) Δt_s = 8μs, (d) Δt_s = 9μs. In each case τ_{cg} is further reduced to (a) 5.9μs (simulation), 5.59μs (experimental), (b) 4.92μs (simulation), 4.60μs (experimental) (c) 3.91μs (simulation), 3.60μs (experimental) (d) 2.90μs (simulation), 2.61μs (experimental). Finally in Fig. 5.14 (a) Δt_s = 10μs; (b) Δt_s = 11μs; (c) Δt_s = 12μs and (d) 17μs respectively. In Fig. 5.13 (b), $\tau_{cg} \approx$ 0.91μs and 0.65μs for simulation and experimental respectively. The magnitude of V_{CG} is slightly reduced to 9.6V. This corresponds to the minimum weight update $\Delta w = \Delta w_{min}$. Fig. 5.13(c) and (d) indicate that once $\Delta t_s \geq$ 12μs then no update in the synaptic weight takes place as $V_{CG} \approx 0$ due to V_{wi} being less the threshold voltage of the first CMOS inverter when V_{post} is high.

Fig. 5.15 presents a plot of Δt_s against τ_{cg} and represents the upper right-hand quadrant of the typical STDP curve, shown as the insert. Fig. 5.15 indicates that as Δt_s is increased τ_{cg} decreases from 10.91μs to ≈0.91μs (simulation), from 10.75μs to ≈0.65μs (measured). It should be noted that Δt_s has the equation form of $y = -mx + c$, it is sufficient approximation of STDP, as the exponential functions of the STDP curve are not a pre-requisite for the implementation of STDP but rather a mathematical convenience. Hence the WP block of the compact STDP circuit can be used to implement STDP in hardware.

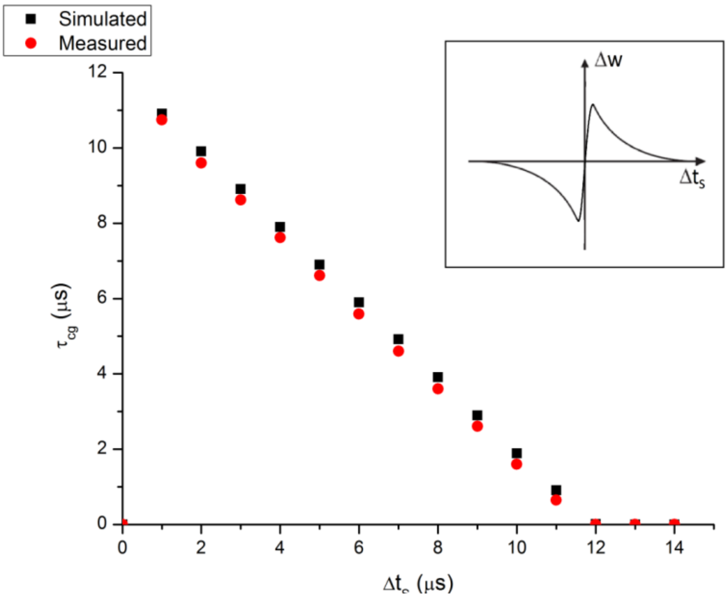


Fig 5.15 – Pre-post spiking STDP Curve – Simulated and Experimental. Insert – Asymmetric STDP Curve

Having presented simulation and experimental results for the WP circuit block, the following series of figures present simulation results undertaken on the WD circuit block from both pre-post spiking and post-pre spiking. Simulations of the back-annotated Virtuoso layout of the WD block, including parasitic circuit capacitances, were undertaken using Cadence IC Design Tools 5.1.41 with SpectreS SPICE. Unless otherwise stated, V_{leak} is set to 400mV, C is 100fF and S_{Mleak} = 1 such that the $t_{cw} \approx 25\mu s$, equation (5.11). Additional parameter for the circuit are; $W_{Mpre} = L_{Mpre} = 0.5\mu m$, $W_{Mreset} = L_{Mreset} = 0.5\mu m$, W_{Mpost} = 0.4µm L_{Mpost} = 0.35µm.

As the WD circuit block is identical to the WP circuit with the exception of the application of V_{pre} and V_{post} its operation is also identical. Fig. 5.16 presents the results of a pre-post spiking event on the WD block. A post-synaptic spike, V_{Post}, occurs 5µs, $\Delta t_s = 5\mu s$, after the end of the presynaptic spike, V_{Pre}. The simulation confirms the theoretical operation outlined previously. When V_{Pre} occurs, C remains discharged with $V_C = 0V$ as per the initial conditions. Thus $V_{wi} \approx 0V$ and $V_{CG} \approx 0V$, and the synaptic weight remains unchanged. V_{Post} goes high 5µs after the end of V_{Pre} causing C to charge and V_C to be pulled up to ≈ 2.4V. When V_{Post} goes low; C discharges via subthreshold transistor M_{leak} causing V_C to discharge linearly. Since node V_C and V_{wi} remain unconnected no change in synaptic weight is seen. This satisfies the condition in the STDP rule that pre-post spiking does not cause a decrease in the synaptic weight.

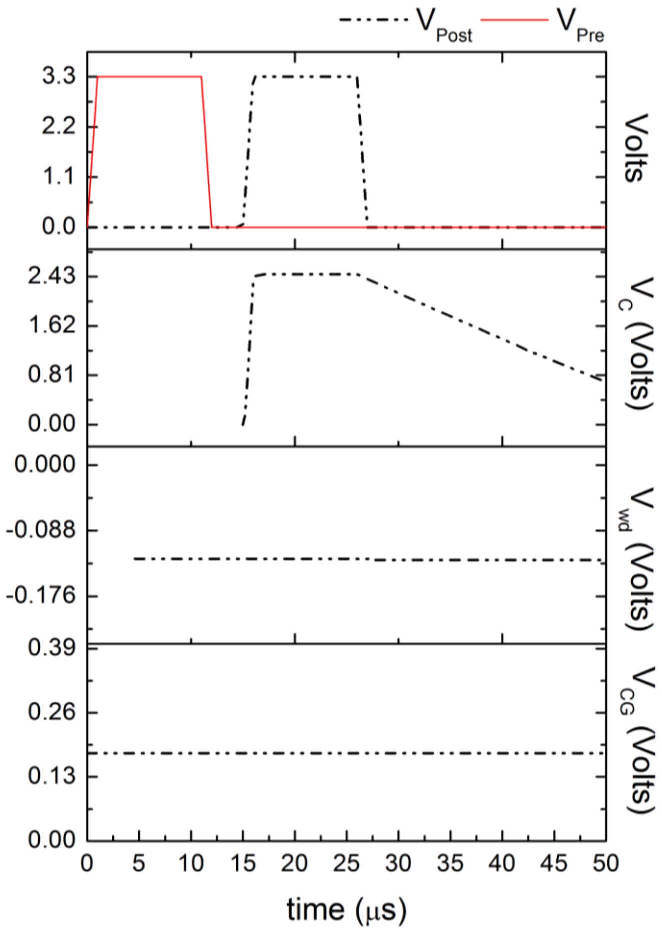


Fig 5.16 – Pre-post spiking event – Δt_s = 5μs

Figures 5.17 and 5.18 present the simulation results for a series of post-pre spiking events upon the WD circuit. Δt_s is once again increased from 1μs to 20μs. Referring to Fig. 5.17(a), as V_{post} is pulled high C is charged to voltage V_M = 2.43V. As V_{pre} goes low, C discharges (initially) linearly via M_{leak}. When V_{pre} goes high, nodes V_C and V_{wi} are connected such that $V_{wi} \approx$ 1.70V. A weight decrease is triggered as V_{CG} is pulled down to -10V. V_{pre} goes low, both V_{wi} and V_{CG} are pulled back to 0V, ending the synaptic weight update. This is consistent with the theoretical operation outlined previously.

For Δt_s=1μs, the maximum value of the weight decrease occurs, $\Delta w = \Delta w_{max}$. V_{CG} is at its maximum pulse width; τ_{cg} = 11.31μs and magnitude, V_{CG} = -10V. Further increasing Δt_s, (b) Δt_s = 5μs, (C) Δt_s = 7μs, (d) Δt_s = 8μs. In each case τ_{cg} is further reduced to (b) 8.14μs, (c) 6.16μs and (d) 5.15μs respectively.

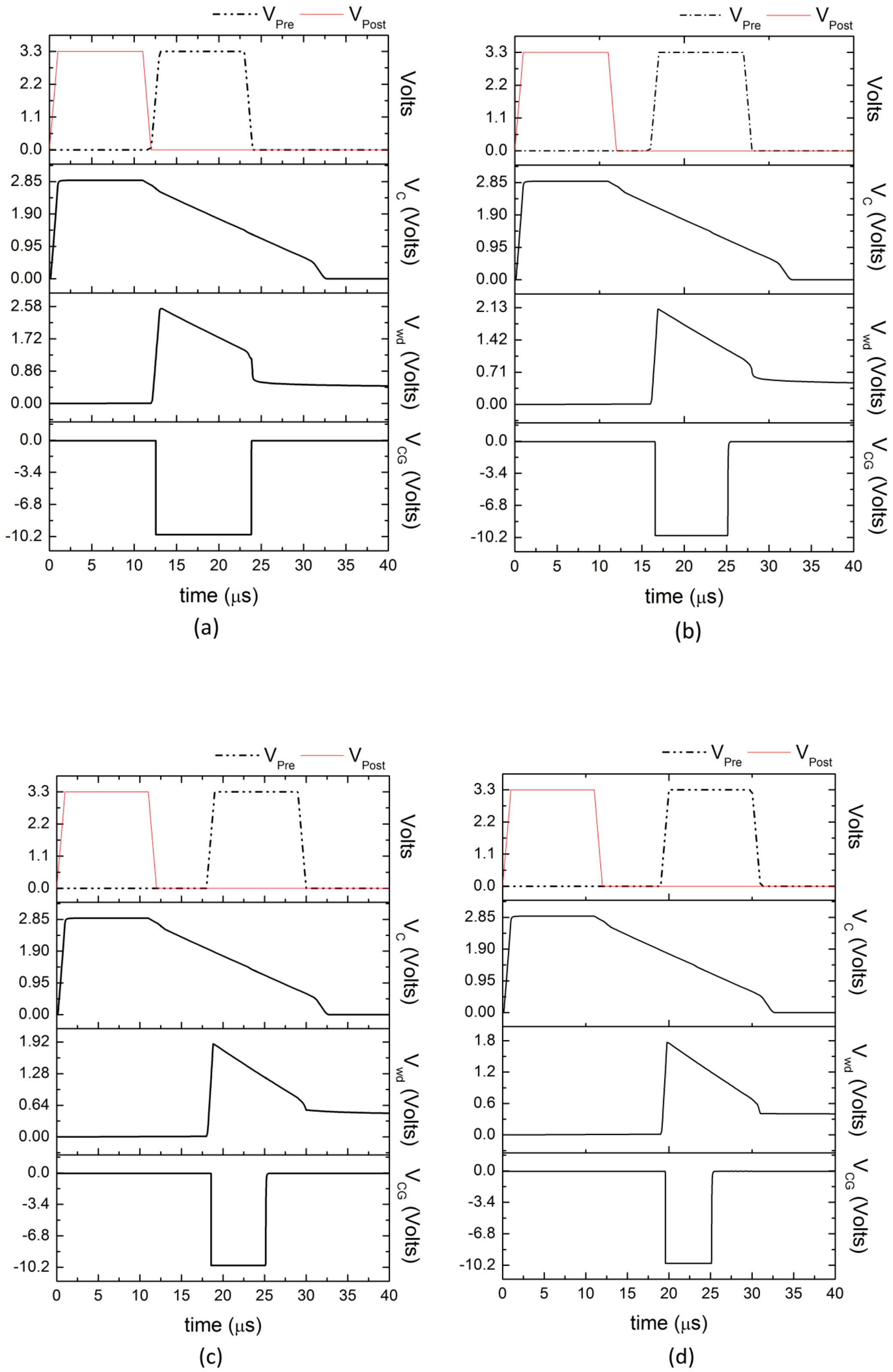


Fig 5.17 – Post-pre spiking events, (a) Δt_s = 1µs; (b) Δt_s = 5µs; (c) Δt_s = 7µs; (d) Δt_s = 8µs.

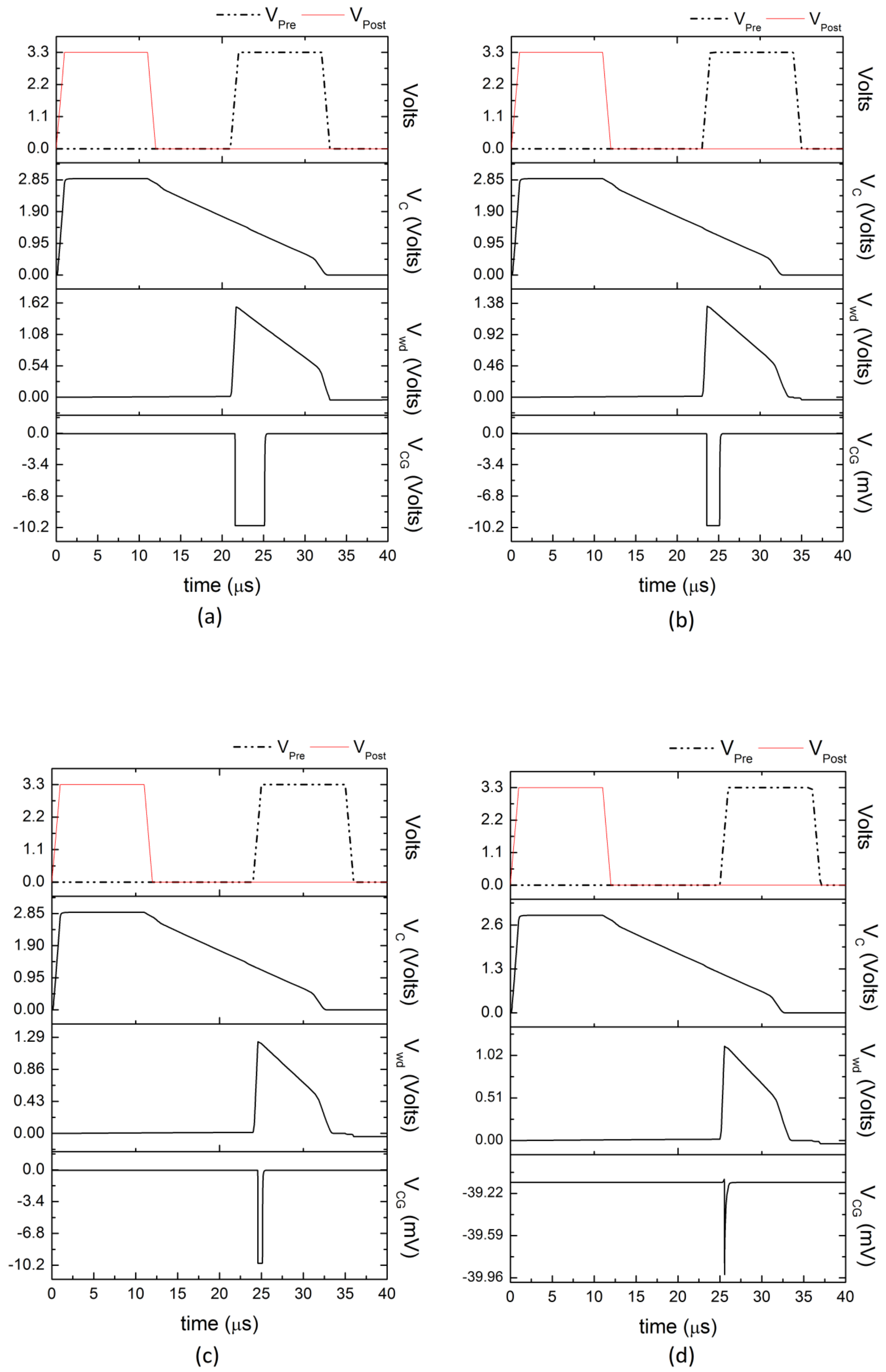


Fig 5.18 – Post-pre spiking events, (a) Δt$_s$ = 10μs; (b) Δt$_s$ = 11μs; (c) Δt$_s$ = 12μs; (d) Δt$_s$ = 13μs.

Finally in Fig. 5.18 (a) Δt_s = 10µs, $\tau_{cg} \approx$ 3.12µs; (v) Δt_s = 11µs, $\tau_{cg} \approx$ 2.06µs. In Fig.5.18 (c) Δt_s = 12µs gives $\tau_{cg} \approx$ 0.96µs, and the magnitude of V_{CG} is slightly reduced to 9.6V. This corresponds to the minimum weight update $\Delta w = \Delta w_{min}$. Fig. 5.18(d) indicates that once $\Delta t_s \geq$ 13µs then no update in the synaptic weight takes place as $V_{CG} \approx 0$ due to V_{wd} being less the threshold voltage of the first CMOS inverter when V_{pre} is high.

Fig. 5.19 presents a plot of Δt_s against τ_{cg} and represents the lower left-hand quadrant of the typical STDP curve, shown as the insert. Fig. 5.19 indicates that as Δt_s increase negatively, τ_{cg} decreases from 11.31µs to ≈0.50µs. This is deemed to be a sufficient approximation of STDP. The exponential functions of the STDP curve are not a pre-requisite for the implementation of STDP but rather a mathematical convenience. Hence the WD circuit block of the compact STDP circuit can be used to implement STDP in hardware.

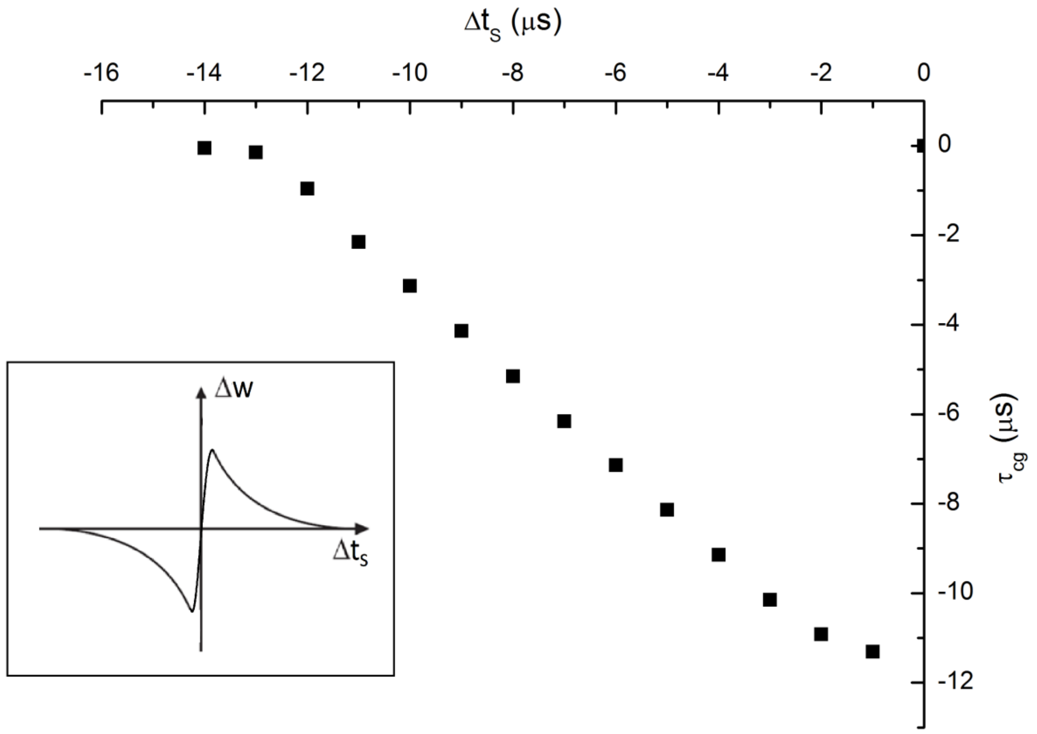


Fig 5.19 – WD STDP Curve. Insert – Asymmetric STDP Curve

Having presented simulation and experimental results for the STDP circuit, it is now possible to construct the full STDP curve for the circuit. Fig. 5.20 presents a plot of τ_{cg} against Δt_s and represents the full STDP curve, shown as the insert.

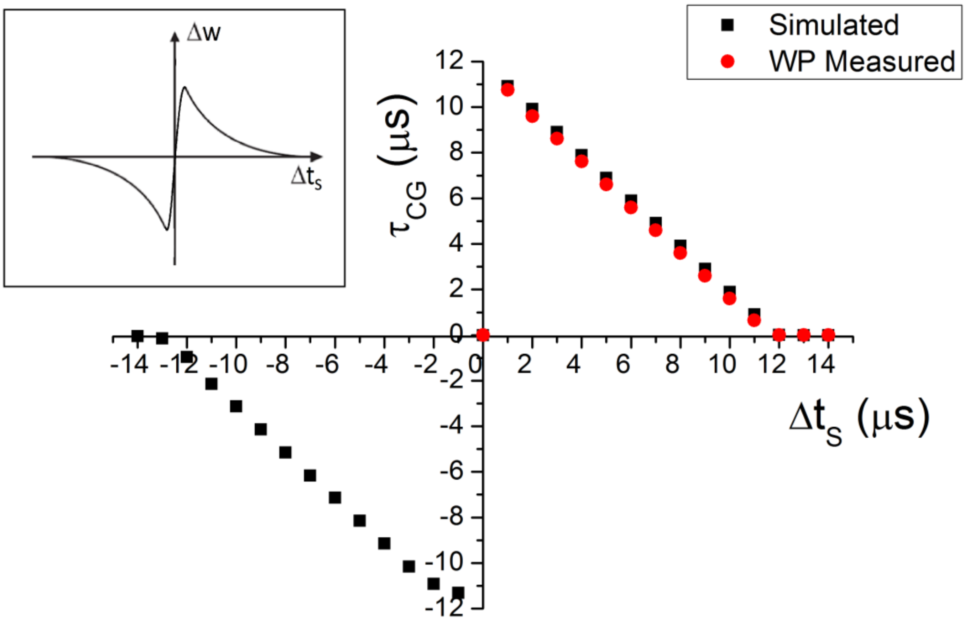


Fig 5.20 – Full STDP Curve – Simulated and Experimental (WP Circuit only)

The STDP circuit is to be used with FG devices, therefore the sensitivity of the weight charge injection to the FG is now considered, in relation to the STDP curve presented in Fig. 5.20 and charging time. In chapter 4 a NVM device was outlined which can be used to store negative charge, on a FG. This charge is used to represent the synaptic weight of an associated synapse, and the charge injected onto the FG represents the change in the associated weight, $Q_{inj} \equiv \Delta w$. In addition to this a model was presented which determines the charge storage (and removal) characteristics of the NVM device.

The charge injected onto the FG represents the change in the associated weight; $Q_{inj} \equiv \Delta w$. The charge is injected by the Fowler-Nordheim mechanism [46].

$$J_{FN} = C_0 \frac{dV_{ox}}{dt} = A{E_{ox}}^2 exp\left(\frac{-B}{E_{ox}}\right) \tag{5.12}$$

Constants A and B are given by equations 5.13 and 5.14 respectively:

$$A = 2x10^{-6}A/V^2 \tag{5.13}$$

$$B = \frac{4}{3}\frac{\sqrt{2m_{ox}}}{qh}\phi_B{}^{3/2}\ V/cm \tag{5.14}$$

Where m_o is the mass of an electron at rest, m_{ox} is the effective mass of an electron (m_{ox} = $0.5m_o$) in the insulator and ϕ_B is the barrier height for injection from semiconductor to oxide.

Fig. 5.21 presents the cross-section of a FG device constructed using a poly-silicon capacitor. The charge injected onto the FG, Q_{inj}, can be found from consideration of the current in the thin tunnelling oxide, t_{ox} over a set period of time, Δt. The following equations (5.16 – 5.20) are used to determine Q_{inj} and the associated potential of charge stored on the FG, ΔV_w.

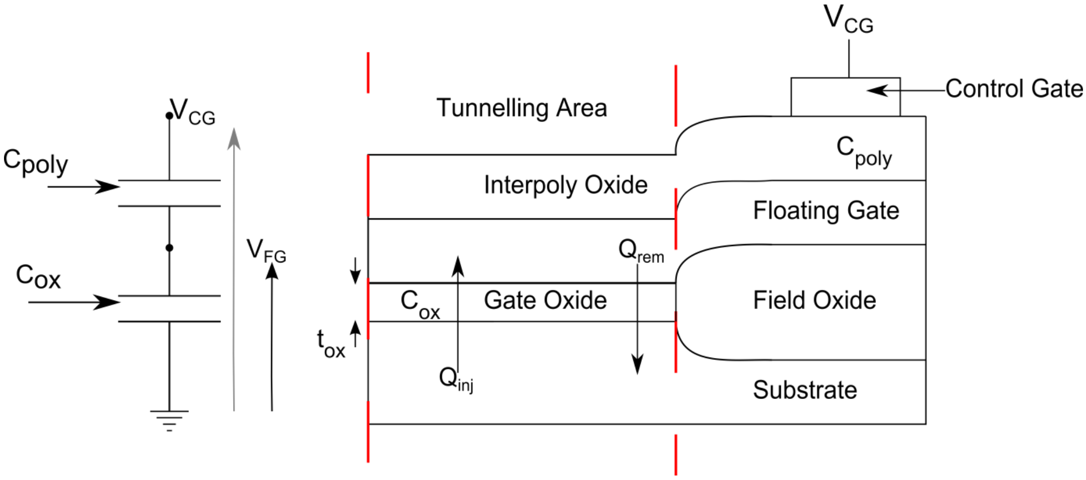


Fig. 5.21 – Equivalent capacitor diagram and cross section of FG device, constructed using poly-silicon and MOS capacitors. C_{poly} is the capacitance of the interpoly oxide, C_{ox} is the capacitance of the tunnelling oxide, V_{CG} and V_{FG} are the voltages applied to the control gate and coupled onto the FG respectively Q_{inj} represents the charge stored on the FG and Q_{rem} represents the charge removed from the FG, both due to FN tunnelling.

The capacitively coupled voltage, as defined in Fig.5.21, which falls across t_{ox} is given by 5.16.

$$V_{FG} = \alpha V_{CG} \tag{5.16}$$

Where α is the capacitive coupling coefficient, defined as $\alpha = \frac{C_{poly}}{C_{ox}+C_{poly}}$. The electric field in the oxide, E_{ox} is given by 5.17, assuming that there is no charge in the oxide or initially stored on the FG

$$E_{ox} = \frac{V_{FG}-\phi_s}{t_{ox}} \tag{5.17}$$

where V_{FG} is the potential of the FG and ϕ_s is the surface potential at the oxide-semiconductor interface. The field at successful time steps, Δt, can be found from Equation 5.18 (see chapter 4 for derivation).

$$E_{ox(i+1)} = B\left[ln\left(\Delta t\frac{AB}{t_{ox}C_0} + exp\left(\frac{B}{E_{ox}(i)}\right)\right)\right]^{-1} \tag{5.18}$$

The potential, ΔV_w associated with the stored charge is calculated by finding the difference between successive steps of field:

$$\Delta V_w = t_{ox}(E_{ox}(i) - E_{ox}(i+1)) \tag{5.19}$$

The charge per unit area injected onto the FG for the duration of the pulse width Δt_s is then found as;

$$\Delta w \propto Q_{inj} = C_0\Delta V_w \tag{5.20}$$

Fig. 5.22 presents plots of (a) Δw against Δt_s and (b) ΔV_w against Δt_s generated using equations 5.20 and 5.19 respectively. Fig. 5.24 presents plots of (a) Δw against Δt_s and (b) $V_{\Delta w}$ against Δt_s generated using equations 5.19 and 5.20 respectively for WP circuit based upon simulated and experimental values of Δt_s.

Fig. 5.22 (a) presents the Q_{inj} (Δw) STDP curve for increasing tunnelling area. The increment of charge injected decreases for increasing Δt because the stored charge serves to reduce the electric field. Similarly as Δt_s is decreased below -1µs, the amount of charge removed is also decreased. The results indicate that Q_{inj} (Δw) (and ΔV_w) tracks τ_{cg} due to the similar shape of the Q_{inj} (Δw) (and ΔV_w) v Δt_s and τ_{cg} v Δt STDP plots. Increasing the device tunnelling area causes a shift in the STDP curve. Specifically this is a shift in the magnitude of the charge injected/removed for the same Δt value.

For the minimum tunnelling area presented, 0.35µm x 0.4µm the maximum amount of charge injected is ≈ 11.2 nC/cm^2, which corresponds to a potential of ΔV_w = 23mV. The maximum amount of charge removed is ≈ 12.7 nC/cm^2 (ΔV_w = 26.2mV). These occur for Δt_s = 1µs and -1µs respectively. Similarly the minimum amount of charge injected is ≈ 0.825 nC/cm^2 (ΔV_w = 2.0mV) and the minimum amount of charge removed is ≈ 0.45 C/cm^2 (ΔV_w = 1.9mV). Again these occur at Δt_s = 11µs and -13µs respectively.

For the largest tunnelling area presented, 2.5µm x 2.5µm the maximum amount of charge injected is ≈ 50.1 nC/cm^2, (ΔV_w = 103mV) and the maximum amount of charge removed is ≈ 56.1 nC/cm^2, (ΔV_w = 113mV). These occur for Δt_s = 1µs and -1µs respectively. The minimum amount of charge injected is ≈ 2.2 nC/cm^2, (ΔV_w = 4mV). Similarly the minimum amount of charge removed is ≈ 1.2 nC/cm^2, (ΔV_w = 2.5mV). Again these occur at Δt_s = 11µs and -13µs respectively. The results presented indicated that the charge injection characteristics for LTP are approximately symmetrical to the charge removal characteristics for LTD.

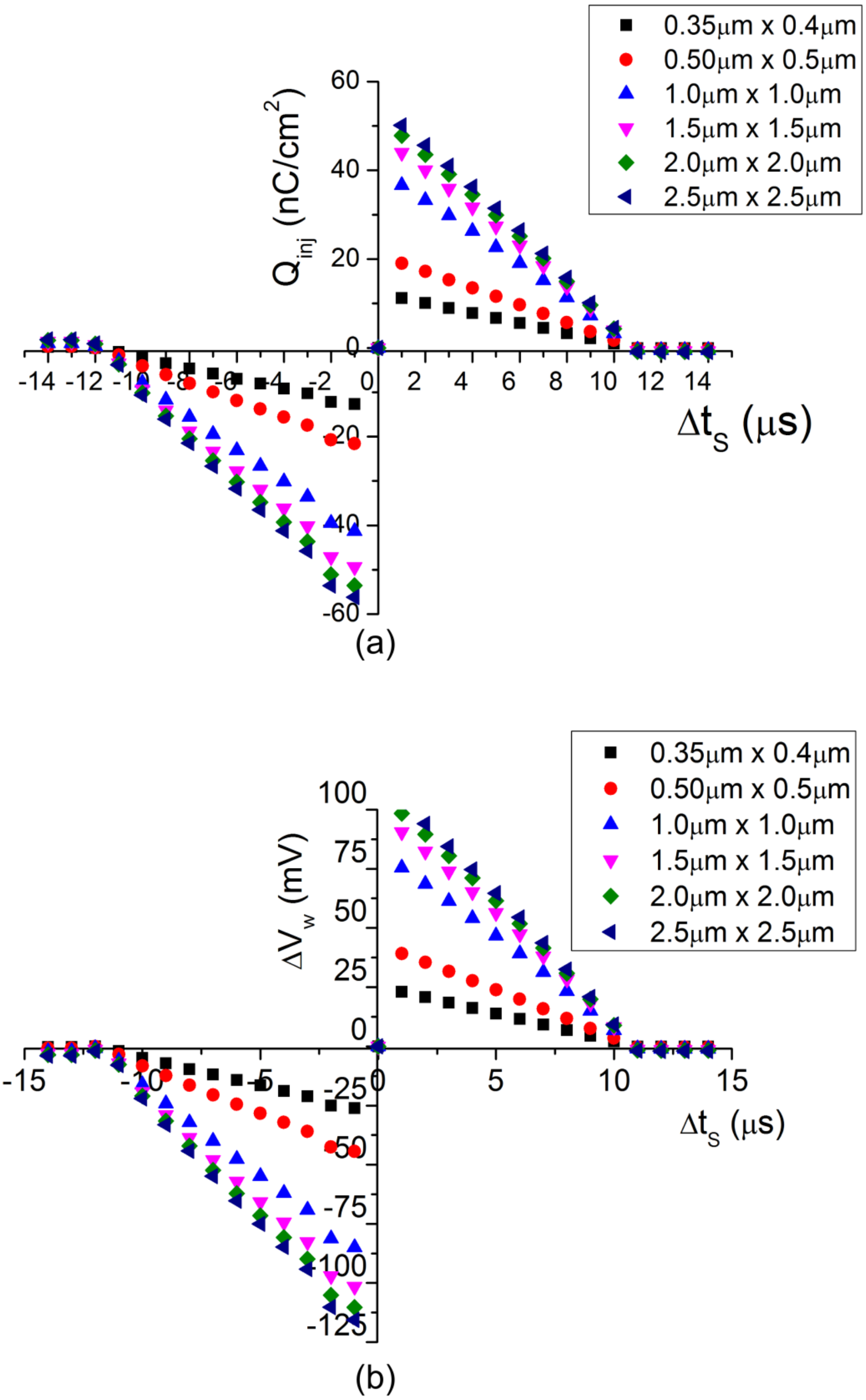
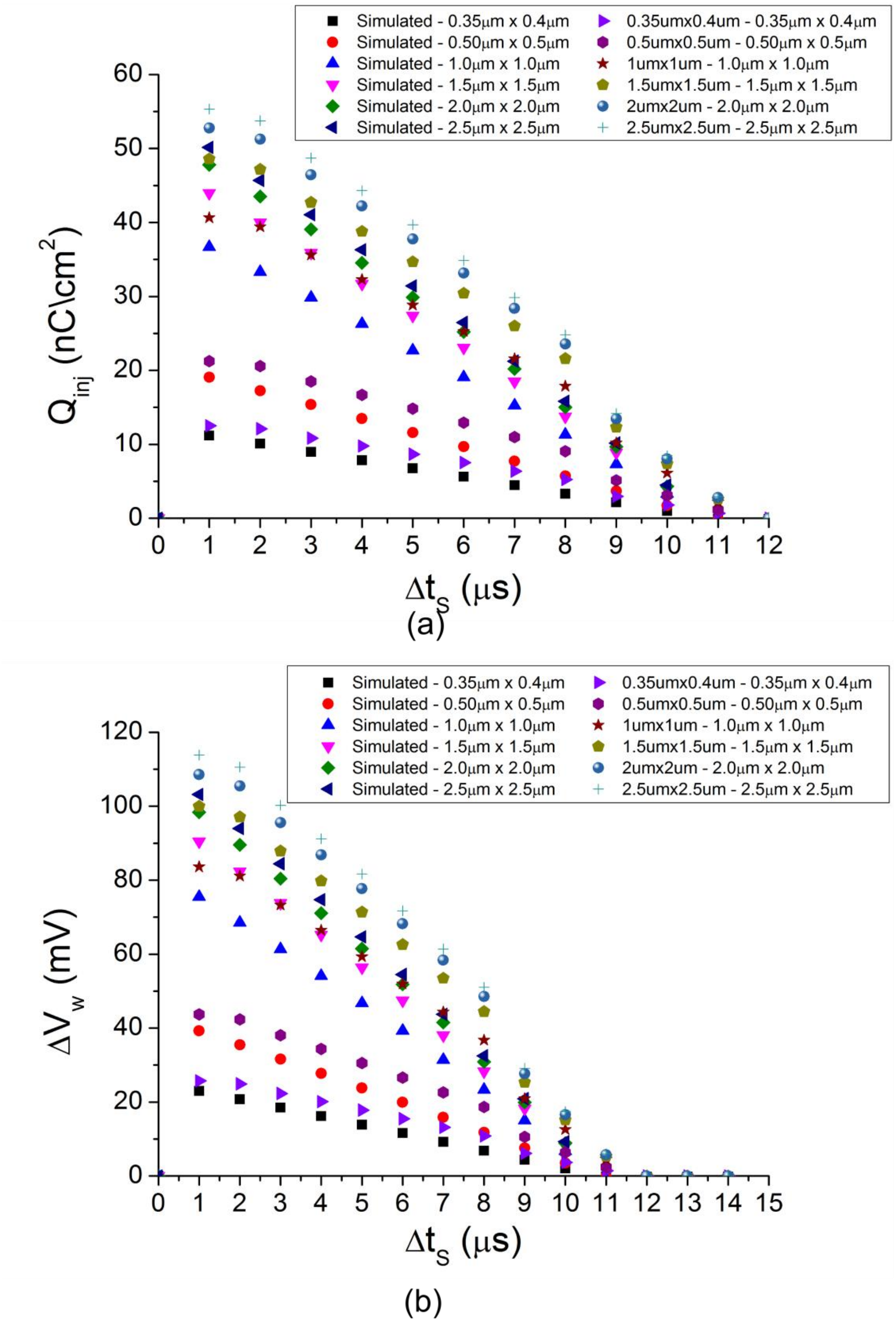


Fig 5.22 – STDP Curve – (a) Q_{inj} (Δw) (b) ΔV_w



(a)

(b)

Fig 5.23 – STDP Curve – (a) Q_{inj} (Δw) (b) ΔV_w – Showing WP simulated and measured results only.

5.3 Process Variation

5.3.1 Introduction

Semiconductor devices are prone to random statistical variations which can affect their performance and intended operation [31]. These random variations occur during the fabrication process and can be considered as two main types; local and global, Fig. 5.24. Local variations are a result of various factors including, but not limited to, doping distribution, junction depths, surface roughness, film thickness, edge definition [31].

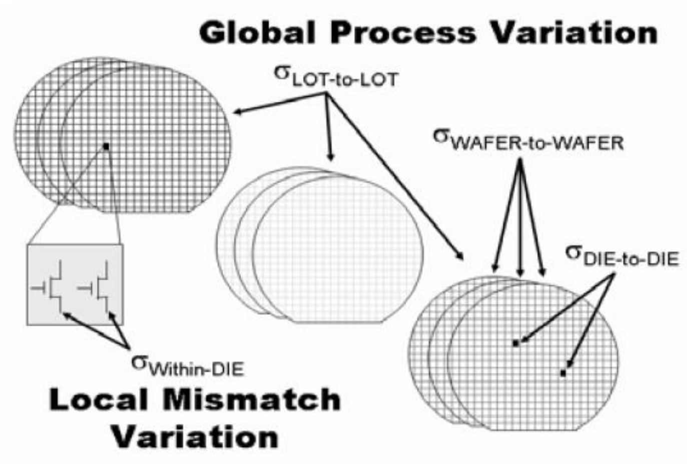


Fig 5.24 – Classification of Process Variations [31]

Local process variations can affect; flat band voltage, V_{fb}, channel length, L, channel width, W, oxide thickness, t_{ox}, oxide capacitance, C_o, substrate doping, N_a, device mobility, μ, current factor, β, and threshold voltage, V_t [31-43]. Subthreshold MOSFETs are particularly sensitive to process variation due to the exponential relationship between drain current, I_D and V_t. V_t is strongly related to several device parameters which are prone to variation during fabrication, equation 5.21. Process variation can affect most parameters of the MOSFET and these can conveniently be represented by the current factor (β) and threshold voltage, V_t [32-43]. Generally for subthreshold devices it is only V_t which is considered as this can incorporate variations in both off-current and subthreshold slope [35, 38, 43-45] as shown in equation 5.21.

$$V_t = t_{ox}\frac{\varepsilon_s}{\varepsilon_{ox}}\sqrt{\frac{2qN_a(2\phi_F)}{\varepsilon_0\varepsilon_s}} + 2\phi_F + \Phi_{MS} + \frac{Q_t}{C_o} \qquad (5.21)$$

Where N_a – acceptor doping concentration, t_{ox} – oxide thickness, ϕ_F – Fermi potential, Φ_{MS} – work function difference, Q_t – trapped oxide charge density, C_o – oxide capacitance, and ε_0, ε_s, ε_{ox} are the permittivity of free space, relative permittivity of silicon and silicon dioxide respectively.

The important effects of process variation on the proposed STDP circuit are now considered with particular reference to the critical timing window. Process variation may cause the circuit to function outside of its proposed, ‘ideal’, operating characteristics. Specifically the critical timing window may deviate from the proposed biologically plausible value. This will affect the synaptic weight update for a given pre-post spiking event.

The approximate effects of process variation on t_{CW} will be assessed by considering a variation in the V_t of M_{leak} Any change in N_a, t_{ox}, ϕ_F, Φ_{MS} and Q_t due process variation will cause V_t to vary, equation 5.21. Equation 5.22 will be used to represent the V_t of M_{leak} in equation 5.07.

$$V_t = V_{t0} \pm \Delta V_t \tag{5.22}$$

Where V_{t0} is the nominal, ideal, threshold voltage for the AMS process, $V_{t0} = 0.48$. ΔV_t will represent the change in V_{t0} due to process variation. For the AMS process $\Delta V_t \approx \pm 20$mV. By substituting equations 5.07, 5.20 into 5.11 it is possible to model the effects of process variation on t_{cw}.

$$t_{cw} = \frac{0.8 V_M C}{I_0 exp\left[\frac{q}{mkT}(V_{leak} - [V_{t0} \pm \Delta V_t])\right]} \tag{5.23}$$

An alternative approach to modelling the effect of process variations is to use a Monte Carlo analysis within a TCAD package, such as Cadence. Monte Carlo analysis involves a set of simulations, typically 100+, where in each simulation the critical model parameters (such as

N_A, t_{ox}, C_{ox}, W, L, etc) are set to random values, (within a deviation of ±3σ from the ideal) not necessarily worst case. This statistical variation is akin to that which can occur due to process variation. It should be noted that Monte Carlo analysis can be used to model lot-to-lot, wafer-to-wafer, die-to-die (including device mismatch) either individually or as a combination.

5.3.2 Simulation Results and Discussion

Monte Carlo analysis was undertaken using Cadence to assess the effects of inter-die (die-to-die) process variation on the critical timing window and results are presented in Fig. 5.25. The results indicate that a range of possible values for t_{CW} are possible. Comparing the results to the model results presented as in Fig. 5.26, it is clear to see that the same range of values for t_{cw} are possible. This indicates that the effects of process variation can be approximately modelled by considering the variation in V_t only, with $\Delta V_t = \pm 20$mV (from AMS specifications). This is further established by considering the equation for V_t, equation 5.21, which shows that V_t is dependent upon N_a, t_{ox}, ϕ_F, Φ_{MS} and Q_t (trapped, mobile, interface and fixed charge), which are all affected by process variations.

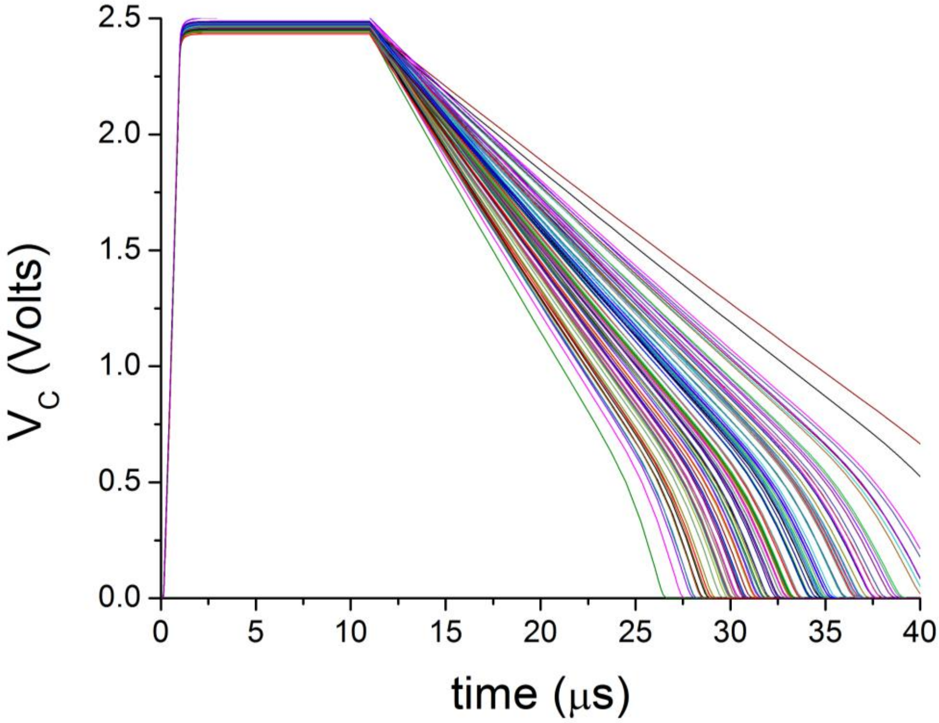


Fig 5.25 – Die-to-Die Process Variation, V_{leak} = 400mV

Fig. 5.26 presents the maximum and minimum t_{cw} due to the effects of process variation (from Fig. 5.26). In addition to this the ideal t_{cw} and maximum and minimum values generated by equation 5.23 are presented. The results indicate that equation 5.21 can be used to approximate the effects process variation, with $\Delta V_t = \pm 17.5$mV, (±2.5mV difference compared to the worst case AMS values). The change in the critical timing window, t_{cw}, from the ideal value of 20µs is dependent upon ΔV_t. For $\Delta V_t = +17.5$mV, $t_{cw} = 30.86$µs, and for $\Delta V_t = -17.5$mV, $t_{cw} = 12.21$µs.

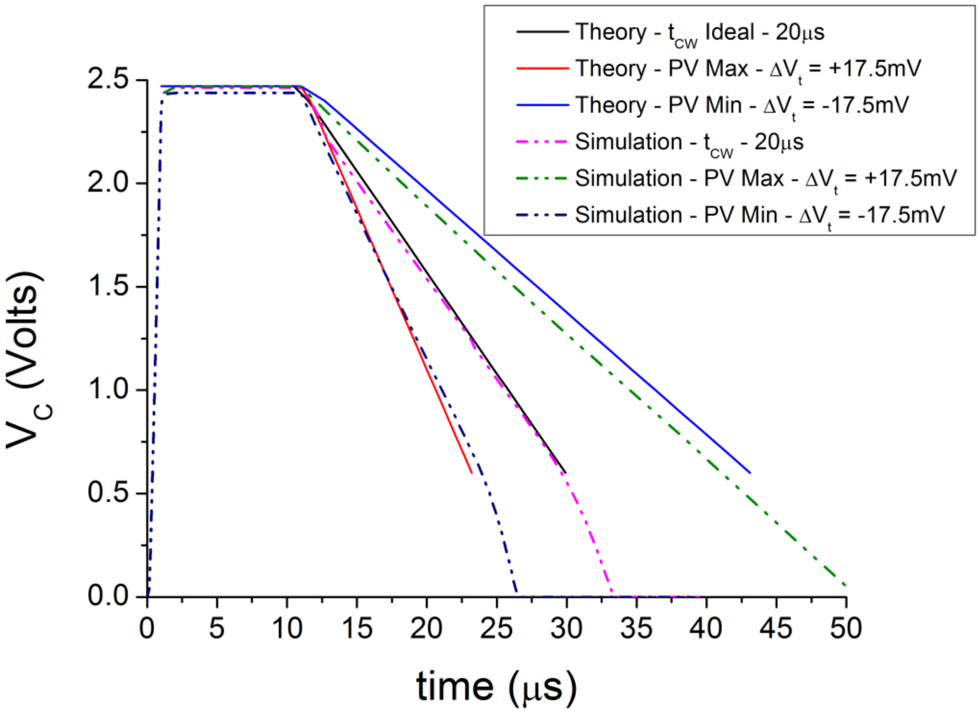


Fig 5.26 – Die-to-Die Process Variation - t_{CW} variations (max, min and ideal) and max, min values from equation 5.14 model. V_{leak} = 410mV.

The effects of process variation will be to cause a shift the ideal STDP plot, Fig. 5.27. PV will vary the amount of charge (hence potential of charge) injected/removed from the FG, Fig. 5.28. For $t_{cw} < 20$µs, a positive ΔV_t, $Q_{inj}(\Delta w)$ (and ΔV_w) curve is shifted to the left. Conversely if $t_{cw} > 20$µs, a negative ΔV_t, $Q_{inj}(\Delta w)$ (and ΔV_w) curve is shifted to the right. Specifically there is no overall change in the magnitude of Δw, Q_{inj}. Rather there is a shift in the magnitude of the charge injected/removed for the same Δt_s value. This does not affect the overall operation of the STDP circuit in that it still follows the STDP rule.

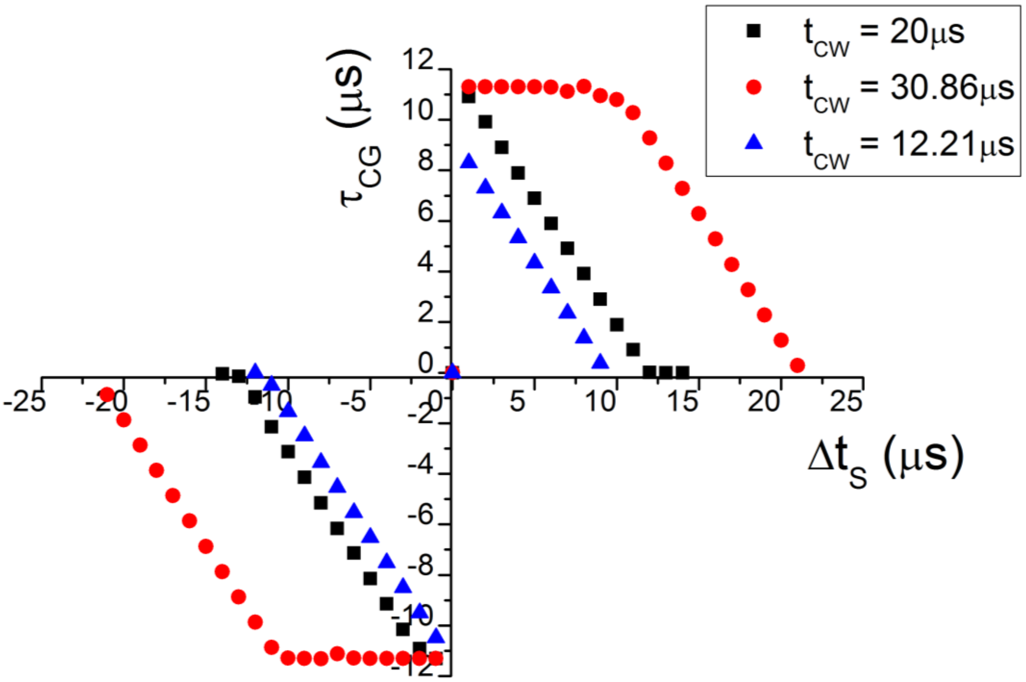
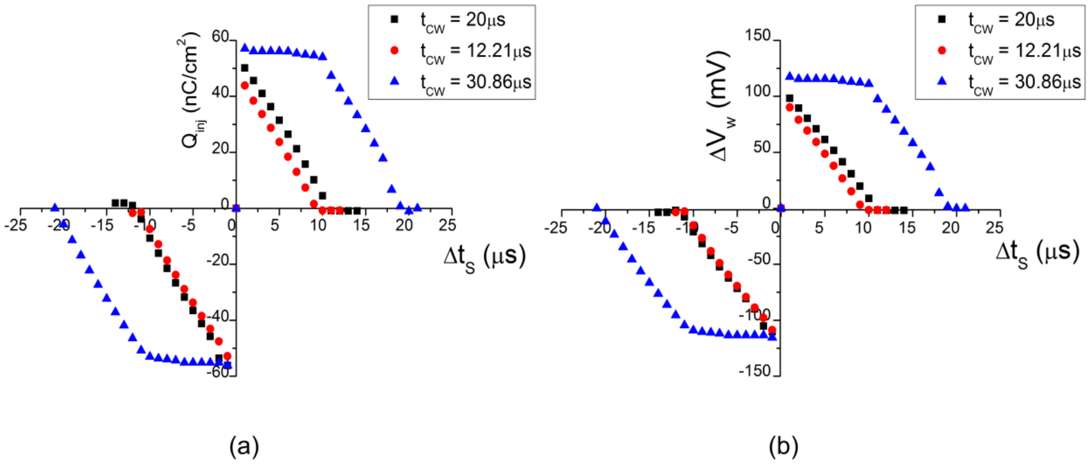


Fig 5.27 – STDP curves showing effect of Process Variation (max, min and ideal)



(a)

(b)

Fig 5.28 – STDP curves showing effect of Process Variation (max, min and ideal) on (a) Q_{inj} (Δw) and (b) ΔV_w against Δt_s

Therefore effect of process variation essentially means that if a synaptic weight is to be equal to say +1V; then if PV causes $t_{cw} < t_{cwideal}$, more pre-post spiking events are required. Similarly if $t_{cw} > t_{cwideal}$, less pre-post spiking events are required when compared to the ideal STDP curve. The same is true for a synaptic weight of -1V. The requirement of more or less spiking events can be compensated for within the learning rule/training of the neural network, such that effects of process variation are reduced.

Since the overall effect of process variation can be modelled simply by a change in V_t, then by rearranging equation 5.23 in terms of V_{leak} gives 5.24 the required value of V_{leak} such that $t_{cw} = t_{cwideal} = 20\mu s$, can be found. This is based on the assumption that $\Delta V_t = \pm 17.5mV$. For $\Delta V_t = 17.5mV$, $V_{leak} = 423mV$ and for $\Delta V_t = -17.5mV$, $V_{leak} = 380mV$.

$$V_{leak} = m\frac{kT}{q}ln\left[\frac{0.8V_M C}{I_0 t_{CW}}\right] + (V_{t0} \pm \Delta V_t) \tag{5.24}$$

The results show the importance of using V_{leak} to ‘tune’ t_{cw} to achieve the desired biologically plausible critical timing window.

5.4 Conclusion

A compact STDP circuit which performs synaptic weight increase and decrease has been presented. The circuit is used to update the synaptic weights within a hardware neural network. The circuit has advantages over current implementations in that it can implement a critical timing window for synaptic modification. The duration of the critical timing window is set by the subthreshold current controlled by the voltage applied, V_{leak}, to transistor M_{leak} in the circuit. Simulation and experimental results of the WP are presented which indicate that for a post-pre spiking event, no update of the synaptic weight occurs. A pre-post spiking event will cause the synaptic weight, which is represented as charge on a FG in a NVM device, to be increased. The amount, by which the synaptic weight is changed, Δw, is determined by the duration that V_{wi} is greater than 1.2V and also by the magnitude of V_{CG}. The maximum weight, Δw_{max} is obtained when V_{CG} has a pulse width of 10µs and a constant magnitude of 10V. The minimum weight Δw_{min}, prior to V_{wi} being less than 1.2V is achieved when V_{CG} has a pulse width of 1µs and magnitude of 9.6V. Similarly simulation results for the WD circuit block indicate that for a pre-post spiking event, no update of the synaptic weight occurs. A post-pre spiking event will cause the synaptic weight, which is represented as charge on a FG in a NVM device, to be decreased. Both the WP and WD circuits have a power consumption of approximately 2.4mW, during a weight update. If no weight update occurs the resting currents within the device are in the nA range, thus each circuit has a power consumption of approximately 1µW.

Since the critical timing window is determined by subthreshold transistor M_{leak}, the effects of process variation must be considered. It has been shown that process variation can be modelled simplistically as a variation in the nominal, ideal, threshold voltage, V_t. This has been verified through the comparison of simulation results for the simple model with Monte Carlo analysis simulations undertaken using Cadence. In addition to this both wafer-to-wafer and die-to-die process variations have a dramatic effect upon the critical timing window.

However by choosing a suitable value of V_{leak} it is possible to tune t_{CW} to ensure a biologically plausible timing window despite the effects of process variation which occur during the fabrication process. Additionally the amount of charge which can be injected on the FG and the controllability of the injection of this charge has also been presented. A physical model has been presented which allows design and analysis of the blocks.

5.5 References

[1] G. Indiveri, E. Chicca and R. Douglas, "A VLSI array of low-power spiking neurons and bistable synapses with spike-timing dependent plasticity," *IEEE. Trans. Neural Networks*, vol. 17, no. 1, pp. 211-221, 2006.

[2] C. Diorio, P. Hasler, B. A. Minch and C. A. Mead, "A single transistor silicon synapse", *IEEE Trans. Electron Devices*, vol. 43, no. 11, pp. 1972-1980, 1996.

[3] D. H. Goldberg, G. Cauwenberghs and A. G. Andreou, "Probabilistic synaptic weighting in a reconfigurable network of VLSI integrate-and-fire neurons", *Neural Networks*, vol. 14, pp. 781-793, 2001.

[4] L. F. Abbott and S. B. Nelson, "Synaptic plasticity: taming the beast", *Nature Neuroscience supplement*, vol. 3, pp. 1178-1183, 2000.

[5] D.O. Hebb. *The Organisation of Behaviour.* Wiley 1949.

[6] W.B. Levy and O. Steward, "Temporal contiguity requirements for long-term associative potentiation/depression in the hippocampus," *Neuroscience,* vol. 8, no. 4, pp. 791-797, 1983.

[7] G.Q. Bi and M.M Poo, "Synaptic modification in cultured hippocampal neurons: Dependence on spike timing, synaptic strength and postsynaptic cell type," *J. Neuroscience*, vol. 18, pp. 10462-10472, 1993.

[8] M.Nishiyama, K. Hong, K. Mikoshiba, M.M. Poo and K. Kato, " Calcium stores regulate the polarity and input specificity of synaptic modification," *Nature,* vol. 408, pp. 584-588, 2000.

[9] M. Tsukada, T. Aihara, Y. Kobayashi and H. Shimazaki, "Spatial analysis of spike-timing-dependent ltp and ltd in the ca1 area of hippocampal slices using optical imaging," *Hippocampus*, vol. 15, no. 1, pp. 104-109, 2005.

[10] H. Tanaka, T. Morie, and K. Aihara, "A CMOS spiking neural network with symmetric/asymmetric STDP function," *IEICE Transactions on Fundamentals,* vol. E92-A, no. 7, pp. 1690-1698, 2009.

[11] G.Q. Bi and M.M Poo, "Synaptic modification of correlated activity: Hebbs postulate revisited," *Annu. Rev. Neurosci*, vol. 24, pp. 139-166, 2001

[12] N. Caporale and Y. Dan, "Spike timing-dependent plasticity: A Hebbian learning rule," *Annu. Rev. Neurosci*, vol. 31, pp. 25-46, 2008

[13] I. B. Levitand and L. K. Kaczmarek, *The Neuron – Cell and Molecular Biology,* 3rd Edition, Oxford University Press, 2002

[14] D. Purves, G. J. Augustine, D. Fitzpatrick,. L. C. Katz, A. LaMantina, J. O. McNamara and S. M. Willians, *Neuroscience*, 2nd Edition, Sinauer Associates Inc, 2001.

[15] N. Rebola, B. N. Srikumar and C. Mulle, "Activity-dependent synaptic plasticity of NDMA receptors", *J. Physiology,* vol. 588, no. 1, pp. 93-99, 2010.

[16] S. Song, K. D. Miller and L. F. Abbott, "Competitive Hebbian learning through spike-timing-dependent synaptic plasticity", *Nature Neuroscience*, vol. 3 no. 9, pp. 919-926, 2000.

[17] N. Carporal and Y. Dan, "Spike timing-dependent plasticity: A Hebbian learning rule", *Annu. Rev. Neurosci*, vol. 31, pp. 25-46, 2008

[18] P. J. Dew and L. F. Abbott, "Extending the effects of spike-timing-dependent plasticity to behavioural timescales", *PNAS*, vol. 103, no. 23, pp. 8876-8881, 2006.

[19] R. C. Froemke, D. Debanne and G. Q. Bi, "Temporal modulation of spike-timing-dependent plasticity", *Frontiers in Synaptic Neuroscience,* vol. 2, no. 1, pp. 1-16, 2010.

[20] K. A. Buchanan, and J. R. Mellor, "The activity requirements for spike-timing-dependent plasticity in the hippocampus", *Frontiers in Synaptic Neuroscience*, vol. 2, no. 11, pp. 1-5, 2010.

[21] Z. F. Mainen and T. J. Sejnowski, "Reliability of spike timing in neocortical neurons", *Science*, vol. 268, pp. 1503-1506, 1995.

[22] S J. Schemmel, K. Meier and E. Mueller, " A new VLSI model of neural microcircuits including spike timing dependent plasticity," *IEEE International Joint Conference on Neural Networks 2004*, vol. 3, pp. 1711-1716, 2004.

[23] J. Schemmel, K. Meier and E. Mueller, " Implementing synaptic plasticity in a VLSI spiking neural network model," *IEEE International Joint Conference on Neural Networks 2006*, pp. 1-6, 2006.

[24] K. Cameron, V. Boonsobhak, A. Murray and D. Renshaw, "Spike timing dependent plasticity (STDP) can ameliorate process variations in neuromorphic VLSI," *IEEE Transactions on Neural Networks*, vol. 16, no. 6, pp. 1626-1637, 2005

[25] Y. Hayashi, K. Saeki, and Y. Sekine, "A synaptic circuit of a pulse-type hardware neuron model with STDP," *International Congress Series*, vol. 1301, pp. 132-135, 2007.

[26] K. Saeki, R. Shimizu and Y. Sekine, "Pulse-type hardware neural network with two time window STDP," *ICONIP 2008, Lecture Notes In Computer Science*, vol. 5507/2009, pp. 877-884, 2009.

[27] M. M. Khan, D. R. Lester, L. A. Plana, A. Rast, X. Jin, E. Painkras and S. B. Furber, "SpiNNaker: Mapping neural networks onto a massively-parallel chip multiprocessor," *International Joint Conference on Neural Networks 2008*, pp.2850-2857, 2008.

[28] X. Jin, M. Lujan, L. A. Plana, S. Davies, S. Temple and S. B. Furber, "Modelling spiking neural networks on SpiNNaker," *Computing In Science and Engineering*, vol. 12, no. 5, pp. 91-97, 2010.

[29] X. Jin, A. Rast, G. Galluppi, S. Davies, and S. B. Furber, "Implementing spike-timing-dependent plasticity on SpiNNaker neuromorphic hardware," *World Congress on Computational Intelligence 2010*, pp. 2302-2309, 2010 Markram, H, "The blue brain project," *Nat Rev Neurosci. vol. 7, pp. 153-160*, 2006.

[30] S. Druckmann, Y. Banitt, A. Gidon, F. Schürmann, H. Markram, and I. Segev, "A Novel Multiple Objective Optimization Framework for Constraining Conductance-Based Neuron Models by Experimental Data," *Frontiers in Neuroscience, vol. 1, no. 1, 2007*

[31] J. Kozloski, K. Sfyrakis, S. Hill, F. Schürmann, C. Peck and H. Markram, "Identifying, tabulating, and analyzing contacts between branched neuron

morphologies," *IBM Journal of Research and Development, Vol 52, Number 1/2, 2008*

[32] D. C. Potts, "Statistical Analogue Circuit Simulation: Motivation and Implementation", *Advances in Analogue Circuits*, InTech, 2011.

[33] Y. Cheng, "The influence and modelling of process variation and device mismatch for a*nalogue/RF circuit design", Proceedings of the 4th IEEE International Caracas Conference on Devices, Circuits and Systems 2002*M

[34] .J.M. Pelgrom, A.C.J. Dunima*iker and A.P.G.* Welbers*, "*Matching Properties of MOS Transistors*", IEEE Journal of Solid State Circuits, vol. 24, no. 5, pp. 1433-1440, 1989.*

[35] M.J.M. Pelgrom, H.P. Tuinhout and M. Vertregt, "Transistor matching in analogue CMOS applications", *IEDM*, pp. 915-918, 1998.

[36] M.T. Terrovitis and C.J. Spanos, "Process Variability And Device Mismatch", *First International Workshop on Statistical Metrology*, 1996

[37] P.G. Drennan and C.C McAndrew, "Understanding MOSFET mismatch for analogue design*", IEEE Journal of Solid State Circuits*, vol. 38, no. 3, pp. 450-456, 2003.

[38] P.R. Kinget, "Device mismatch: An analogue design perspective", *ISCAS 2007*, pp. 1245-1248, 2007.

[39] R. Jaramillo-Ramirez, J. Jaffari and M. Anis, "Variability aware design of subthreshold devices", *ISCAS 2008*, pp. 1196-1199, 2008.

[40] H. Kosina, M. Nedjalkov and S. Selberherr, "Theory of the Monte Carlo method for semiconductor device simulation", *IEEE Transactions on Electron Devices*, vol. 47, no. 10, pp. 1898-1908, 2000.

[41] H. . Hung and V. Adzic, "Monte Carlo simulation of device variation and mismatch in analogue integrated circuits", *NCUR 2006*, 2006.

[42] J. B. Shyu, G.C. Temes and F. Krummenacher, "Random error effects in matched MOS capacitors and current sources", *IEEE Journal of Solid State Circuits*, vol. sc-19, no. 6, pp. 948-955, 1984.

[43] J. B. Shyu, G.C. Temes and K. Yao, "Random error in MOS capacitors", *IEEE Journal of Solid State Circuits*, vol. sc-17, no. 6, pp. 1070-1076, 1982

[44] B. Zhai, S. Hanson, D. Blaauw and D. Sylvester, "Analysis and mitigation of variability in subthreshold design", *ISLPED 2005*, pp. 20-25, 2005

[45] S. N. Mozaffari and A. Afzali-Kusha, "Statistical model for subthreshold current considering process variation", *ASQED 2010*, pp. 356-360, 2010

[46] Z. A. Weinberg, "On tunnelling in metal-oxide-silicon structures", *Journal of Applied Physics*, vol. 53, no. 7, pp. 5052-5056, 1962.

[47] J. R. Brews, “Subthreshold behaviour of uniformly and non-uniformly doped long-channel MOSFETs”, *IEEE Transactions on Electron Devices,* vol. ED-26, no. 9, pp.1282-1291, 1979

[48] P. D. Roberts and C. C. Bell, “Spike timing dependent synaptic plasticity in biological systems”, *Biological Cybernetics*, vol. 87, no. 5-6, pp. 392-403, 2002.

[49] B. Lu, W.M. Yamada, and T. W. Berger, “Asymmetric Synaptic Plasticity Based on Arbitrary Pre- and Postsynaptic Timing Spikes Using Finite State Model”, *Proceedings of International Joint Conference on Neural Networks*, Orlando, Florida, USA, August 12-17, 2007

Chapter 6 – Limited Precision Weights

6.1 Introduction

In order for neural networks, NNs, to be a potential replacement to Von Neumann and other computing architectures, there is a need for them to be designed as effective and efficient systems. Software and hardware NNs therefore need to be designed such that they are compact and low powered. However the storage and updating of the synaptic weight becomes a major challenge when trying to establish a compact system.

Traditionally in software the synaptic weights are implemented using floating point arithmetic with single or double precision weights [1-2]. In HNN there are several methods which can be used to store synaptic weights. For digital HNN external RAM or ROM chips are often used, in analogue HNN custom local FG devices are one potential method used. In either case due to the large interconnection of synapses between neurons, the synaptic weight storage device (and interconnect) can account for a large proportion of the silicon for the network. Therefore there is a need to reduce the size of the storage device/method to gain a compact neural network as envisioned in chapter 1.

Considering the FG device presented in chapter 4, the synaptic weight of an associated synapse is stored, as charge, on a FG. The FG device is designed such that it can be integrated with a charge transfer synapse [14]. Chapter 4 also presented a model of the charge storage characteristics of the FG device, Fig. 6.1. Fig. 6.1(b) shows the total weight charge, Q_w, stored and (a) shows the corresponding potential of Q_w, V_w. The plots indicate that Q_w (and as such V_w) is dependent upon the tunnelling area within the FG device. Specifically a large tunnelling area allows for more charge to be stored over the same time period and vice versa. Figure 6.1(b) also indicates that Q_w will saturate causing little or no further charge to be stored on the FG with increasing time. The point, at which saturation occurs is dependent upon the tunnelling area for the FG device. Since Q_w represents the associated synaptic weight, the saturation of Q_w also imposes a limit on the number of possible weights which can be implemented.

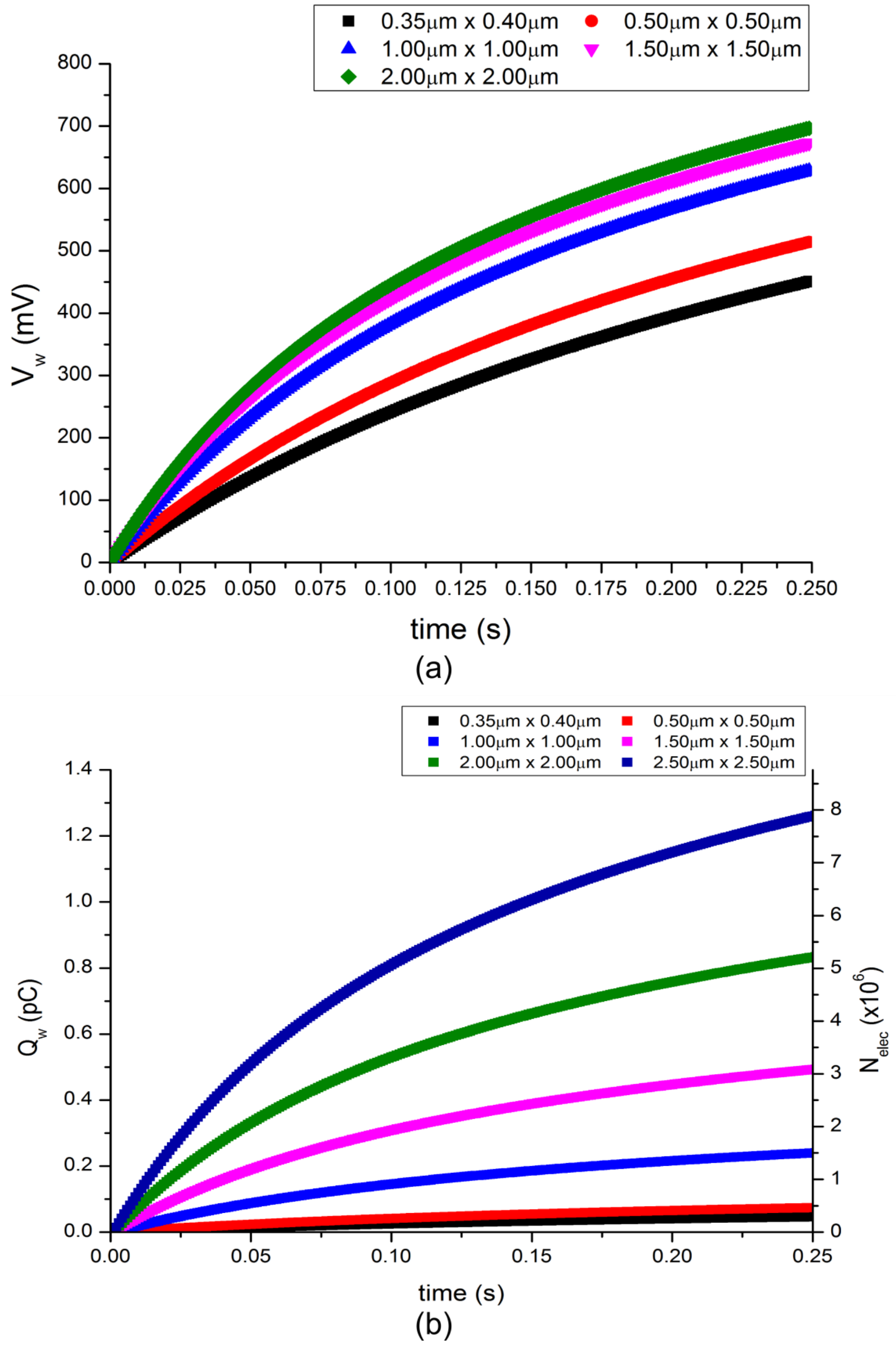


Fig 6.1 – FG device charge storage characteristics showing (a) potential of total charge stored on the FG and (b) total stored charge and number of electrons

Within an analogue HNN, synaptic weights can be represented in two main ways; firstly Q_w can directly map to the synaptic weight, w →Q_w. Use of this method allows for a large number of weights with a high precision. The second method is to use a packet or range of charge to represent each synaptic weight. This method can significantly reduce precision and number of possible weights implemented compared with the first method. In either case there is still a limit on the number of weights due to the saturation effect.

In the SpiNNaker system, synaptic weights are stored externally, off-chip and away from the synapses, using 1G Synchronous dynamic random access memory, SDRAM [15, 16]. Each synaptic weight is stored as a 16-bit weight value with an additional 4 bits for the synaptic delay value and 11 bits for the post-synaptic neuron index [15-17]. Each 32-bit synaptic weight value typically requires 2-4 bytes in memory, with each SpiNNaker core needing at least 10^6 words, 4MB, of storage [17, 18]. Assuming integer values are used, the synaptic weights can be one of 2^{16} potential values. This can be either from 0 – 65,536 or -32,768 to -32,767. Additionally SpiNNaker can also represent the synaptic weight more accurately as a 16 bit floating point number. This reduces the total possible number of weights but increases the resolution of each weight.

By contrast the FACETS and BrainScaleS projects store the synaptic weight using a 4 bit SRAM locally at each synapse [19-21]. Within the FACETS synapse the synaptic weight is converted from its 4 bit value to a current. Within the FACETS chip, the maximum conductance for a column of synapses is controlled by an analogue input g_{max}. The g_{max} values of two adjacent columns which share the same pre-synaptic input can be programmed to be a fixed multiple of each other. These columns can therefore combine their synapses to increase the weight resolution to between 6 to 8 bits. This causes a reduction in the synapse number within these columns [21]. Unlike the SpiNNaker project, the FACETs chip synaptic weights are discretised when using a 4 bit implementation. In [21] it has been shown that the FACETS chip can function as required with just a 4-bit resolution for each synaptic weight.

In software neural networks, a compact and efficient system can be achieved by reducing the number of bits required to represent and store the synaptic weight, while maintaining a suitable number of weights. This ensures that the network can still find a solution to the proposed problem. The reduction results in a network which has a uses a reduced weight range and limited precision weights [1-6]. Typically digital HNNs employ a high precision (and large number of weights) by utilizing a large amount of RAM/ROM to store the synaptic weights. However it is possible to implement digital HNNs with low/limited precision/number of weights, requiring significantly less memory to represent the synaptic weights [5- 6]. Therefore there exists a trade-off between the number of weights/weight precision and the size of the synaptic weight storage device. However if limited precision weights, akin to those utilized in software neural networks is used, then the weight precision and memory requirement is reduced while still allowing for the NN to perform its intended function.

As with digital HNN, analogue HNN, which use FGs, also have trade-offs between size of synaptic weight storage and the number and precision of weights. Therefore it is proposed that limited precision weights, currently used with training of digital HNN, can be expanded for use with analogue HNN. Limited precision weights can allow for compact neural networks by reducing the number and/or precision off synaptic weights while allowing the network to function as intended. In the following section an overview of limited precision weights (LPW) and current LPW training algorithms used in hardware neural networks is presented.

6.2 LPW Algorithms

Since the 1980s there has been a growing interest and research in neural networks which utilize low/ reduced precision synaptic weights. Two main avenues of research into this approach have been established. The first avenue focuses on the adaption and modification of traditional training algorithms such that they can be used with reduced precision weights, RPW [5, 7]. In [5, 7] it is shown that neural networks with a precision of 5-7 bits (representing each synaptic weight) can be trained to function as intended by simply using a continuous-discrete learning algorithm. An alternative algorithm which reduces the number of possible weights which can be used in each training session has also been shown to allow a neural network to function with RPW [8]. While there has been considerable focus on adapting traditional training algorithms for use with RPW, the second avenue of research was to consider the use of novel techniques and algorithms which use limited precision weights, LPW to train the neural network.

6.2.1 Probabilistic Rounding Algorithm

In [1, 9] probabilistic round algorithms were used resulting in minimal weight updates. In [1] a new training algorithm based upon the cascade correlation, CC, is outlined for use with neural networks. Specifically the new algorithm is adapted to use limited precision weights. The CC algorithm is an incremental supervised learning algorithm for use with feed-forward neural networks. The algorithm starts with a minimal neural network (input and output layers only) with additional hidden layers added during training. Each layer receives weighted connections from the previously added hidden layers. The CC algorithm then trains (and updates) the weights such that a maximum correlation measure exists with all hidden layer connections to the output layer are unconnected, as described in Equation. 6.1. The weights are then frozen and the network is retrained with the hidden layers connected to the output layer.

$$w(t + 1) = w(t) + \Delta w(t) \tag{6.1}$$

Additionally [1] has been shown that when the CC algorithm is used with limited weights and depending upon the problem, the CC can only derive a minimal network with the following weight ranges:

- For the sonar problem ±128
- For the spirals task ±64
- For the parity task ±8

Equation 6.1 is adapted such that the weights are restricted to [-1, 1), as shown in equation 6.2.

$$w(t + 1) = P(w(t) + \Delta w(t)) \tag{6.2}$$

To minimize the precision error, the weights are re-scaled during training. Results of an investigation into the effect of LPW on training using equation are described in [1]. Reduction of the weight precision has little effect on the learning of the network until a critical value of *n* is reached, where n is dependent upon the specific problem. For example for the two spirals problem, n = 12 whereas n = 10 for both the sonar and parity problems. In the latter two cases, n=10 occurs as the weight update, Δw, falls below the minimum possible weight update step size.

An improvement is to use probabilistic weight updates [1]. When the proposed weight update, Δw, is below the quantized level, a minimum step with probability, p, is taken as the weight update. The use of probabilistic weights has been shown to give better results when low weight precision is used [1, 9]. This is due to the fact the use of LPW has little effect until a critical point, n, is reached. After this training will fail due to $\Delta w \rightarrow 0$, with the probabilistic rounding fails at n = 6 compared to n=12-14 for the CC algorithm.

6.2.2 Adapted Differential Evolution Algorithm

Another approach to implementing neural networks with LPW is presented in [2]. In [3] it is proposed that feed-forward neural networks (FNN), can be implemented in hardware using integer weights such that they offer a reduced cost in implementation, reduction in memory for weight storage, and increased noise immunity due to the use of integer weight values rather than real numbers, [10, 11]. Again integer weights are restricted to a set weight range similar to [1, 9], in this case to $[-2^k+1, 2^k-1]$, where k can be 3, 4, 5. This approach reduces the required memory storage required for the synaptic weights, [10, 11]. In addition the inputs are also restricted to {-1,1} [3, 10, 11].

One potential algorithm for use with FNN is the adapted differential evolution (DE). It is based upon the DE algorithm but is adapted for use with the restricted weights [3, 10-12]. The adapted DE algorithm is used to train the network by firstly taking a specific number, NP, of N-dimensional integer weight vectors as the initial weight population. These are evolved over time by iteration, to generate the final synaptic weightings. NP remains fixed throughout training. The weight population is initially taken randomly using $[-2^k+1, 2^k-1]$, where k can be 3, 4, 5 following a uniform probability distribution. After each iteration or generation, new weight vectors are established from the combination of randomly chosen ones from the weight population. These are rounded up to the nearest integer value. Mutation is applied such that the weight vectors are forced into the range $[-2^k+1, 2^k-1]^N$ to form a so-called trail vector. The trail vector will only be used in the next generation if it reduces the error function, E, this process is known as selection. The mutation operator generates a new mutation vector v^i_{g+1} based on one of the following relations, 6.3-6.8, from each weight vector w^i_g, where i=1,…, NP, g represents the current generation.

$$v^i_{g+1} = w^{r1}_g + \mu(w^{r1}_g - w^{r2}_g) \tag{6.3}$$

$$v^i_{g+1} = w^{best}_g + \mu(w^{r1}_g - w^{r2}_g) \tag{6.4}$$

$$v^i_{g+1} = w^{r1}_g + \mu(w^{r2}_g - w^{r3}_g) \tag{6.5}$$

$$v^i_{g+1} = w^i_g + \mu(w^{best}_g – w^i_g) + \mu(w^{r1}_g - w^{r2}_g) \tag{6.6}$$

$$v^i_{g+1} = w^{best}_g + \mu(w^{r1}_g – w^{r2}_g) + \mu(w^{r3}_g - w^{r4}_g) \tag{6.7}$$

$$v^i_{g+1} = w^{r1}_g + \mu(w^{r2}_g – w^{r3}_g) + \mu(w^{r4}_g - w^{r5}_g) \tag{6.8}$$

Note w^{best}_g is the best member of the previous generation and $\mu > 0$ is a real parameter. Since v^i_g is a real number, it is rounded to the nearest integer value, as shown in Equation 6.9.

$$v^i_{g+1} = \text{sign}(v^i_{g+1}) \text{ x } (| v^i_{g+1} | \bmod 2^{k+1}) \tag{6.9}$$

The adapted DE algorithm has been applied to the XOR and 3-bit parity problems, with NP=2N [3]. This was to assess the performance of each DE algorithm, equations 6.3-6.8 with respect to the resulting error function and success rate. The success rate is based upon the number of simulations which succeed out of the 1000 undertaken [3]. The results presented indicate that for the XOR problem with N=18, an error function of E=0.00221 is achieved after training with a maximum success rate of 93.4% when 6.6 is used and k=3. Over 90% success rate is accomplished when k is increased to 4 for equations 6.5, 6.6. Similarly when k=5 using equations 6.5, 6.6 and 6.8. The success rates indicate that the adapted DE algorithm performs better than other continuous weight training algorithms such as the adaptive back propagation algorithm.

For the 3 bit parity problem, the network is to produce the sum, mod 2, of three binary inputs, a 97.7% success rate is achieved using equation 6.5 when k=3; 95.6% and 96.5%

success rates are also achieved for equations 6.5 and 6.6 when k is increased to 4. For k=5, over 90% is achieved using equations 6.4, 6.5 and 6.6. Since k represents the number of bits which are used to store the synaptic weights and hence the amount of storage space required, then the use of the DE algorithm introduces a trade-off between the memory storage size and the success rate of training; the effectiveness of the neural network. While the results indicate that it is possible to train a neural network with a limited number of bits, a specialized training algorithm which takes into account the use of limited precision integer weights, LPIW, must be used.

6.2.3 Quantize Back-propagation Step-by-Step Algorithm

The idea that neural networks which use LPW can be trained using specialized training algorithms is further echoed in [4-6]. In [4] it is proposed that a simple training algorithm, Quantize Back-propagation Step-by-Step, QBPSS, can be used with LPW in NNs. The QBPSS uses a look-up table based upon weight precision such that it does not introduce any new errors to the training of the network. In [4] it is proposed to train neural networks such that each neuron implements a hyperplane in an n_{q-1} dimensional space. For weights w;

$$w \epsilon \; [-r, r] \; (r = 1, 2, \ldots.) \qquad (6.10)$$

β is defined as the weight precision;

$$\beta = w_j - w_i, \; (\beta > 0, \; j\text{-}i=1) \qquad (6.11)$$

The weights can be expressed as

$$w \; \epsilon \; \{\lambda\beta \,\big|\, \lambda = 0, 1, \ldots, [r/\beta]\} \qquad (6.12)$$

Thus if $\beta = 0.1$, and $w \; \epsilon \; [-1, 1]$, then the possible weights which can be used are from the following range

$$w \in \{-1, -0,9, \ldots , 0, \ldots, 0.9, 1\} \quad (6.13)$$

If β is 1, then the hyperplanes which can be drawn as a mesh as shown in Fig. 6.2 [4]. Figure 6.2 indicates that the learning capabilities of the chosen neural network decreases as the weight range is reduced.

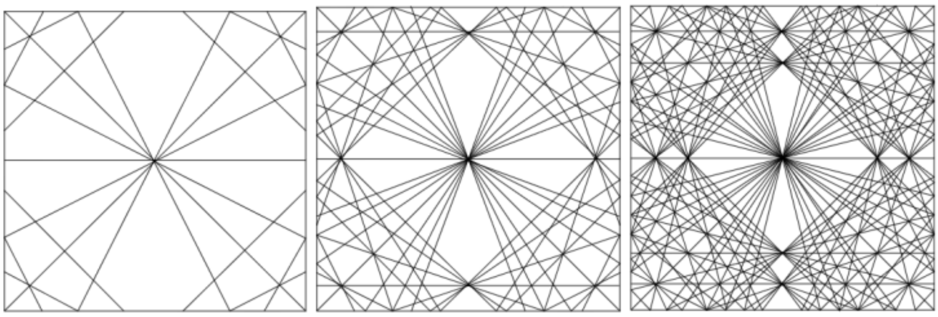


Fig 6.2 – Hyper planes which can be implemented when β=1, w∊ [-2, 2], [-3, 3], and [-4, 4] respectively. All planes are drawn on a [-0.4, 0.4] square [4].

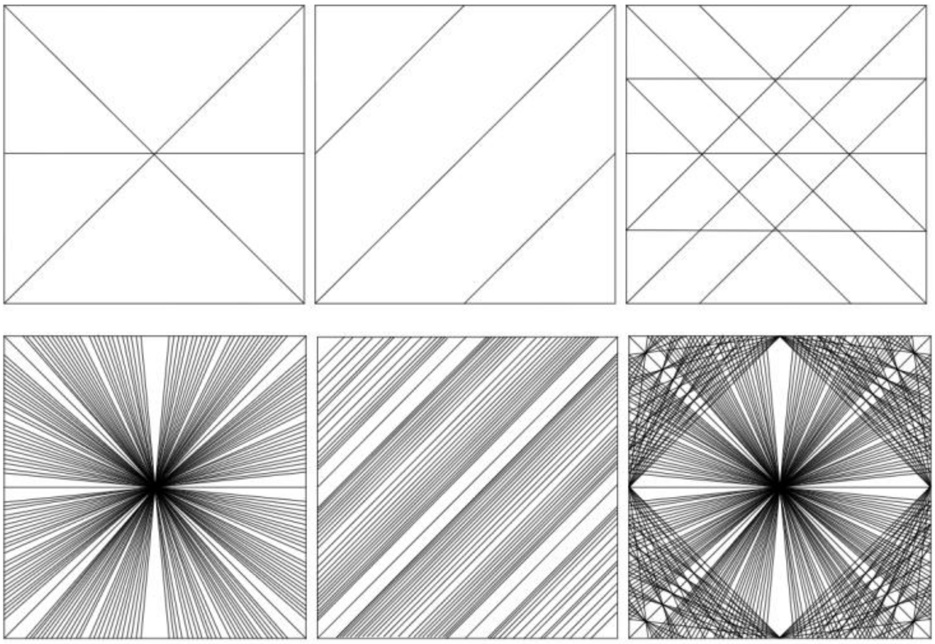


Fig 6.3 – Hyper planes which can be implemented when β=1 (top row), and β=0.1 (bottom row) with w∊ [-1, 1]. In the first column the hyper planes are drawn in the [-0.3, 0.3] square, in the second and third column the hyper planes are drawn in [-2, 2] square and [-0.1, 0.1] square respectively [4].

In Fig. 6.3 the weight precision, β, is increased from 0.1 (bottom row) to 1, (top row) with w$\in$ [-1, 1], and the possible hyper-planes are plotted. When β is decreased, the number of hyper-planes which can be implemented by the network is increased. Therefore it can be shown that the precision of the weights also affects the effectiveness of the neural network. A trade-off between the relative sizes of w and β is present. A NN with a large w range and low β is more capable than a neural network which has a small w range and large β. Conventional training algorithms have trouble training a network with low w and β, therefore [4] propose using QBPSS to overcome this problem.

The number of hidden layers can play an important role in the training and functionality of the network [4]. The more hidden layers that are present, the greater the chance that the training algorithm will converge to a solution. However large networks will take longer to train [13]. For neural networks which have a limited precision, a large learning rate is required. It has been shown that with a learning rate $\alpha = 0.25$ and $\beta = 0.1$ it is possible to train a NN [4, 13].

QBPSS is based upon the conventional BP algorithm. The BP has been shown to fail to converge when $\beta = 0.1$ where as LPW have been used successfully. In BP, if the weight increment is equal to or greater than 0.1, then the weights can be updated. However if this is not the case, then the weights will be quantized to 0. This can lead to weights being trapped in local minima. The QBPSS algorithm with $\beta = 0.1$ and an allowed error function of $E_{allowed} = 0.05$ can reduce this effect. In QBPSS, the synaptic weights are then updated using equation 6.14, where η is the momentum which allows the network to learn at a faster rate when plateau surface errors are present. If η is in the range, $0 \leq \eta < 1$ then $\Delta W^q(k-1)$ is a negative weight vector.

$$W^q(k+1) = \text{round}_\beta(W^q(k)+\alpha\Delta W^q(k)+\eta\Delta W^q(k-1)) \qquad (6.14)$$

The network is initially trained with $\beta < 0.1$, (high precision) until a pre-determined error or iteration number is reached, after which the weights are quantized to a lower β. The quantization process is repeated until the desired β is reached. The advantage of using QBPSS is that the desired response to the input stimuli is similar to the MTM (multithreshold method) training algorithm but with a higher accuracy. By using QBPSS it has been shown that the number of weights, and bits required to store the weights can be reduced while still allowing the network to perform its intended function.

6.2.4 VLSI Friendly LPW Algorithm

The work of [5-7] describes the use of an LPW algorithm with integer weight values for use with VLSI NN. If the range of the LPWs is not chosen correctly, then a neural network which uses hyperplanes and LPW will not be able to find a solution to the problem [5-7]. Additionally it is possible to calculate the integer weight range, [-p, p], such that a solution exists as a function of the minimum distance between the patterns of opposite classes, √n/2p. The effect of reducing the weight precision on the capabilities of the network is described in [5-7]. Using real number weights, hyperplanes can be implemented in any position within the problem space square by the training algorithm. In this case the only difficulty during training is whether the algorithm will converge. Provided the training algorithm is chosen correctly then a solution will exist such that all patterns are separated. If the weight precision is finite but sufficiently high, then the hyper-planes are sufficiently dense to allow convergence. If the weights are restricted to say integer values, then the number and positions of the hyper-planes is reduced and the chance of convergence is also reduced. In [5-7] it is stated than an axiom occurs;

"if a hyperplane cannot be implemented which will separate two patterns, then no matter what training algorithm is used, training will not converge to a solution".

Additionally if the problem is considered in a 2-D space, then the set of hyper-planes which are available can be viewed as a mesh within the problem space similar to that proposed in Fig 6.4 [4, 5].

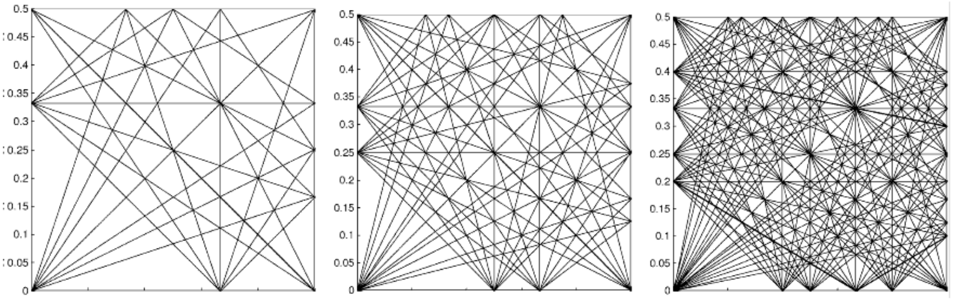


Fig 6.4 – Hyper planes which can be implemented with, β=1, and integer weights ranges [-3, 3], [-4, 4] and [-5, 5] respectively. All planes are drawn on a [0, 0.5] square [5].

To prove that it is possible to calculate the integer weight range, [-p, p], such that a solution exists as a function of the minimum distance between the patterns of opposite classes, $\sqrt{n}/2p$, [5-7] makes two propositions;

Proposition 1 – A strictly layered neural network using integer weights in the range [-p, p] can classify correctly any set of patterns included in a hypercube of side length 2l, l<0.5 centred around the origin of R^n, $n \geq 2$, for which the minimum Euclidean distance between the two patterns of opposite classes is $d_{min} \geq l_o = \max \{l, \sqrt{n}/2p\}$.

Proposition 2 –Consider a set of m patterns (in general positions) from two classes in the hypercube of side l centred in the origin of R^n. Let d and D be the minimum and maximum distance between patterns of opposite classes, respectively. The number of information entropy bits necessary in the worst-case for the correct classification of the patterns using weights in the set {-p, -p+1, … , 0, … p} where $p = [\sqrt{n}/2p]$ is lower bounded by $N_H = [m \bullet n \bullet \log(4pD/\sqrt{n})]$.

The lower bound of the number of weights which is required for the network is given by;

$$w > N_H/\log(2p+1) = N_H/\log(\sqrt{n}/d + 1) \qquad (6.15)$$

The number of entropy bits is not the same as the number of bits required to store the synaptic weight. Equation 6.15 is improved by taking in to account the average number of bits, k, required to store 1 bit of entropy information, as described in Equation 6.16. Equation 6.17 gives the minimum number of bits to store a binary decision in a generalized worst-case. Since equation 6.17 represents the worst-case implementation efficiency, equation 6 can be rewritten as equation 6.18.

$$w > kN_H/\log(2p+1) = kN_H/\log(\sqrt{n}/d + 1) \tag{6.16}$$

$$k = (n + 1)\log(2p + 1) \tag{6.17}$$

$$w > [(n + 1)\log(2p + 1)N_H]/\log(2p+1) = (n+1)N_H \tag{6.18}$$

Finally if a statistical view is taken with regards to w, and assuming a Gaussian distribution as the distance between patterns, then equation 6.18 becomes equation 6.19, assuming that majority of patterns from opposite classes are separated by $d=đ-1.5\sigma$. $đ$ is the average minimum distance between classes and σ is the standard deviation.

$$w \approx [(n + 1)m \bullet n\{1+\log(D) - \log(đ-1.5\sigma)\}] \tag{6.19}$$

Results for the XOR and 2-spiral problems are presented in [5] and shown in Fig. 6.5 and Fig. 6.6 respectively. From equation 6.19 the theoretical minimum value of the weight range p for XOR and 2-spiral problems are 1 and 6.96 respectively. The experimental results for the same problems indicate that in the 2-spiral problem, p = 5 can be used to successfully train the network. The equations presented in [5] thus give a worst-case scenario for the weight range required to solve the problem.

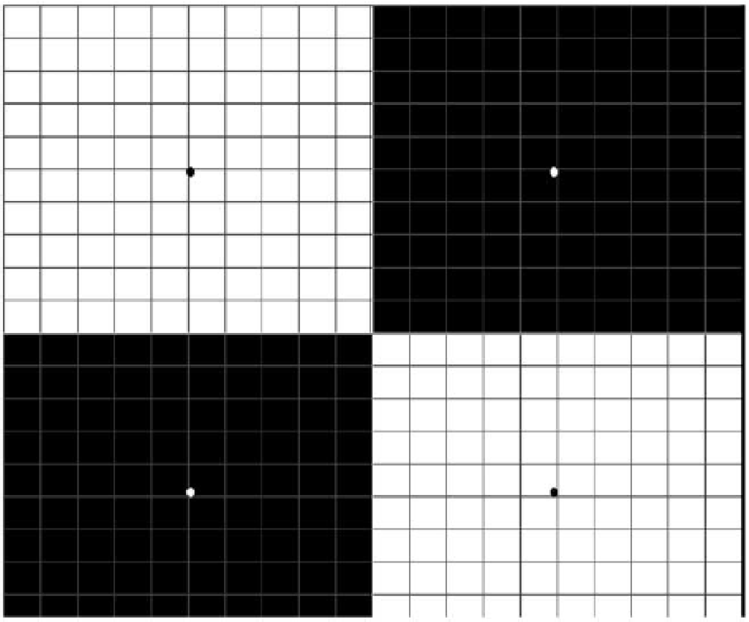


Fig 6.5 – XOR problem space results with integer weights ranges [-1, 1] [5]

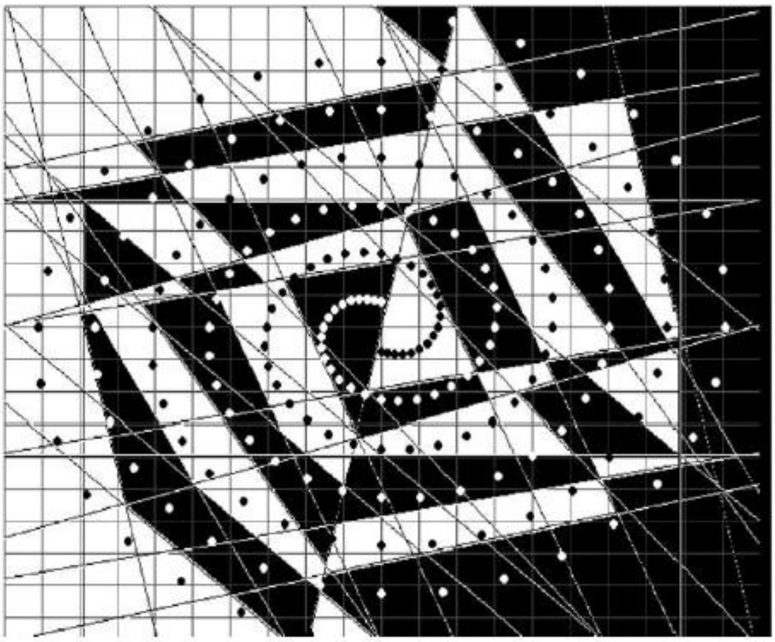


Fig 6.6 – 2-spirals problem space results with integer weights ranges [-5, 5] [5]

The theoretical minimum number of weights required for each problem are; 18 for the XOR problem and 2793.9 for the 2-spiral problems. However the experimental results indicate that on average the number of weights will be 12.4 for the XOR and 551.4 for the 2-spirals problem. Again this means that the actual number of weights is less than the theoretical further emphasizing that the equations presented earlier give the worst-case implementation for the network.

From the results presented in [5,6], it is possible to train a neural network using LPW. However consideration needs to be given to the weight range, p. If p is chosen to be a small value, then fewer hyper-planes will be implemented, and the change of convergence is reduced. A possible solution to this is to add additional hidden layers to the network.

6.3 Discussion and Conclusions

A driving factor in the design of hardware neural networks within this thesis is the need for efficient, biologically plausible and compact systems. A key component of a HNN which affects the size of the network is the device used to store the synaptic weight. In digital HNNs synaptic weights are stored using either RAM or ROM devices. In analogue HNNs synaptic weights are stored as charge on a FG devices. In either case the size of the storage device can account for the majority of the silicon. In addition to this the size of the storage device can affect the number and precision of the synaptic weights. A large storage device can allow for a high precision and large range of possible weights; conversely a small device gives a smaller range and in some cases a reduction in the precision of the weights. Therefore there is a trade-off between the size of the storage device and the range/precision of the synaptic weights. However simply reducing the number of possible weights poses several problems when implementing a neural network. The main problem is that a reduction in the number of weights can cause the network to fail to converge to a solution for its intended problem. A solution to this is to use LPW and training algorithms which utilize LPW it is possible overcome this problem. LPW allow for networks to be implemented and trained using a reduced number of and/or precision for the synaptic weights.

The algorithms presented in this chapter have been shown to be effective in training neural networks which utilized LPW. From the algorithms presented, a common theme has arisen in that consideration of the range and precision of the weights must be taken into account. If the weight range and precision is too small then the training algorithms will not be able to train the network. However by increasing the number of hidden layers within the network it is to be possible to train a neural network with a small weight range and precision for the same problem. Therefore LPW allows for a reduction in the synaptic weight storage device while permitting the network to function as intended. While the algorithms presented within this chapter have been shown to work with digital hardware neural networks further investigation in to their use in analogue hardware neural networks is need.

For the FG device presented in chapter 4 and Fig. 6.1 the implications of using LPW are now discussed. Since LPW can allow for a smaller FG device to be used, devices with a large tunnelling area (2.0µm x 2.0µm and 2.5µm x 2.5µm, Fig. 6.1) do not have to be implemented. This reduces the required silicon required for the CTS (with integrated FG device). However the minimum sized FG device (with tunnelling area of 0.35µm x 0.40µm) could be too small to use with LPW as the number and resolution of weights possible is small compared to the other potential FG devices. This would have to be investigated further. As LPW is dependent upon the number and precision of the synaptic weights, then no matter what size FG device is used, consideration of how Q_w maps to each synaptic weight must also be considered. Q_w can either directly map to the synaptic weight, w $\rightarrow Q_w$ or Q_w can be split into different packets/range of charge to represent each synaptic weight w $\rightarrow Q_{wpakcet}$. Using LPW with w $\rightarrow Q_w$ implies that a smaller FG device could be used, as a reduction in device size would correspond to a reduction in the total number of weights available. The precision of each weight remains fixed. If LPW is used with w $\rightarrow Q_{wpakcet}$, consideration must be made to the device size and the number of weights required. If the number of weights is fixed prior to implementation then the size of FG device affects the precision of the weights only. A reduction in device size will reduce the precision of each weight. However if the number of weights is not fixed, then a reduction in device size will reduce both the number and precision of each weight available.

LPW offers a potential solution in the quest to design a compact FG device for use with CTS. However consideration of the algorithm which is to be used along with the potential number and precision of synaptic weights required must be conducted prior to fabrication. A FG with a small tunnelling area can lead to a weight range and precision which prevents the training algorithms from training the network. However by increasing the number of hidden layers within the network it is to be possible to train a neural network but this can lead to an overall increase in the size of the neural network as a whole. Using LPW with larger FG device, while increasing required silicon area can permit the network to function as intended.

6.4 References

[1] M. Hohfeld and S. E. Fahlman, “Probabilistic rounding in neural network learning with limited precision”, *Neurocomputing,* no. 4, pp. 291-299, 1992.

[2] S. Draghici and I. K. Sethi, “On the possibilities of the limited precision weights neural networks in classification problems”, *WANN '97 Proceedings of the International Work-Conference on Artificial and Natural Neural Networks: Biological and Artificial Computation: From Neuroscience to Technology*, pp. 753-762, 1997.

[3] V. P. Plagianakos and M. N. Vrahatis, “Neural network training with constrained integer weights”, *Proceedings of the 1999 Congress on Evolutionary Computation*, pp. 2007-2013, 1999

[4] B. Jian, C. Yu, Y. Jinshou, “Neural networks with limited precision weights and its application in embedded systems”, *Second International Workshop on Education Technology and Computer Science*, pp. 86-91, 2010

[5] S. Draghici, “On the capabilities of neural networks using limited precision weights”, *Neural Networks*, no. 15, pp. 395-414, 2002

[6] S. Draghici, “On the computational power of limited precision weights neural networks in classification problems: How to calculate the weight range such that a solution will exists”, *Lecture Notes in Computer Science*, vol. 1606, pp. 401-412, 1999

[7] E. Fiesler, A. Choudry and H. J. Caulfield, “A weight discretisation paradigm for optical neural networks”, *Proceedings of the international congress on optical science and engineering, ICOSE’90 SPIE*, vol. 1281, pp. 164-173, 1990

[8] K. Nakayama, S. Inomata and Y. Takeuchi, “A digital multilayer neural network with limited binary expressions”, *Proceedings of international joint conference on neural networks, IJCNN’90*, vol. 2, pp. 587-593, 1990

[9] J. M. Vincent and D. J. Myers, “Weight dithering and word length selection for digital back propagation neural networks”, *BT Technology journal,* vol. 10, no. 3, pp. 1180-1190, 1992.

[10] V. P. Plagianakos and M. N. Vrahatis, “Training neural network training with 3-bit integer weights”, *Proceedings of the Genetic and Evolutionary Computation Conference*, 1999

[11] V. P. Plagianakos, D. G. Sotiropolous and M. N. Vrahatis, “Integer weight training by differential evolution algorithms”, *Recent advances in circuits and systems,* N.E. Mastorakis Ed. World Scientific, 1998

[12] R. Storn and K. Price, “Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces”, *Journal of Global Optimization*, vol. 11, pp. 341-359, 1997

[13] G. B. Huang and H. Babri, “Upper bounds on the number of hidden neurons in feed forward networks with arbitrary bounded nonlinear activation functions”, *IEEE*, vol. 9, pp. 224-229, 1998

[14] Y. Chen, L. McDaid, S. Hall, and P. Kelly, “A programmable facilitating synapse device.”, *2008 International Joint Conference on Neural Networks, IJCNN 2008, 2008, Institute of Electrical and Electronics Engineers Inc*, pp.1615-1620, 2008

[15] X. Jin, S. B. Furber and J. V. Woods, “Efficient modelling of spiking neural networks on a scalable chip multiprocessor”, *2008 International Joint Conference on Neural Networks, IJCNN 2008, 2008, Institute of Electrical and Electronics Engineers Inc,* pp. 2812-2819, 2008

[16] M.M. Khan, D.R. Lester, L.A. Plana, A. Rast, X. Jin, E. Painkras and S.B. Furber, “SpiNNaker: mapping neural networks onto a massively-parallel Chip multiprocessor”, *2008 International Joint Conference on Neural Networks, IJCNN 2008, 2008, Institute of Electrical and Electronics Engineers Inc,* pp. 2849-2856, 2008

[17] M.M. Khan, J. Navaridas, X. Jin, L.A. Plana, J.V Woods and S.B. Furber, “Real-time application support for a novel Sock architecture”, *4th United Kingdom Embedded Forum 2008,* pp. 8-9, 2008

[18] A. D. Rast, S. Yang, M. Khan, S. B. Furber, “Virtual synaptic interconnect using an asynchronous Network-on-Chip”, *2008 International Joint Conference on Neural Networks, IJCNN 2008, 2008, Institute of Electrical and Electronics Engineers Inc,* pp. 2727-2734, 2008

[19] T. Pfeil, T. C. Potjans, S. Schrader, W. Potjans, J. Schemmel, M. Diesmann, K.Meier, “Is a 4-bit synaptic weight resolution enough? – Constraints on enabling spike-timing dependent plasticity in neuromorphic hardware”, available arXiv:1201.6255v4

[20] R. Cattell and A. Parker, "Challenges for brain emulation: Why is building a brain so difficult?”, available http://synapticlink.org/

[21] J. Schemmel, D. Bruderle, A. Grubl, M. Hock, K. Meier and S. Millner, “A Wafer-Scale Neuromorphic Hardware System for Large-Scale Neural Modelling”, *International Symposium on Circuits and Systems (ISCAS) 2010*, pp. 1947-1950, 2010

Chapter 7 – Conclusions and Future Work

Hardware neural networks offer a promising and effective alternative to traditional computational systems. It is possible to replicate parallelism of interconnecting neurons, synapses and synaptic plasticity using semiconductor devices. These biologically inspired neuronal circuits aim to solve problems where conventional computational circuits/solutions are complex or do not exist. Pattern recognition, image processing, noise cancellation within mobile devices, medical implants such as silicon retinas/optical implants, control and robotic systems are some examples where neural networks are frequently utilized. Hardware neural networks offer a greater deal of biological plausibility, flexibility and potential in terms of parallelism, real-time operation and speed over their software counterparts. Synaptic plasticity is a fundamental part of a synapse with respect to learning and long-term memory. The adaptation, modification, of synaptic weight within of a synapse to various input stimuli determines the effect that one neuron has upon another. A key synaptic plasticity learning rule in biology is spike timing dependent plasticity, STDP. This thesis presented a method for storing synaptic weight as well as a compact circuit which implements STDP. Test devices were fabricated and tested over 3 chip runs; each chip was fabricated using a 0.35μm process from Austria MicroSystems (AMS). In this chapter an overview of the work undertaken within this thesis was presented with suggestions for future work.

Chapter 1 presented a brief review of the fundamentals of neural networks. Typical features of biological neurons and synapses were identified as well as the importance of synaptic plasticity in the networks ability to adapt and learn from various input stimuli. A review of current hardware neural networks was also presented which identifies the various requirements and constraints for implementing biologically plausible networks. It has been identified that HNNs need to be biologically plausible, low-powered, small, scalable circuits in order to implement efficient, effective and useful neural networks.

A review of spike timing dependent plasticity, STDP, in hardware neural networks was presented in chapter 2. The circuits presented were either symmetric with respect to pre/post synaptic spikes or of a decision circuit type. However while it has been shown to be possible to implement STDP within HNN, little consideration has been given as to how achievable the functionality of more complex circuits; that is, the scalability of the circuit blocks. A trade-off between the accuracy of the STDP implemented and the need of additional circuitry. For STDP to be biologically accurate for example by implementing ion channel functionality then complex mathematical modelling and dedicated processors are required, as is used in SpiNNaker, the Blue Brain Project and the Facets project. Using biologically plausible circuits allow for the implementation of compact CMOS circuits.

When scaling is considered, it is clear to see that the additional circuitry required to implement STDP will take up the majority of the silicon area. Therefore an alternative method of implementing synaptic plasticity within HNNs needs to be considered. These new methods must allow for the update in synaptic weight while maintaining a compact and low power circuit design, such that more complex neural networks can be constructed. Future work will look at alternative biologically synaptic weight update methods which can be implemented in HNN using compact circuits. The proposed methods will be assessed both for their feasibility when implementing a neural network and their biological plausibility.

A summary of the relevant semiconductor physics was given in chapter 3. An explanation of the operating modes for an MOS capacitor was provided and explained with respect to CV characteristics. Non-ideal effects on the performance of the ideal CV curve and device operation were also considered along with an overview of Fowler-Nordheim tunnelling. The principles of operation of MOS transistor were also discussed with respect to IV characteristics. Various device parameters were extracted from IV and CV plots from simulation results and experimental results. These were then compared in conjunction with nominal AMS process values where provided. Test devices were fabricated and tested during the 1st chip run using a 0.35μm process from AMS.

Chapter 4 outlined a FG device which can be integrated with the charge-coupled synapse. The device is designed using a polysilicon and MOS capacitor; the gate of the MOS capacitor and lower plate of the polysilicon capacitor forms the electrically isolated floating gate. The theoretical operation and design equations presented are derived from MOS physics to produce design guidelines. A theoretical model has been developed and compared with experimental results obtained from fabricated devices. The obtained experimental results from both HFCV and PCV measurements indicated that negative charge stored and removed from the floating gate occurs via Fowler-Nordheim tunnelling. The application of a large negative voltage to the control gate caused electrons to tunnel back through the gate oxide into the semiconductor. This served to reduce the number of electrons on the floating gate and increases its potential. Test devices were fabricated and tested during the 2nd chip run using a 0.35µm process from AMS. The experimental HFCV results presented confirmed that FN tunnelling does occur and caused charge to be stored on the FG. This was shown by a shift of the HFCV plot to the right of the ideal HFCV. The transient charge storage characteristics have been modelled using physical equations. Experimental results obtained using the pulsed CV technique were used to validate the model.

Chapter 5 presented a compact STDP circuit for use with floating gate synapses. The compact STDP circuit performs synaptic weight potentiation and depression. The circuit has advantages over current implementations in that it can implement a critical timing window for synaptic modification. The duration of the critical timing window was set by the subthreshold current controlled by the voltage, V_{leak}, applied to transistor M_{leak} in the circuit. Simulation and experimental results of the WP indicated that for a post-pre spiking event, no update of the synaptic weight occurs. A pre-post spiking event caused the synaptic weight, represented as charge on a FG, to be increased. The amount, by which the synaptic weight is changed, Δw, was determined by the duration that V_{wi} was greater than 1.2V and also by the magnitude of V_{CG}. The maximum weight, Δw_{max} was obtained when V_{CG} had a pulse width of 10µs and a constant magnitude of 10V. The minimum weight Δw_{min}, prior to V_{wi} being less than 1.2V was achieved when V_{CG} had a pulse width of 1µs and magnitude of 9.6V. Similarly simulation results for the WD circuit block indicated that a pre-post spiking event caused no update of the synaptic weight occurs. A post-pre spiking event caused the synaptic weight to be decreased. Both the WP and WD circuits have a power consumption of approximately 2.4mW, during a weight update. If no weight update occurs the resting currents within the device are in the nA range, thus each circuit has a power consumption of approximately 1µW.

Since the critical timing window was determined by subthreshold transistor Mleak, the effects of process variation were also considered. It was shown that process variation can be modelled simplistically as a variation in the nominal, ideal, threshold voltage, V_t. This was verified through the comparison of simulation results for the simple model with Monte Carlo analysis simulations undertaken using Cadence. In addition to this both wafer-to-wafer and die-to-die process variations have a dramatic effect upon the critical timing window. By choosing a suitable value of V_{leak} it was possible to tune t_{CW} to ensure a biologically plausible timing window despite the effects of process variation. Additionally the amount of charge injected on to the FG and the controllability of this charge was also been presented. A physical model was presented which allows design and analysis of the blocks.

Limited precision weights and its potential use in hardware neural networks was considered in chapter 6. LPW offers a potential solution in the quest to design a compact FG device for use with CTS. The algorithms presented in chapter 6 have been shown to be effective in training neural networks which utilized LPW. A common theme in the algorithms presented is that consideration of the range and precision of the weights must be taken into account. If the weight range and precision is too small then the training algorithms will not be able to train the network. However by increasing the number of hidden layers within the network it is to be possible to train a neural network with a small weight range and precision for the same problem. Therefore LPW allows for a reduction in the synaptic weight storage device while permitting the network to function as intended. While the algorithms presented within chapter 6 have been shown to work with digital hardware neural networks further investigation in to their use in analogue hardware neural networks is needed.

By considering which algorithm are to be used along with the potential number and precision of synaptic weights required must be conducted prior to fabrication. A FG with a small tunnelling area can lead to a weight range and precision which prevents the training algorithms from training the network. However by increasing the number of hidden layers within the network it is to be possible to train a neural network but this can lead to an overall increase in the size of the neural network as a whole. Using LPW with larger FG device, while increasing required silicon area can permit the network to function as intended.

This thesis has shown that it is possible to implement a biologically plausible version of STDP in hardware using a compact, scalable circuit. In addition to this it has also shown that the STDP circuit can be used to add and remove charge from a floating gate device. This represents the storage and modification of synaptic weight. Future work should focus upon the integration of the STDP circuit and FG device with silicon synapse and neuron circuits. This is to assess whether the proposed circuits are compatible and the overall functionally which can be achieved. Secondly consideration of the construction of large scale neural networks using the proposed devices also needs to be undertaken. In order to accomplish this, there is a need to examine how to interconnect these devices. Conventional metal interconnects will quickly dominate the chip area when scaling the network. A time multiplexing architecture is one potential solution to this problem. Further work should also consider how to further reduce the size of the STDP circuit and synaptic weight storage device proposed in this thesis. The use of high-k dielectrics is one particular avenue which could be explored. A high-k dielectric material has a higher permittivity then that of silicon, this allow for the reduction in the area of devices such as capacitors, without impacting on their required value of the capacitor. Additionally future work should also consider the fabrication of the proposed circuit using difference fabrication processes, such as 0.25µm, 0.18µm or using amorphous silicon. In addition to this consideration of specialised NVM processes or devices, such as memristors could further aid in the reduction in both circuit size and power consumption.

Publications

A. Smith, L.J. McDaid, S. Huang and S. Hall, "A compact spike-timing-dependent-plasticity circuit for floating gate weight implementation", Neurocomputing – Accepted subject to modifications.

A. Ghani, L.J. McDaid, A. Belatreche, S. Hall, S. Huang, J. Marsland, T. Dowrick and A. Smith, "Evaluating the generalisation capability of a CMOS based synapse", Neurocomputing, vol. 83, pp. 188-197, 2011

A. Smith, L.J. McDaid, and S. Hall, "Implementing spike timing dependent plasticity in hardware neural networks", To be submitted.

A. Ghani, L.J. McDaid, A. Belatreche, P. Kelly, S. Hall, S. Huang, J. Marsland, T. Dowrick and A. Smith, "Evaluating the training dynamics of a CMOS based synapse", International Joint Conference on Neural Networks, San Jose, California, July 31 – August 5, 2011

A. Smith, "Synaptic weight storage and update in silicon neurons", ESSDERC/ESSCIRC Fringe Poster Session 2010, 4 pages, available on IEEE Xplore.

A. Smith, "A novel STDP control circuit for use with floating gate synapse", Virtual Worldwide PhD Forum for PhD students in Design and Automation (VW-FEDA), University of Southampton, November 30, 2011.

J.S. Marsland, S. Hall, S. Huang, T. Dowrick, A. Smith, L.J. McDaid, J. Harkin and A. Ghani, "Limited precision weights: a way forward for hardware neural networks?", UK Design Forum, Chancellor's Conference Centre, Manchester, April, 2011.

Appendix

Fig. A.1 and A.2 present the schematic and PCB layout of the transimpedance amplifier, TA. The TA is used in chapter 4 to amplify and convert the displacement current, i_{disp} into a voltage during PCV testing. The TA was designed by Mr. A. Edwards.

The input impedance to the negative input of the TA is approximately zero. A series of feedback resistors, R_f can be connected from the output of the TA to the negative input allowing for a choice in the gain of the amplifier. When a current, i_{in}, is applied to the negative input terminal the output voltage of the TA, V_{out}, is given by equation A.1. R_f can be 1KΩ, 10KΩ, 50KΩ, 100KΩ or 1MΩ.

$$V_{out} = i_{in} R_f \tag{A.1}$$

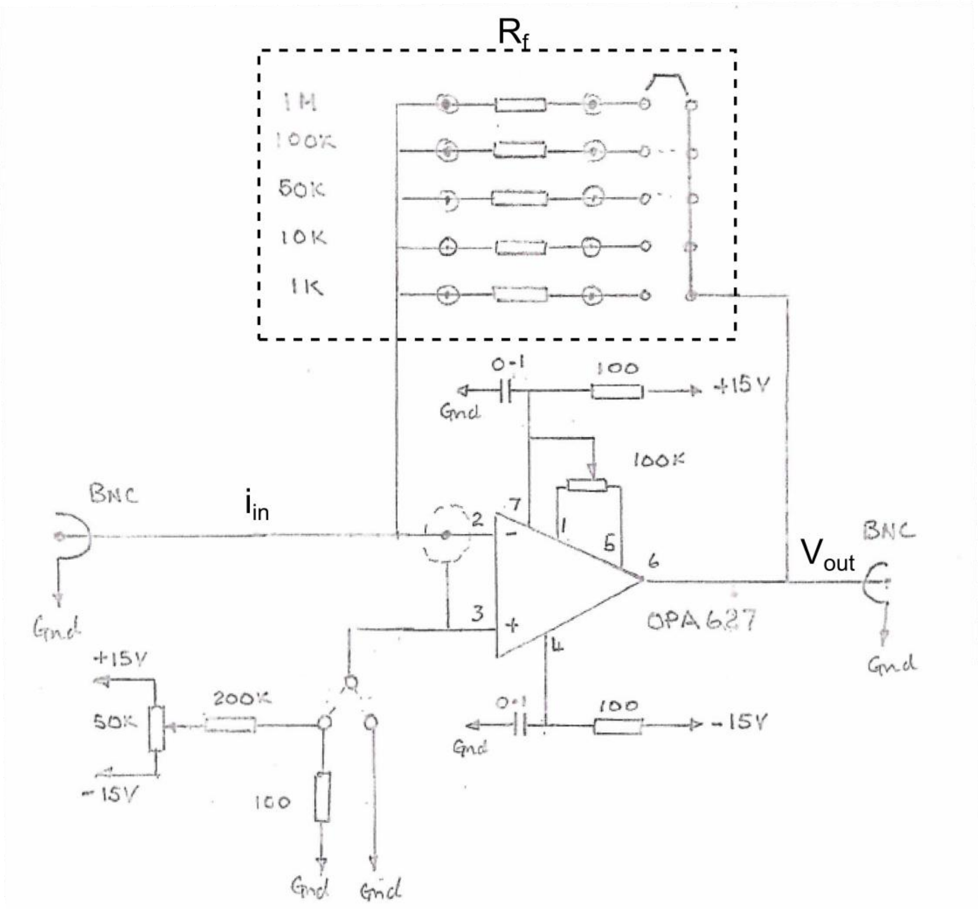


Fig A.1 – Transimpedance amplifier schematic – designed by Mr. A. Edwards

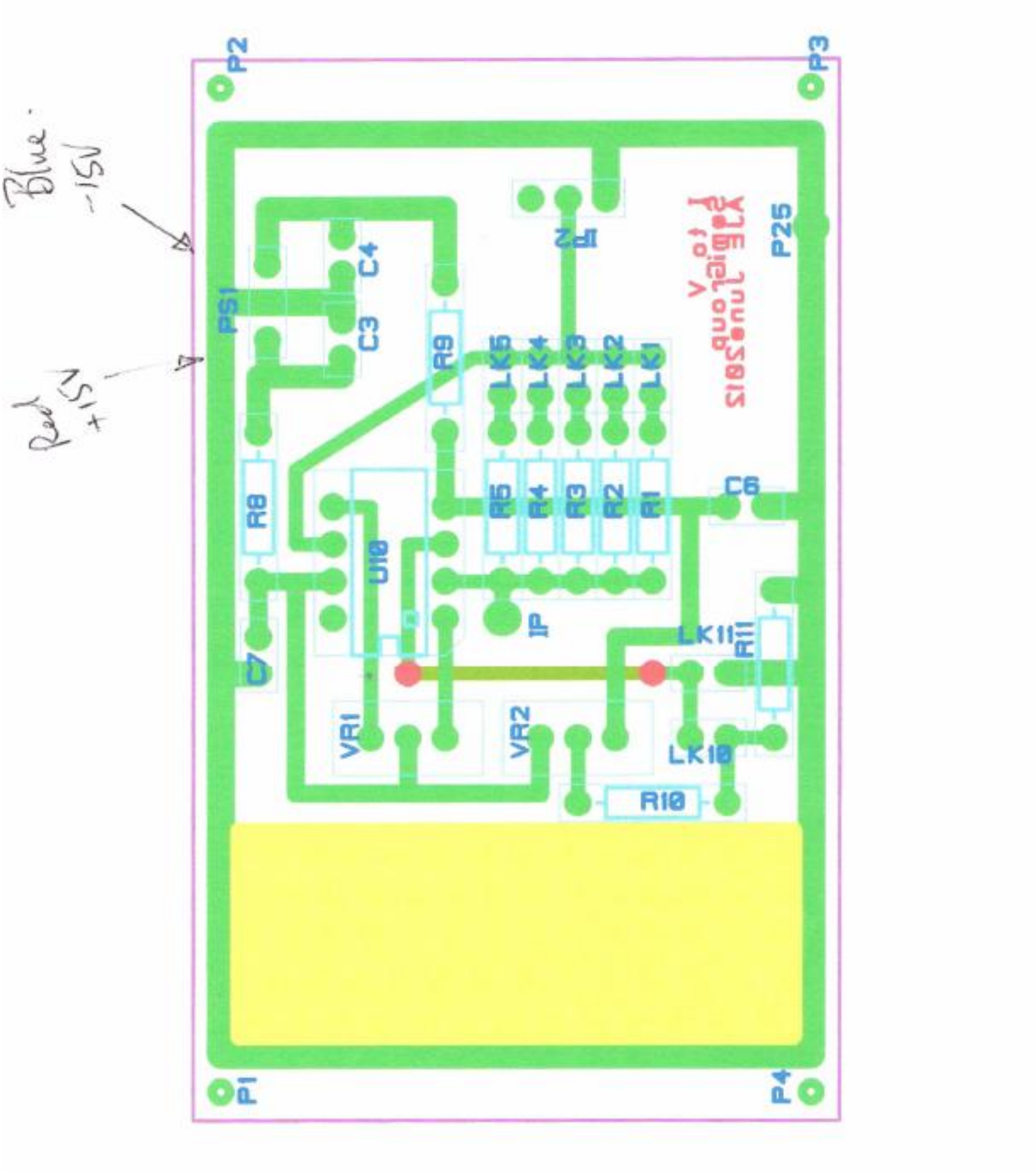


Fig A.2 – Transimpedance amplifier PCB layout – designed by Mr. A. Edwards

All test chips were fabricated in a three metal layer 0.35µm n-well process provided by Austria MicroSystems (AMS). Fabrication was coordinated through the Europractice service. All layouts were created and verified using the Cadence software package configured for the 0.35µm AMS process. Three separate prototype chips were produced. The first was received in June 2009, the second in June 2010 and August 2011.

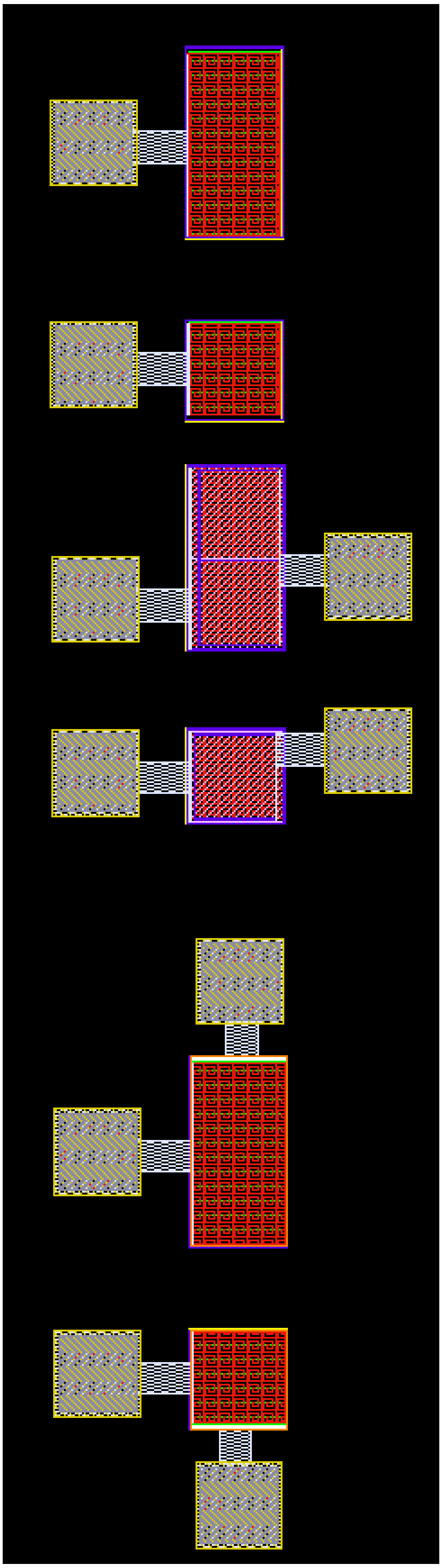


Fig A.3 – MOSC (p-type and n-type) and polysilicon capacitor test devices – dimensions 50µm x50µm and 100µm x 50µm

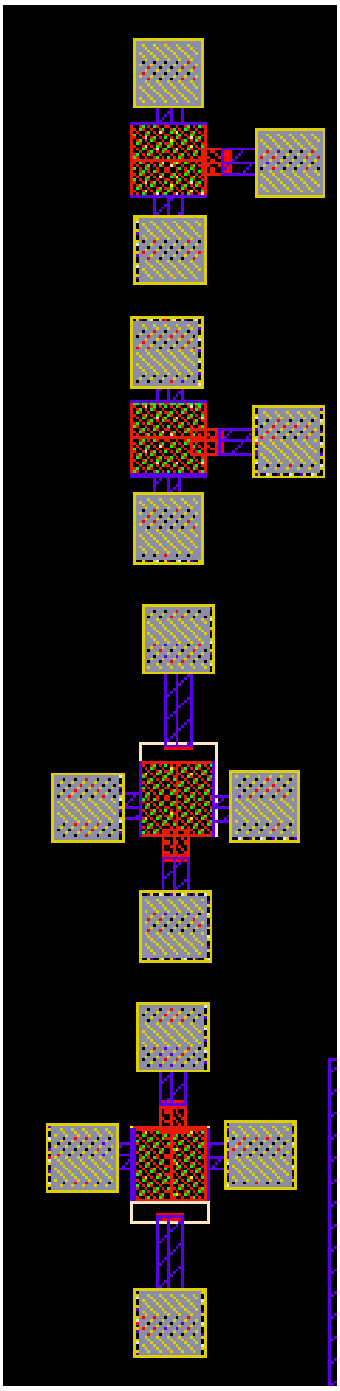


Fig A.4 – MOSFET (p-type and n-type) test devices – dimensions 100µm x 100µm

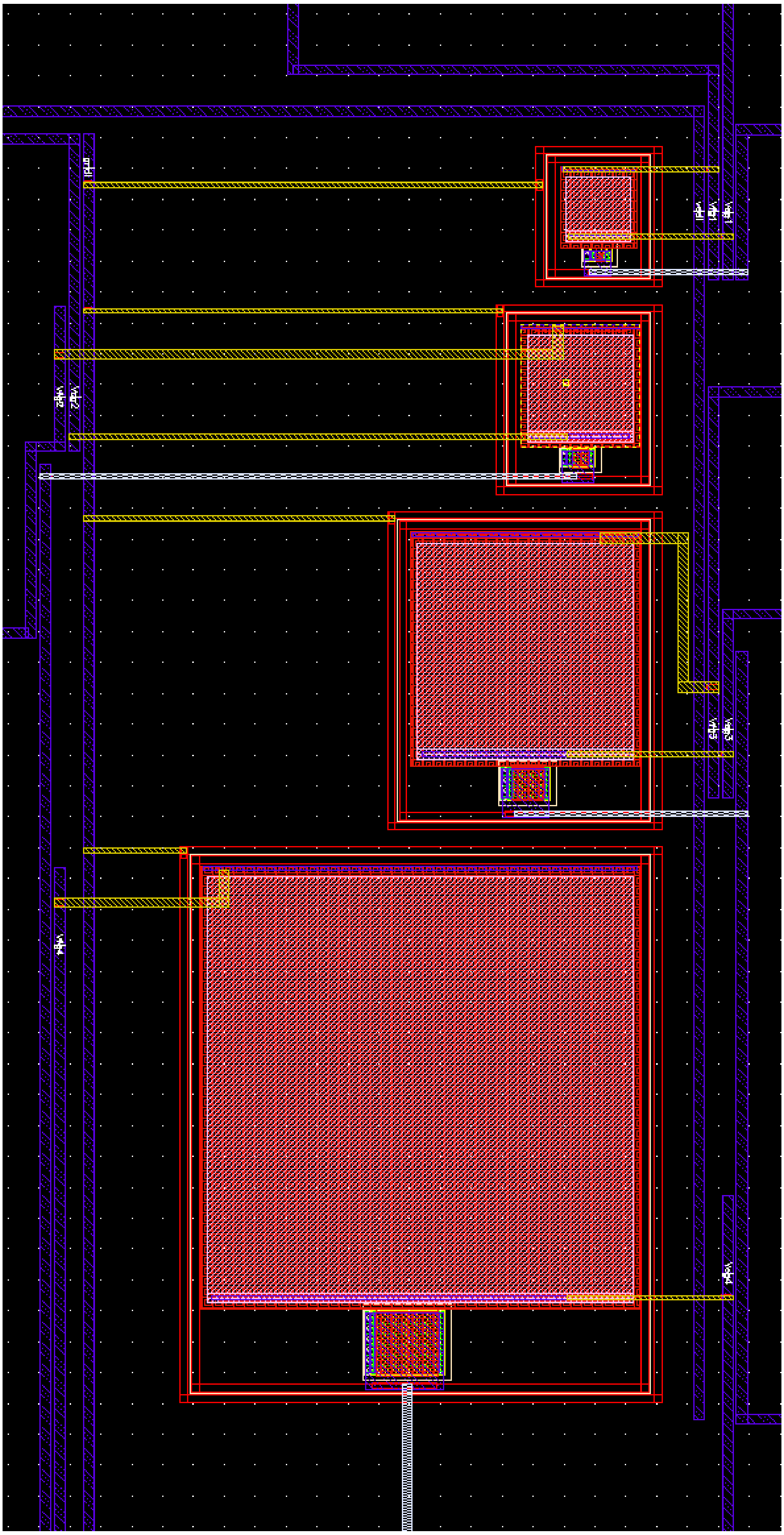


Fig A.5 – FG test devices

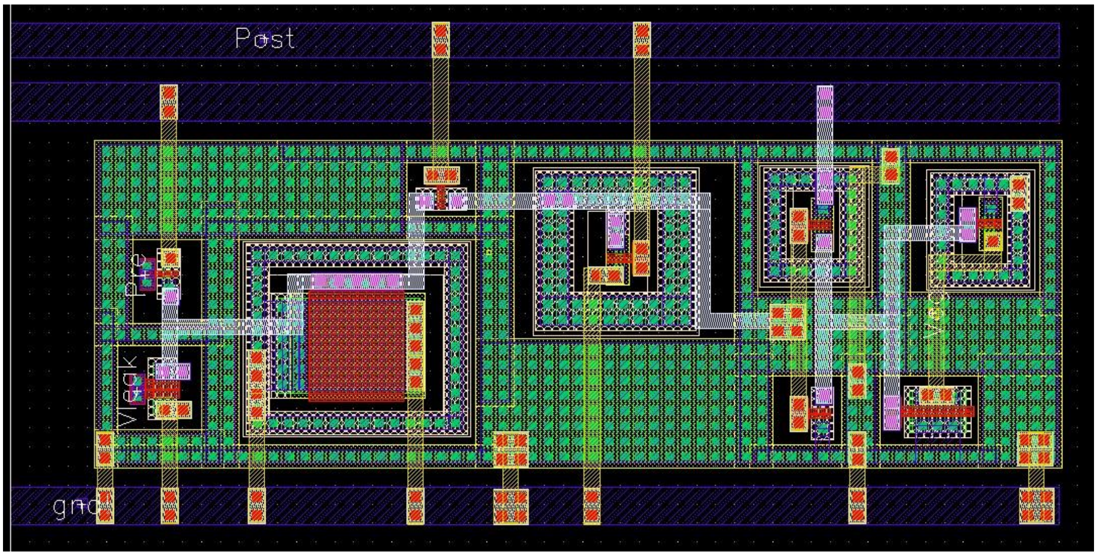


Fig A.6 – WP circuit and output buffer layout

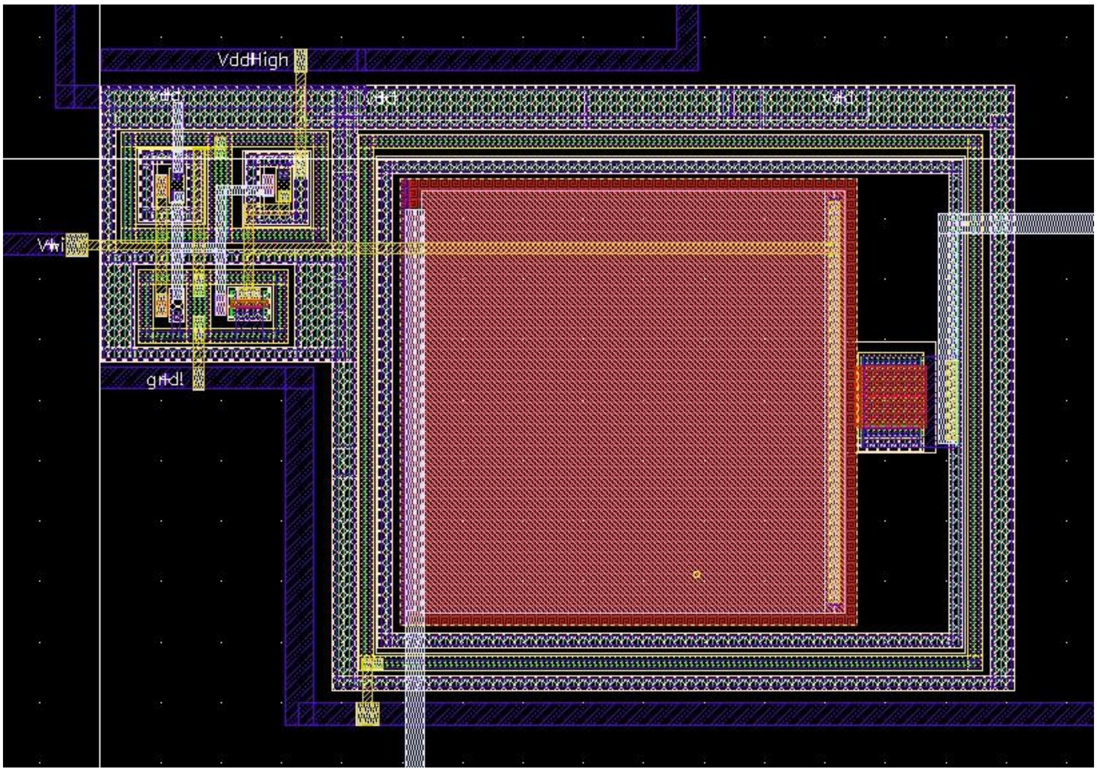


Fig A.7 – WP Circuit and FG Device Layout

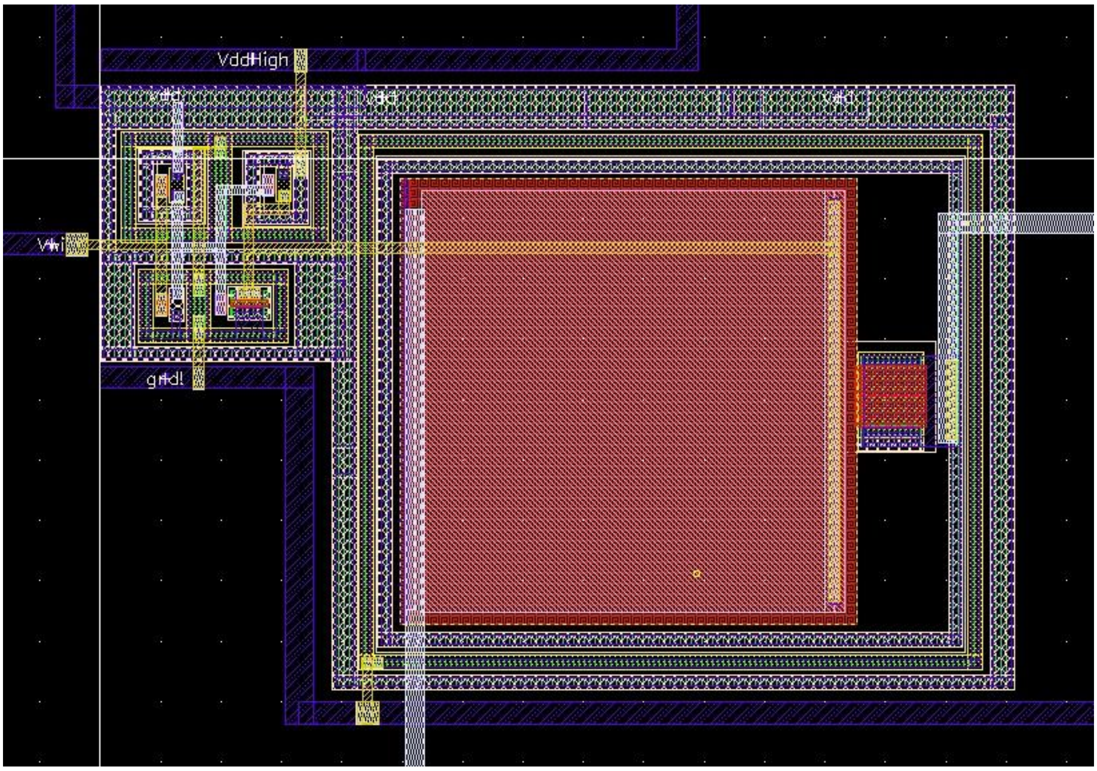


Fig A.8 – WP circuit and FG device layout