Standard CMOS Floating Gate Memories for Non-Volatile Parameterisation of Pulse-Stream VLSI Radial Basis Function Neural Networks

L. William Buchan

Thesis submitted for the degree of
Doctor of Philosophy
The University of Edinburgh
November 1997

Abstract

Analogue VLSI artificial neural networks (ANNs) offer a means of dealing with the non-linearities, cross-sensitivities, noise and interfacing requirements of analogue sensors (the problem of *sensor fusion*) whilst maintaining the compactness and low power of direct analogue operation. Radial Basis Function (RBF) networks, as a means of performing this function, have several advantages over other ANNs.

The pulse-stream ANN technique developed at Edinburgh provides the additional benefit of implicit analogue-digital conversion and signal robustness.

However, progressing this work requires the integration of high density analogue memory for parameterisation of the ANN since conventional weight refresh methods are too area and power hungry. For this purpose, standard CMOS floating gates have been proposed as these maintain the low process cost and easy availability of the neural circuitry.

Investigation of this proposition proceeded in three stages:

1. Evaluation of the suitability of a standard process for the fabrication of floating gates and exposure of the issues involved: feasibility, *analogue* programmability, layout optimisation and modelling.

2. The interfacing of floating gates to Radial Basis Function (RBF) neural network circuits and development of programming approaches to cope with potentially destructive characteristics of high voltages and currents.

3. Development of circuits for programming floating gates using continuous-time feedback to facilitate a rapid weight downloading phase from a software model.

Three chips were designed, fabricated and tested to explore each of these sets of issues. Detailed discussion and measurements are presented.

Conclusions have been drawn about layout optimisation, programmability and device aging and on the design and general suitability for purpose of standard CMOS floating gates. While these can be designed, interfaced to RBF circuits, and programmed to perform useful functions, their disadvantages make them more useful as a prototyping technique than as memory modules for inclusion in a final product.

Declaration

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

Acknowledgements

Specific individuals and organisations stand out in their contributions to this work and towards my time in the Neural Group.

First, I must thank my academic supervisors, Alan Murray and Martin Reekie for their help, advice and support over the last three years, and for providing an environment free from worries of funding or equipment.

I am grateful to the Engineering and Physical Sciences Research Council (EPSRC), which funded the RBF project. Special thanks go to the other two Edinburgh participants in the the project: Robin Woodburn and David Mayes. Robin has taken an interest in my work from the onset and been an invaluable source of encouragement and advice, only rarely resorting to brutal sarcasm to make his point. To David I am indebted for all I ever wanted to know - and possibly even more - about the Edinburgh RBF circuit designs. Thanks are also due to Alister Hamilton and Drew Holmes who put me onto the floating gates 'angle' in the first place. And one day they are going to pay for it!

My studentship was funded by GEC Marconi Avionics Limited of Edinburgh which I'd like to thank for paying my fees and for keeping the wolf from the door. Special mention goes to Paul Holbourn of the Radar Systems Division who acted as my industrial supervisor.

Thanks go to fellow researchers for their valuable input to my knowledge or work. There are too many names to mention, but for particular contribution, I'd like to thank Tony Snell from the department, Gillian Marshall and Steve Collins from the Defence Research Agency at Malvern and Tor Sverre Lande of the Department of Informatics at Oslo University. I'd also like to thank Jim Goodall who provided the SEM photomicrographs.

I must offer my thanks to the other hardware bods of the Neural Group for their technical advice, Cadence-users comradeship and generally being a splendid gang of people. Special mention goes to Geoff Jackson, proud creator of a neural robot later to be ruthless scavenged for parts for a nightmarish experiment in giraffe locomotion; Kostas Papathanasiou for his unlimited enthusiasm; Torsten Lehmann for designing and laying out a chip in about three minutes and the detrimental effect this had on group morale; John Breslin for instigating the trend of turning development boards into rivals to the Blackpool illuminations; Cati Collin for her leadership of the Cardboard Rocket project and Mark Glover for reliably having even less luck with the CAD tools than me. Thanks also go to those members of the group who never dared enter the Hardware Zone or run a Cadence start-up script. Founts of knowledge on all things neural-ie or computer-ie the lot of them. And yes - they are a splendid gang of people too. Special

mention goes to Andy Myles for being, unquestionably, Andy Myles; Richard Reavy and Mike Smart, custodians of RSVP productions, putting a whole new meaning to the 2nd year poster; Emma Braithwaite for not bringing the squishy aspects of her project into too many lunchtime conversations; Pete Edwards for timing coffee breaks with atomic accuracy (or is that precision?), and Marcus Alphey and Ryan Dalzell though they may be for ever divided by windows managers, Andy Connelly, Neil Marston, Alex Astaras, Andrew Murray, Dwayne Burns, Xavier Parra, Ånen Abusland, Hao-Hsiung Yang, Olivier Chapuis, all the neural newbies, casual interlopers and all the people I can't remember (I haven't forgotten about you). I couldn't have hoped for a better bunch of colleagues.

Thanks go to my parents for their support and encouragement during my education. Finally, I must thank my fiancée, Carole Morrison, for her love, support - and quite outstanding tolerance - over the last three years.

Contents

1 Introduction **1**

1.1 Microelectronic Artificial Neural Networks 1

1.2 SLV-CMOS Floating Gate Memories 3

1.3 Project Objectives 3

2 Pulse Stream Radial Basis Function Circuits **5**

2.1 Introduction 5

2.2 Radial Basis Function Neural Networks 5

2.2.1 RBF Architecture 5

2.2.2 RBF Training 7

2.2.3 RBF Sensor Classifier 7

2.2.4 Requirements of a VLSI RBF 8

2.3 PWM Circuit Implementation 8

2.3.1 Euclidean Distance Calculator 9

2.3.2 Non-Linearity Calculation 11

2.3.3 Multiplier 17

2.4 Summary 19

3 Review of Long-Term Synaptic Weight Storage **20**

3.1 Introduction 20

3.2 Properties of Non-Volatile Technologies 20

3.3 Power-Up Non-Volatile Technologies 22

3.3.1 Leakage Limitation and Compensation 23

3.3.2 Weight Refresh and Regeneration 23

3.3.3 Local digital storage 23

3.3.4 Global Refresh 24

3.3.5 Local Regeneration 25

3.3.6 Continually Adaptive Circuits 26

3.3.7 Battery-Backed RAM 27

3.4 Power-Down Non-Volatile Memories 28

3.5 Floating Gate Devices 29

3.5.1 Ultraviolet (UV) Illumination 30

3.5.2 Fowler-Nordheim Tunnelling (FNT) 31

3.5.3 Hot Electron Injection 35

3.5.4 Digital Floating Gate Memories 37

3.5.5 Analogue Floating Gate Memories 45

3.6 Summary 50

4 Standard CMOS Floating Gates **51**
4.1 Introduction 51
4.2 Floating Gate Requirements 51
4.3 Limitations of SLV-CMOS 51
4.4 Review of Preceding Research in Standard Low-Voltage CMOS Floating Gate Memory Cells 52
4.4.1 Features Under Control: VLSI Layout 52
4.4.2 Corner Effects 52
4.4.3 Edge Effects 54
4.4.4 Thinning 55
4.4.5 Survey of SLV-CMOS Research 55
4.5 Discussion 60

5 *TARDIS*: A Floating Gate Test Chip **61**
5.1 Introduction 61
5.1.1 Design Motivations 61
5.1.2 Design Constraints 62
5.1.3 Layout of Tunnelling Capacitors 64
5.2 Experiments with Tunnelling 67
5.2.1 Voltage and Time Programming Dependency of Gate and Interpoly Oxides 70
5.2.2 Reprogrammability and Device Aging 72
5.2.3 Interpoly Tunnelling Capacitor Design Comparison 73
5.2.4 Capacitor Network Model 73
5.3 Analogue Memory Cell Applications 85
5.3.1 Thermal Trap Annealling 93
5.4 Data Retention and Accuracy 95
5.4.1 Long Term Data Retention 96
5.5 Hspice Modelling of the Floating Gate 98
5.5.1 Fowler-Nordheim Coefficient Derivation by Capacitor Network Model 99
5.6 Experiments with CHE Programming 105
5.6.1 Current Limitation 110
5.6.2 Self-Induced Programming 111
5.6.3 Aging and Retention 114
5.6.4 Polarised Detrapping 114
5.7 Discussion 114

6 *NEMO*: Non-Volatile RBF Subcircuits and Addressing 117
6.1 Introduction . . . 117
6.2 Design Motivations . . . 117
6.3 Issues in Programming Hardware ANNs . . . 118
6.3.1 Chip-in-the-Loop (CIL) Programming . . . 118
6.3.2 Programming Fidelity . . . 120
6.4 Chip Design Issues . . . 121
6.4.1 Choice of Floating Gate Designs . . . 121
6.4.2 RBF Components . . . 122
6.4.3 Programming Arrays . . . 122
6.4.4 *NEMO* Floorplan . . . 124
6.5 Floating Gate RBF Building Blocks . . . 124
6.5.1 Revised Floating Gate Layout . . . 124
6.5.2 Euclidean Distance Calculator . . . 125
6.5.3 Pulse Stream Multiplier . . . 131
6.5.4 Fowler-Nordheim Programming Characteristics . . . 133
6.6 Programming Algorithm and Accuracy . . . 135
6.6.1 Circuit Imprecision . . . 135
6.6.2 Algorithmic Improvements . . . 137
6.7 Programming Arrays . . . 140
6.8 High Voltage Switches . . . 140
6.8.1 Mietec High Voltage Devices . . . 146
6.8.2 Circuit Designs for High Voltage Switches . . . 147
6.8.3 Experimental Results . . . 150
6.9 Dual Phase Programming Arrays . . . 155
6.9.1 Circuit Implementation . . . 162
6.9.2 Programming Characteristics . . . 163
6.9.3 Retention Characteristics . . . 167
6.10 CHE Flash . . . 174
6.10.1 Programming . . . 177
6.10.2 Programming Accuracy . . . 178
6.10.3 CHE Program Disturbance . . . 182
6.10.4 CHE Multiplier Array . . . 183
6.10.5 Programming Retention . . . 183
6.11 Conclusions . . . 185
6.11.1 RBF Subcircuits . . . 185
6.11.2 Comparison of Programming Techniques . . . 185
6.11.3 Layout . . . 188

7 *PARAFIN*: Feedback Programming of RBF Arrays 189
7.1 Introduction . . . 189
7.1.1 Design Motivations . . . 189
7.1.2 Choice of Addressing Scheme . . . 190
7.2 Review of Programming Techniques . . . 190
7.2.1 Decaying Sinusoid . . . 190
7.2.2 Threshold Drain Switch . . . 193
7.2.3 Continuous-Time Injection . . . 194
7.3 *PARAFIN* Programming . . . 195
7.4 *PARAFIN* Floorplan . . . 199
7.5 Feedback Programming of Euclidean Array . . . 199
7.5.1 Experimental Results . . . 206
7.6 Feedback Programming of Multiplier Array . . . 212
7.6.1 Experimental Results . . . 217
7.7 Feedback Programming of Non-Linearity Array . . . 224
7.7.1 Circuit Description . . . 224
7.7.2 Experimental Results . . . 226
7.7.3 Width Mapping Characteristics . . . 226
7.7.4 Disturb Protection and Programming Endurance . . . 227
7.7.5 Programming Accuracy . . . 228
7.8 Suggested Future Work: Offset Compensation . . . 228
7.9 Retention . . . 229
7.10 Discussion . . . 230

8 Discussion and Conclusions 232
8.1 Introduction . . . 232
8.2 Experimental Observations and Conclusions . . . 232
8.2.1 *TARDIS* . . . 232
8.2.2 *NEMO* . . . 234
8.2.3 *PARAFIN* . . . 236
8.2.4 Project Summary . . . 236
8.3 Observations on Non-Volatile ANN Weights . . . 240
8.4 Observations on the Applications of Analogue SLV-CMOS Floating Gate Memories to ANNs . . . 240
8.5 Summary . . . 242

A Layout, Bonding and Pin Diagrams 244
A.1 Introduction . . . 244
A.2 Cell Layout Sizes . . . 248

B Test Equipment and Programmer Boards **249**
B.1 Introduction 249
B.2 *TARDIS* 249
B.3 *NEMO* 249
B.3.1 *NEMO* Board 1 251
B.3.2 *NEMO* Board 2 251
B.3.3 *NEMO* Board 3 257
B.3.4 *NEMO* Board 4a & 4b 259
B.4 *PARAFIN* 259
B.4.1 *PARAFIN* Board 1 259
B.4.2 *PARAFIN* Board 2 261
B.4.3 *PARAFIN* Board 3 261
B.5 PC Interfaces 261

C Dickson Charge Pump **264**

D Common CMOS Failure Modes **266**
D.1 Introduction 266
D.2 Trap Up 266
D.3 Oxide Breakdown 268
D.3.1 Intrinsic Oxide Breakdown 269
D.3.2 Defect-Related Oxide Breakdown 269
D.4 Electromigration 270
D.5 Avalanche Breakdown 270
D.5.1 Gate-Field Enhanced Breakdown 271
D.5.2 Snapback 271
D.6 Punchthrough 271
D.7 Latch-Up 272
D.8 Field Transistors 273
D.9 Electrostatic Discharge Damage (ESD) 273

E Bake Retention Testing **275**
E.1 Introduction 275
E.2 Thermionic Emission Theory 275
E.2.1 Boltzmann Distribution of Electrons 275
E.2.2 Thermionic Charge Loss 276
E.2.3 Arrhenius Model 276
E.3 Commercial EEPROM Testing 277

F Publications and Articles **279**
F.1 Publications 279
F.2 Other Articles 279

References **280**

Chapter 1

Introduction

1.1 Microelectronic Artificial Neural Networks

Artificial neural networks (ANNs) are highly abstracted and formalised models of natural neural systems which owe more to statistics than to biology. Their architecture is radically different from that of a conventional computer and their computational power depends on an adaption of the parameters ('weights') defining the interconnect and response of large numbers of very simple processors. ANNs are applied to problem domains where algorithmic approaches are difficult, favouring instead a adaption of the weights of the nonlinear model implemented by the ANN to the form defined by training data.

ANNs can be developed rapidly in software (either using custom code or off-the-shelf packages), which provides easy flexibility for experimentation and customisation of algorithms. If the application demands speeds higher than can be serviced by generic serial computers, gains of several orders of magnitude can be obtained by mapping algorithms into fast DSP technology, especially if the inherent parallelism is exploited. For embedded systems, such as domestic appliance control, digital ASICs, which can easily be constructed by silicon compilers, can lead to significant reductions in physical system size.

Recent advances in micromachining, smart materials, biotechnology and in other related fields have led to a remarkable growth in practical integrated sensor technologies. Many of these new sensors produce analogue data streams, which in addition may be *(i)* non-linear with regard to the physical stimuli, *(ii)* contain offsets, *(iii)* contain parasitic cross-sensitivities and *(iv)* subject to noise.

Certain applications may further demand banks of sensors which may be poorly matched or even of entirely different types. Fusing of such data into sensible and usable outputs may be a complex task.

Hardware ANNs may offer a solution to this problem in some cases. Demand for small, low-power, integrated packages (which in the case of so-called smart sensors may even incorporate *on chip* the sensor element itself) plays directly to the strengths of *analogue* VLSI ANNs since these provide compactness, low power and the ability to operate directly on analogue data streams. Typical systems interface to analogue sensor(s), provide signal conditioning, and classification or signal manipulation, and

provide digital output in a form suitable for subsequent digital processing or interfacing to standard digital busses. The use of the analogue/pulse-stream technique developed at Edinburgh University Electrical Engineering Department for building VLSI ANNs is pertinent here, since with analogue inputs and easily sampled pulse-coded outputs, the normally expensive analogue-digital conversion is implicit in the neural signal processing.

Gain and offset trimming in existing analogue sensor processing circuits has been achieved by potentiometers or via D/A conversion from on-chip RAM/ROM. However, such approaches are inappropriate for the high analogue data density demanded by ANNs whilst maintaining compact area and lower power; small, non-volatile, on-chip, analogue memory elements are required. Unfortunately such sensor systems are predominately a low volume product, constrained by a highly fragmented applications market [174] so it is not practical to resort to special, high cost, non-volatile fabrication processes to address this need. In fact it is imperative to control costs by subcontracting sensor processing circuits out to compliant standard process fabrication as far as possible, limiting the sensor integration to minimal post-processing steps, wafer-wafer bonding or hybrid integration.

The preceding discussion leads to two premises which form the basis of this work:

1. aVLSI ANNs for embedded sensor processing (intelligent analogue-to-digital conversion) require a non-volatile memory technology in order to comply with low power and area budgets.

2. High volume applications are rare; such commercial constraints depend on the development of systems which are *integrable in widely-available (and thus low cost) generic fabrication processes*. In addition, from a smart sensor perspective, where microstructures may be added in intrusive post-processing steps, standard multi-purpose processes with conservative design rules tend to be more compliant [174]. Additionally, without process changes, existing designs and cell-libraries are available, allowing for simpler and more convenient implementation.

For this reason, the development of non-volatile synaptic memory for pulse-stream neural networks was done in a low-cost standard low-voltage CMOS process (SLV-CMOS) which had not been intended for the design of floating gate memory cells.

A further practical reason for this choice was the desire to build such circuits in a fabrication process easily available for future extensions or applications by the Edinburgh research group or even for adoption by other academic groups. At the time of writing neither Europractice (Europe) nor MOSIS (USA) offered special non-volatile processes to its academic customers. Use of standard processes also opens up greater opportunities for fabrication second-sourcing.

1.2 SLV-CMOS Floating Gate Memories

The only non-volatile technology completely compliant with SLV-CMOS is the *floating gate* memory. Floating gates are electrically isolated capacitors whose level of stored charge modulates the behaviour of the 'sense' transistor which it drives. Although in the past floating gates have always been implemented in special fabrication processes and only used for digital data storage there has been much interest (and some success) in recent years in confronting the problems of implementing SLV-CMOS floating gates and also using floating gates to store analogue data.

1.3 Project Objectives

The particular ANN of choice in this thesis is the radial basis function (RBF) neural network. This architecture offers some advantages over the ubiquitous multi-layer perceptron (MLP) which will be mentioned in course and has also been subject of concurrent work mapping the RBF feedforward algorithm into pulse-stream analogue VLSI implementation using volatile capacitor weight storage with RAM-refresh [110]. The aim of the project then was to replace the capacitor used for weight storage in RAM-refreshed RBF circuits with a SLV-CMOS floating gate. The aim was not to build a complete non-volatile RBF chip (although that would be an ultimate goal) but rather to

- Investigate the optimisation of SLV-CMOS floating gate design (for programmability) including following up issues of contention in the literature.
- Characterise the pertinent non-documented features of the available SLV-CMOS process for selection of programming mechanism and modelling.
- Build non-volatile floating gate adaptations of existing RBF subcircuit designs.
- Develop SLV-CMOS circuits for on-chip addressable programming of the floating gates.
- Develop algorithms for iterative downloading of network weights to the RBF from a software model.
- Extend the RBF subcircuit designs to allow continuous time feedback programming of the floating gates to speed the downloading phase.

Initially the long-term non-volatility of the floating gates was also to be investigated in depth but this proved to be both more complicated and problematic than first envisaged and so has only been treated briefly to restrict the scope of the research.

The potentially complicated solid state physics issues behind floating gates such as electron transport and oxide systems is an extremely broad subject and often requires a high degree of scientific expertise, specialised laboratories and custom fabrication. Whilst these issues have not been ignored, the balance of treatment in this thesis is more abstract, considering SLV-CMOS floating gates as an *engineering solution* to non-volatile weight storage in analogue neural sensor interfaces.

During the course of the project three test chips were designed and fabricated. Thus the thesis has been divided into two main sections, a review of ANN design, non-volatile memory and SLV-CMOS floating gate specifics followed by three chapters of experimental results, one on each of the chips:

Part One		
Chapter 2		- An introduction to the RBF algorithm and a review of pulse stream elements for constructing a VLSI implementation.
Chapter 3		- A review of non-volatile weight storage focusing on the context and technology of floating gates.
Chapter 4		- A review of SLV-CMOS specific issues for floating gate implementation, current literature and specific context of the work.

Part Two		
Chapter 5	$TARDIS$	- Non-neural floating gate test chip, investigating floating gate optimisation and programming mechanisms.
Chapter 6	$NEMO$	- RBF subcircuits based on SLV-CMOS floating gates, programming circuits and algorithms.
Chapter 7	$PARAFIN$	- Modified RBF subcircuits for continuous-time feedback rapid programming of floating gates by development of one technique tested on $NEMO$.

Finally Chapter 8 summaries the work and its findings and then discusses the appropriateness of SLV-CMOS to implementing neural analogue sensor interfaces.

Chapter 2

Pulse Stream Radial Basis Function Circuits

2.1 Introduction

This chapter describes the radial basis function (RBF) neural network and existing analogue pulse stream circuits for its implementation in VLSI. Since these topics are dealt with in some depth in [110] which describes largely concurrent work on the implementation of a RAM-refreshable pulse stream VLSI RBF demonstrator chip, it was felt unnecessary to duplicate here such discussion in similar detail.

2.2 Radial Basis Function Neural Networks

The RBF neural architecture provides function approximation (or classification) based on a three-layer structure whose data processing behaviour depends on internal parameters of the network. In fact it can be shown that given unlimited network resources any function can be represented [133].

2.2.1 RBF Architecture

The general structure is illustrated in figure 2.1 for a 2:5:1 network (2 inputs, 5 hidden units, 1 output). The input units pass the input data vector directly to each of the hidden units. Each hidden unit responds to an input vector dependent on the distance to the hidden unit *centre* applied to some non-linear function (which may be weighted). The hidden units in figure 2.1 have various centre locations and use a weighted radial Gaussian non-linearity[1].

[1]The radial Gaussian non-linearity is the most popular applied to the RBF but it is not exclusive. Functions such as thin-plate splines, multiquadratic and inverse multiquadratics are also common; it is the nonlinearity which is essential rather than its specific form. A exception might be if the RBF is attempting to fit some known non-linearity. For example, an RBF digital channel equaliser which must interpret data corrupted by random Gaussian channel noise [29] would have an optimal, Bayesian, solution with centres at locations exactly specified by the channel transfer function and with Gaussian non-linearity with the same standard deviation of the noise. A solution with non-Gaussian non-linearities would be possible but less optimal.

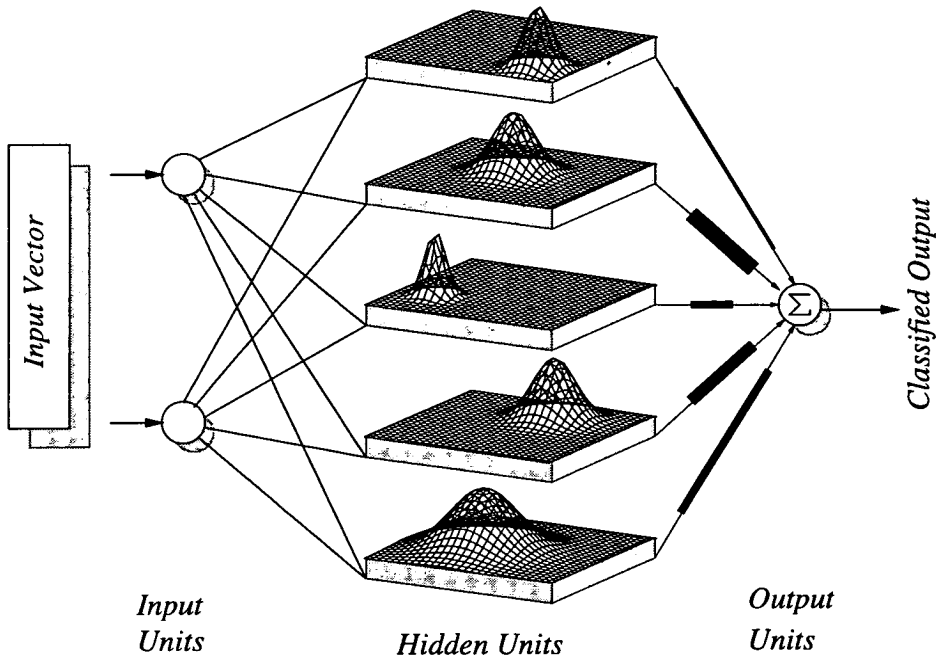


Figure 2.1: Radial basis function neural network architecture.

The weighted summation of the hidden-unit responses is calculated by the output unit or units.

Thus the complete operation of the RBF can be expressed as

$$y_k = \sum_{j=0}^{m} \omega_{jk} \varphi \left(\| \bar{x} - \bar{c}_j \| \right) \tag{2.1}$$

where y_k is the output of output unit k, $\bar{x}$ is the input vector, $\bar{c}_j$ is the centre location of the hidden unit j, $\|$ is the distance metric, $\varphi(.)$ is the non-linearity of the hidden units and ω_{jk} is the weight associated with the connection between hidden unit j and output k. Hidden unit $j = 0$ is usually a bias unit whose output is always unity - this is to allow a non-zero RBF response to an input vector $\bar{x}$ which is outside the receptive field of the hidden units.

The RBF may be used to build a multi-dimensional non-linear model to represent an *underlying function* of the input data, or else a *discriminatory function* used to classify the input data. An RBF classifier therefore expands feature space non-linearly into a higher dimensionality where linear classification techniques may be applied. Notice that there are many variants of the general RBF form shown (see, for example, [73]) but these will not be considered here.

2.2.2 RBF Training

A simple way of setting the hidden-unit centre locations is to populate, densely and uniformly, the whole of input feature space (the multi-dimensional 'space' defined by all valid input vectors). However this is very computationally expensive and impractical for parallel VLSI implementations. To reduce the number of hidden-units a more parsimonious approach is required. This may be by taking some random subset of the training data and using these to define the centres, or by using some form of clustering algorithm such as k-means [116]; here centres become localised about centres of data density. Orthogonal Least Squares (OLS) [28] is an alternative method which involves systematic elimination of hidden units which contribute least to minimising the mapping error of the RBF.

Once the hidden units have been set, the connection weights to the output units can be found by fast linear techniques such as least mean squares (LMS) [172] or the pseudo-inverse matrix technique [14].

Off-line, hidden-layer training and fast linear output layer training make RBFs faster to train than multilayer perceptrons (MLPs) [14] and avoid pathologies of MLP backpropagation or gradient descent training such as local minima. It is also possible to train an RBF with supervised error minimisation techniques; while this may lead to an improved solution over unsupervised training, the RBF training advantages are lost.

2.2.3 RBF Sensor Classifier

In addition to the fast training and avoidance of MLP pathologies, there are other features which make the RBF attractive for the specified aim of classifying and conditioning analogue sensor data:

- Use of a clustering algorithm can lead to clearly identifiable centre locations which have a real-world interpretation which can aid development and debugging.
- Novelty detection is more natural to RBF architectures [103] due to the closed form representation of feature space. If an input vector is far from any network centre, then the incoming data is not similar to any of the training data. This can allow the network to void erroneous reactions to unknown conditions and allow flagging of anomalous data, such as that due to sensor failure.

Some RBF sensor processing applications which are already under development include:

- Real-time detection of differential species in a gas mixture to overcome the poor selectivity of existing surface-acoustic-wave gas sensors [75].

- Remote sensing of water vapour and temperature for weather forecasting [168].
- Object form discrimination from arrays of piezoelectric polymer stress sensors [18].

All of these applications would benefit from a compact low-power implementation. Gas sensors must be suitable for mounting in inaccessible locations and run from small batteries. Remote sensing systems may be based in satellites (small size required to reduce payload launch expenses) or remote self-powered ground station (transmission of raw data back to base would require too much power and excessive receiver-end storage) [2]. And development of the object discrimination work could lead to an artificial tactile system for remote robotics applications where again physical size and power consumption would become critical for autonomous operation in inaccessible environments.

2.2.4 Requirements of a VLSI RBF

Construction of a VLSI RBF requires the implementation of three components

1. Distance metric calculator for hidden units.
2. Non-linearity function of hidden units.
3. Weighted summation function of the output layer (multiply-accumulate operation).

Circuits implementing these functions were being developed concurrently to this work for a RAM-refreshed pulse stream RBF demonstrator chip called PAR [110]. Some competing RBF circuit designs have been described in [110], therefore such discussion has not been replicated here. However, due the central nature of the *pulse-stream* RBF circuits to the work of this thesis it was worthwhile to briefly describe how these functions have been implemented. The non-volatile RBF circuits developed in this project are *not* identical to these circuits. The details of these differences will be discussed in later chapters where floating gate memories are applied.

2.3 PWM Circuit Implementation

Pulse-width Modulation (PWM) calculations are performed by compact PWM-controlled analogue circuits and the resultant *activation voltage* stored on a capacitor (Fig. 2.2).

[2] Watkins *et al* [168] have actually pursued an digital-analogue hybrid RBF sensor chip. Analogue weights were stored on capacitors and refreshed from RAM using D/A conversion. Watkins *et al* admit that this limitation must be addressed for an operational system.

This voltage is pulse-width modulated by a comparator whose other input is a ramp which may define a linear modulation (a), or some form of neural squashing function (b). With the signal thus temporally-encoded, propagation, to a second network layer (whether on or off chip) or to an output, is robust. This technique and implementations are described in more detail in [120].

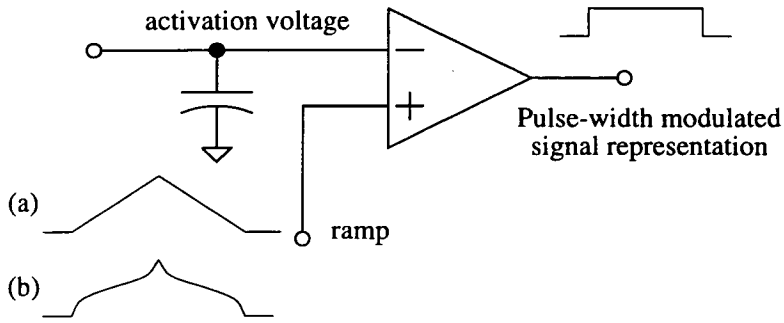


Figure 2.2: Schematic representation of PWM encoding for pulse-stream ANNs.

The use of the pulse-stream approach is especially pertinent for the described sensor data conversion, since with analogue inputs and pulse-coded outputs, the normally expensive analogue-digital conversion is implicit in the neural signal processing. Pulse-stream ANNs may then be considered to form a class of 'intelligent' analogue-digital data-reducing converters naturally bridging analogue sensors and subsequent digital data processing. While pulse-stream computation may not achieve the densities or speed of purely analogue approaches, the value of the above advantages makes it more attractive for such applications.

2.3.1 Euclidean Distance Calculator

The Euclidean distance is a simple geometric metric for calculating the distance between two point vectors, **a** and **b**, in multi-dimensional Euclidean space [3] :

$$distance = \sqrt{\sum_{j=0}^{m} (b_j - a_j)^2} \tag{2.2}$$

where are there $0..m$ dimensions to the Euclidean space and a_j and b_j are the point vector components in dimension j. In classifier terminology the Euclidean space may be called *feature* space where each dimension represents one feature of measurement.

[3]The Euclidean distance metric is the most common and intuitive for the RBF network but other metrics such as the Manhattan may be equally applicable. Here an *approximation* to the exact Euclidean distance is proposed exploiting the MOSFET saturation region characteristics.

The Euclidean distance calculator circuit uses *ratioed pairs* of transistors [125] so called because of the W/L ratio between the constituent transistors of the pair.

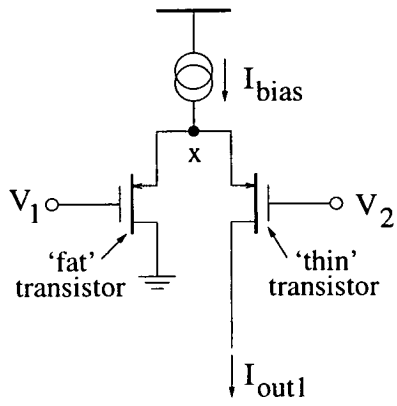


Figure 2.3: Ratioed pair schematic

A simple ratioed pair is shown in figure 2.3. The transistor on the left has a very much larger W/L than the one on the right and they are hence referred to as fat and thin as shown. The high transconductance of the fat transistor means that it can pass up to the whole bias current, I_{bias} with a V_{gs} of only about a threshold. Thus, while both the fat and thin transistors are in saturation, node x is always clamped approximately one threshold voltage above V_1. This means the output current of the pair, I_{out1}, depends on the saturation characteristic of the thin transistor with V_{gs} which is a function of the difference between V_2 and V_1 and is an approximation of one half of the Euclidean distance characteristic (ie. where $V_2 > V_1$).

Mathematical analysis begins by taking the first order saturation region approximation of MOSFET operation:

$$I_{ds} = \frac{\beta}{2}\left(V_{gs} - V_{T_p}\right)^2 \tag{2.3}$$

so then the currents in both transistors ($t1$ = thin, $f1$ = fat) can be estimated:

$$I_{f_1} = \beta_{f_1}\left(V_1 - V_x - V_{T_p}\right) \tag{2.4}$$

$$I_{t_1} = \beta_{t_1}\left(V_2 - V_x - V_{T_p}\right) \tag{2.5}$$

Equation 2.4 can be re-arranged in terms of V_x and substituted into equation 2.5 to

provide an expression for $I_{t_1} \rightarrow I_{out1}$:

$$I_{out_1} = \frac{\beta_{t_1}}{2}\left(V_2 - V_1 + V_{T_p} + \sqrt{\frac{2I_{f_1}}{\beta_{f_1}}} - V_{T_p}\right)^2 \quad (2.6)$$
$$= \frac{\beta_{t_1}}{2}\left(V_2 - V_1 + \sqrt{\frac{2I_{f_1}}{\beta_{f_1}}}\right)^2$$

For the other half of the Euclidean distance characteristic, where $V_1 > V_2$, a complimentary ratioed pair is required where V_1 drives the gate of the *thin* transistor and V_2 the gate of the *fat* transistor. This gives

$$I_{out_2} = \frac{\beta_{t_2}}{2}\left(V_1 - V_2 + \sqrt{\frac{2I_{f_2}}{\beta f_2}}\right)^2 \quad (2.7)$$

Since only one ratioed pair output is non-zero for any V_1,V_2 combination, a complete Euclidean distance approximation can be achieved by Kirchoff summation of I_{out_1} and I_{out_2} to yield $I_{distance}$:

$$I_{distance} = \frac{\beta_t}{2}\left(|V_1 - V_2|\right)^2 + \beta_t\sqrt{\frac{2I_f}{\beta_f}}\left(|V_1 - V_2|\right) \quad (2.8)$$

where $I_{distance}$ is dominated by the correct formulation immediately to the right of the equality. This circuit was initially proposed in [35][4]. (See figure 6.8).

2.3.2 Non-Linearity Calculation

As mentioned in section 2.2.1, the non-linearity function of the hidden-units can be a Gaussian or some other function which adequately represents a closed form region within feature space about the defined centre location. In [35] the distance current was used to linearly discharge a capacitor and thus compute an activation *voltage* proportional to the distance calculation. Similar to the scheme described in section 2.3, this activation voltage could be transformed into a pulse width. Using a non-linear ramp such as a Gaussian 'on its side' simultaneously implements computation of the non-linearity function. This implementation, however, had several disadvantages: (*i*) Requirement for large area activation capacitor, (*ii*) Finite evaluation time to allow

[4]The original published incarnation used n-channel rather than p-channel transistors. The p-channel version is actually preferred since the lower p-channel conductivity provides a wider dynamic signal range for a given I_{bias}

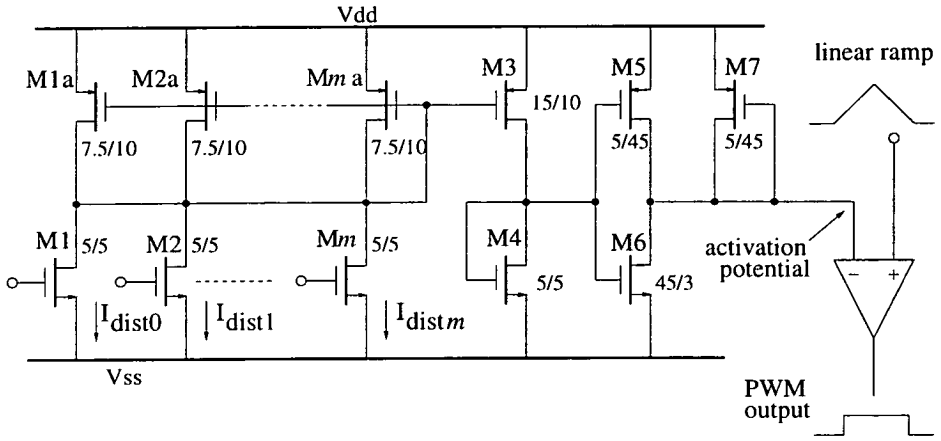
discharge of capacitor, and the need for additional circuitry to control this period, and (*iii*) Dynamic voltage storage.

Therefore, instead of directly copying the implementation in [35], some consideration was given to replacing this circuit with one which uses some inherent non-linearity of MOSFET operation. A physical non-linearity of natural transistor characteristics would seem as potentially valid as a strictly mathematical one and also easier to implement. Since there is currently no accepted standard method to determine the standard deviation of a Gaussian non-linearity, actually choosing the optimal 'width' of an arbitrary non-linearity presents no additional difficulties.

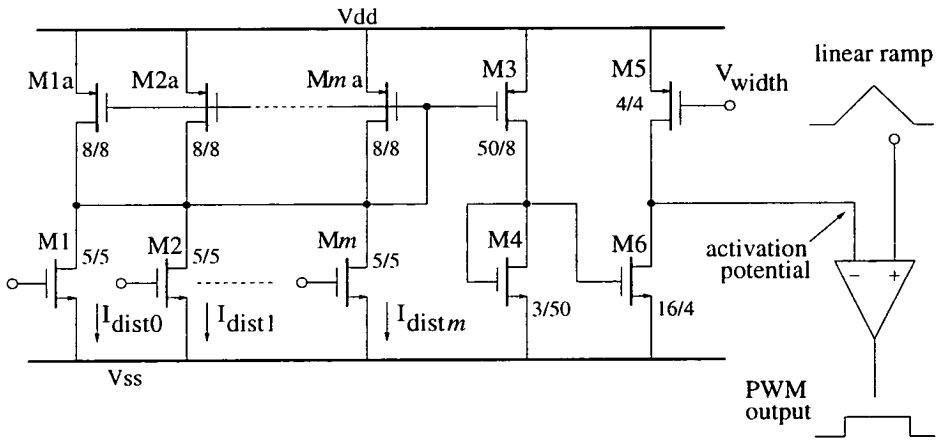
These designs are described in this section. The design actually used on the test chip described later was not that considered the best, but the reason for its selection will be described shortly.

A circuit proposed by the author is shown in figure 2.4(a). Transistors M1 to Mm mirror the currents from the Euclidean distance circuits of each dimension of the network. The Kirchoff summation of these currents flows through M1a to Mma which mirrors this to M3. M4 is a diode connected load which maps this current into a voltage which drives the nonlinearity circuit of M5-M7. This circuit comprises an inverter with a transition curve which is smoothed by the feedback action of the diode-connected load M7. The width of the centre in this circuit is controlled by transistor sizes. Whilst it is possible for the RBF to use a fixed-width non-linearity, in the case of limited network resources, the ability to modify the non-linearity width adds an extra degree of freedom which, depending upon the data being modelled, may much improve the classification response. The simple modification of removing the pull-up transistor of the inverter and the diode-connected load and replacing these with a single p-channel transistor driven by a V_{width} parameter (ie. a current-sink n-type inverter) is shown in figure 2.4(b). It was found that an effective adjustable non-linearity could be produced as shown in figure 2.5.

Mayes adapted the diode-connected load idea in the circuit of figure 2.4(a) to make it operate directly on the distance current and to have an electrically modifiable width. One diode-connected load and one pull-up width transistor is distributed for each dimension as shown in figure 2.6. These non-linear resistances are thus connected in parallel and so scale with the total distance current. Thus a non-linear transresistance implements the RBF non-linearity function. The width transistor M1a,M2a,..,Mma modulated the non-linear pull-up resistance under the control of V_{width} such that a family of non-linear transfer curves could be produced (see figure 2.7).



(a)



(b)

Figure 2.4: (a) Summation and feedback-inverter non-linearity circuit, (b) Summation and current-sink n-type inverter non-linearity circuit.

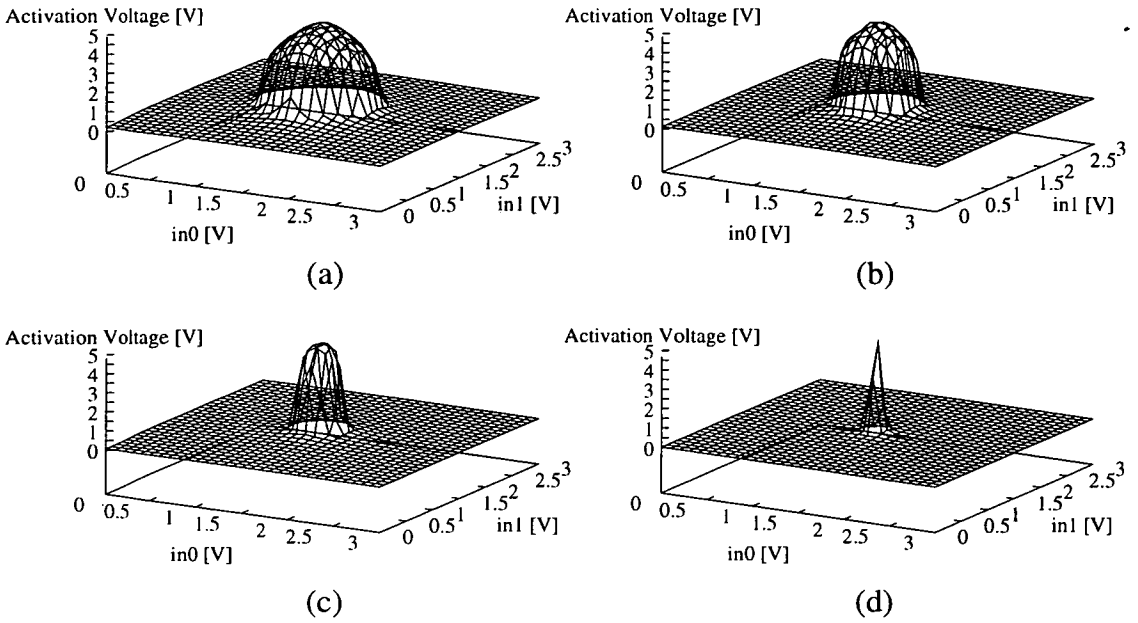


Figure 2.5: Simulated distance circuit and current sink n-type inverter non-linearity response in two dimensions with centre located at (1.5V,1.5V) for (a) V_{width} = 0V, (b) V_{width} = 1V, (c) V_{width} = 2V and (d) V_{width} = 3V.

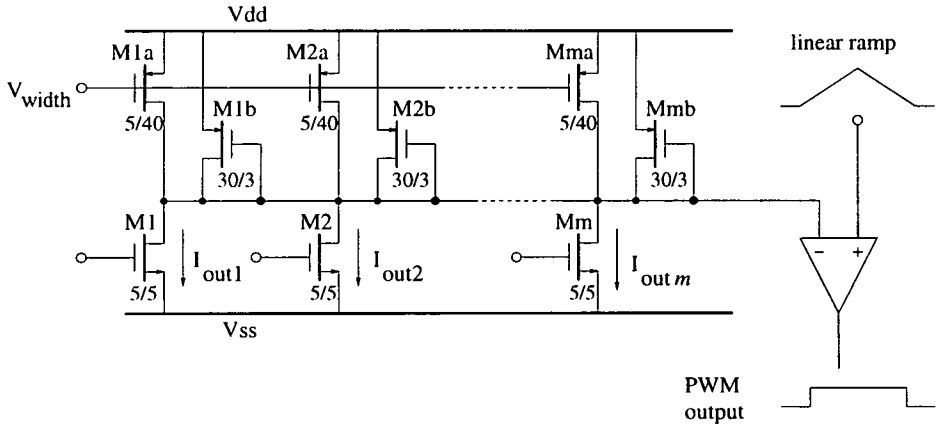


Figure 2.6: Circuit schematic of distributed load non-linearity

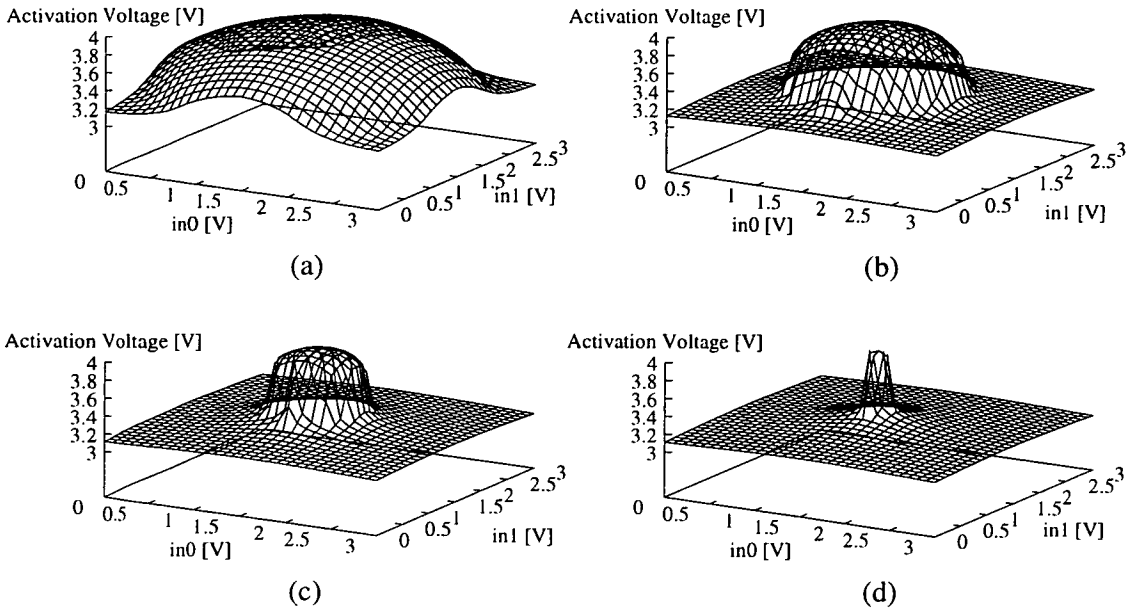


Figure 2.7: Simulated distance circuit and distributed transresistance load non-linearity response in two dimensions with centre located at (1.5V,1.5V) for (a) V_{width} = 2.0V, (b) V_{width} = 2.5V, (c) V_{width} = 2.75V and (d) V_{width} = 3.0V.

Compensation Circuit

Unfortunately a problem exists with the new non-linearity circuit when large values of V_{centre} and V_{width} are combined. As seen in figure 2.8(a) the non-linearity never peaks at the centre value. The reason for this behaviour will be described briefly.

Expanding equations 2.6 and 2.7 gives

$$I_{out_1} = \frac{\beta_{t_1}}{2}(V_2 - V_1)^2 - \beta_{t_1}\sqrt{\frac{2I_{f_1}}{\beta_{f_1}}}(V_2 - V_1) + \frac{\beta_{t_1}}{\beta_{f_1}}I_{f_1} \tag{2.9}$$

$$I_{out_2} = \frac{\beta_{t_2}}{2}(V_1 - V_2)^2 - \beta_{t_2}\sqrt{\frac{2I_{f_2}}{\beta_{f_2}}}(V_1 - V_2) + \frac{\beta_{t_2}}{\beta_{f_2}}I_{f_2} \tag{2.10}$$

Therefore the output, I_{dist}, at $V_1 = V_2$ is

$$I_{distance} = \frac{\beta_{t_1}}{\beta_{f_1}}I_{f_1} + \frac{\beta_{t_2}}{\beta_{f_2}}I_{f_2} \tag{2.11}$$

This current provides an inherent offset, such that the I_{dist} is not truly zero when $V_{in} = V_{centre}$ as it should be. However I_{dist} is minimal at this point so it is not generally a problem. But due to the *body effect* changing channel conductivity, this current is

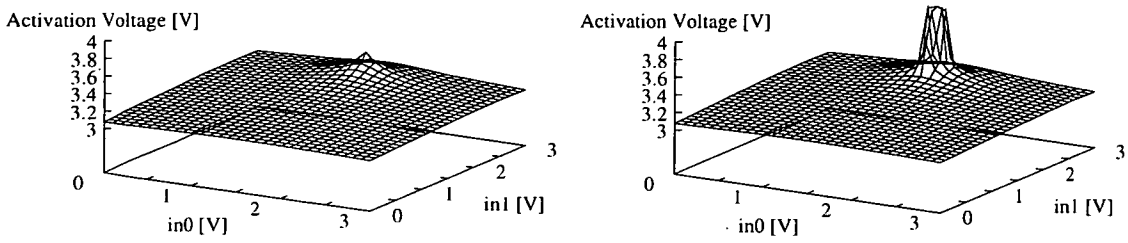


Figure 2.8: Simulated response of Euclidean distance circuit and distributed load non-linearity for centre at (1.5V,2.75V) with V_{width}=3V. (a) without compensation circuit, (b) with compensation circuit.

dependent upon V_{centre} and can rise to several tens of nanoamps [5]. When V_{width} is large, the conductivity of the pull-up transistors in the non-linearity circuit is low, this offset current can then be so high as to prevent the non-linearity peaking at the centre location as seen in figure 2.8(a). Because it is important to allow very narrow centres to define localised receptive fields, it is not appropriate to simply limit V_{width}. Therefore some means of eliminating the offset is required.

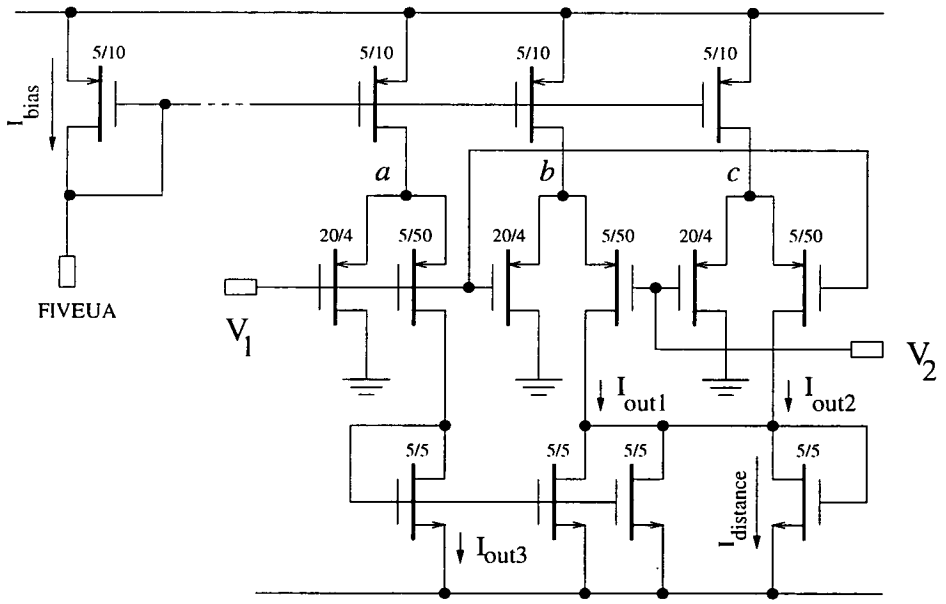


Figure 2.9: Circuit schematic of Euclidean distance calculator cell

Mayes designed a *compensation circuit* which was added to the basic Euclidean

[5]Mayes attempted to annul the body effect by use of separate wells such to force V_{bs} = 0V but found this solution to be too area intensive.

distance calculator circuit. This comprised a third ratioed pair which had both gate terminals connected to V_{centre} or V_{in}. This compensation pair then outputs only the offset current, which can be doubled and removed from I_{dist} by Kirchoff summation. This is shown in figure 2.9 where the ratioed pair connected at a is the compensation pair. The current mirrors are used to remove twice the bias current from I_{dist}.

To ensure that the distance response is symmetrical and intersects the x-axis at $V_1 = V_2$ requires

$$\frac{\beta_{t_1}}{\beta_{f_1}} I_{f_1} = \frac{\beta_{t_2}}{\beta f_2} I_{f_2} = \frac{\beta_{t_3}}{\beta_{f_3}} I_{f_3} \tag{2.12}$$

In fact the original current-sink n-type inverter non-linearity (figure 2.5) would appear to offer a number of advantages over the modified distributed transresistance load version (figure 2.7):

- There is no need for the extra complexity and *area* of the compensation circuit since the non-linearity works in voltage rather than current mode.
- The original circuit has a wider dynamic range both of V_{width} and activation potential.
- The original circuit allows for smaller receptive fields and these are preferable since data may cover small and intricate regions of non-linear feature space.

However, fabrication and evaluation of this alternative design would require additional effort somewhat removed from the principal aim of adding non-volatility to the RBF. Since Mayes' circuit was already fabricated and tested and had been used with some success for classification it was decided to use this, and the compensated distance cell for the purposes of this project.

2.3.3 Multiplier

The final functional element required for the RBF is a linear multiplier-accumulator for the output layer. Multiply-accumulate is also a fundamental operation of the MLP neural network and so this functionality has already been the subject of significant previous study within the Edinburgh neural group, culminating in the $EPSILON$ [33, 34, 71] and $EPSILON2$ [87, 86] pulse-stream MLP chips.

The $EPSILON$ multipliers used two transistors biased in their linear region of operation. Churcher [36] proposed an alternative design using two transistors biased in saturation. Although this circuit was not implemented by Churcher for funding reasons, a version incorporating amorphous silicon (a-Si:h) resistors was successfully fabricated by Holmes [80]. He observed that the circuit was simpler, required much

fewer bias voltages and currents than the *EPSILON* design and that they were less sensitive to instability. Therefore this design was selected for the multiplication operation.

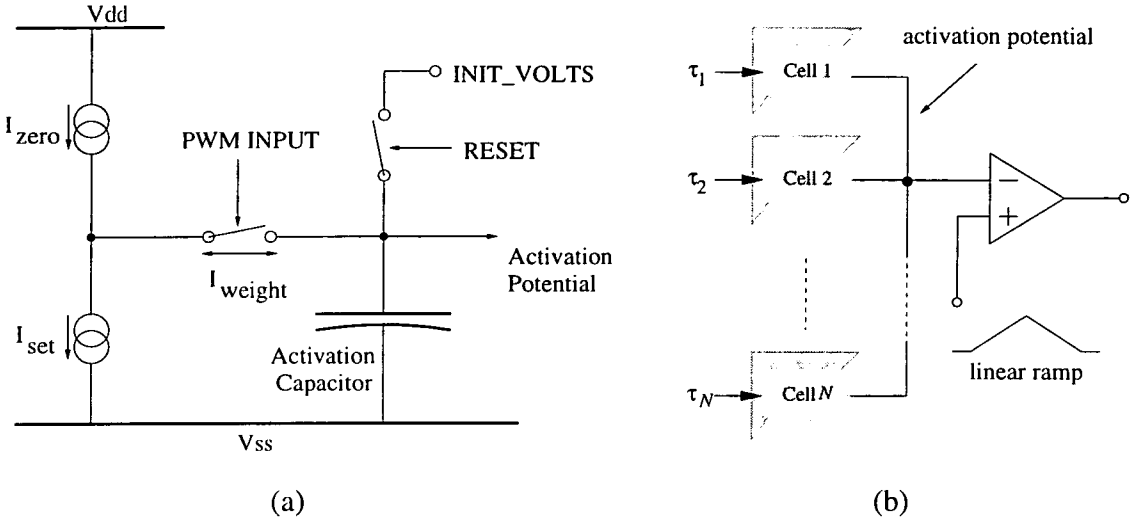


Figure 2.10: (a) Schematic of multiplier circuits, (b) Cascading of multiplier to build multiply accumulate circuit

The operation of the multiplier is illustrated schematically in figure 2.10(a). I_{set} may be higher or lower than I_{zero} such that when the PWM-modulated input closes the switch connected to the `sum` node, I_{weight} is established equal to the difference between the currents. I_{weight} may linearly charge the activation capacitor (if $I_{set} < I_{zero}$) or linearly discharge the activation capacitor (if $I_{set} > I_{zero}$) – the capacitor has already been preset to some intermediary voltage by pulsing of the `reset` switch. Thus the activation potential, V_{act} is

$$V_{act} = V_{INIT\_VOLTS} + \frac{I_{weight} \cdot t_{PWM-input}}{C_{act}} \tag{2.13}$$

where $t_{PWM-input}$ is the duration of the PWM-modulated input (ie. the length of time the switch stays closed) and C_{act} is the activation capacitance. Thus a 2-quadrant multiplication is implemented; V_{INIT_VOLTS} represents an offset zero point.

This circuit is cascadable as shown in figure 2.10(b); the activation capacitance is distributed across all cells. Therefore the activation voltage at the input terminal to the comparator becomes

$$V_{act} = V_{INIT\_VOLTS} + \frac{1}{N \cdot C_{act}} \sum_{j=1}^{N} I_{weight(j)} \cdot \tau_j \tag{2.14}$$

This then implements the multiply-accumulate function (with implicit normalisation to keep the voltage within range) required by the RBF (where $\tau_1..\tau_N$ are the PWM-coded outputs from the N hidden units of the RBF).

The weight, ω of each multiplier is determined by I_{set}. Churcher proposed using a differential stage to implement an approximately linear mapping from the voltage stored on a RAM-refreshed capacitor; Holmes used a current mirror to copy the current in an a-Si:h resistor; and Mayes used *dynamic* current mirrors to set up I_{set} *and* I_{zero} [111].

2.4 Summary

This chapter has briefly described the components required for RBF operation: a multi-dimensional distance calculator, a non-linear mapping function and multi-dimensional multiplier-accumulator. The implementation of these functions in pulse-stream VLSI has then been briefly described. These circuits will be considered further in chapter 6 when arrays of RBF subcircuits are constructed and discussed.

Chapter 3

Review of Long-Term Synaptic Weight Storage

3.1 Introduction

This chapter contains a brief review of various technologies and techniques used for non-volatile weight storage in analogue VLSI ANNs. This serves to provide a context for the work of this thesis, describing in particular the operation of standard CMOS floating gates, showing how they relate to the many other approaches of providing non-volatility in analogue VLSI ANNs but, it is argued, the most appropriate technology for providing the requirements detailed in chapter 1.

3.2 Properties of Non-Volatile Technologies

The simplest and most easily programmable analogue VLSI memory, as opposed to parameters hardwired by device geometries[1], is the storage of charge on poly-poly or gate oxide capacitors. Here

$$V_{weight} = \frac{Q_{stored}}{C} \tag{3.1}$$

Unfortunately, since a pass transistor must be provided to initialise the level of charge on the storage capacitor, the stored charge will leak due to the reverse-biased junction diode and subthreshold conduction of the pass transistor as shown in figure 3.1. V_{weight} then degenerates over time. The acceptable hold time depends on the required precision but is typically of the order of milliseconds.

Improvement on this hold time is not simple; convenient non-volatile analogue memory is one of the significant shortcomings of VLSI technology. However, there are many approaches to this problem - both circuit designs and special devices - which can be categorised according to a number of features such as those listed in table 3.1.

[1] Hardwiring of weights is undesirable because it precludes re-training of the network, even to trim hardware offsets.

Programmed Parameter	Does the memory involve the reprogramming of a resistance element, a voltage, a transistor threshold, etc.?
Power Dependency	Does the memory require a continual application of power to retain its value?
Retention Time	How long is the stored weight non-volatile and how does it eventually decay?
Precision	How precisely can the weight be set and what happens to the precision over time. eg. exact hold, gradual decay, abrupt decay?
Endurance	How many times can the memory be reprogrammed before the device fails? This is particularly important for chip-in-the-loop training of ANNs: one-time programmability is insufficient here.
Failure Behaviour	How does the memory fail? eg. Does it become unprogrammable, or only programmable over a limited range. Does the existing weight value collapse?
Yield	How many inoperational memories can be expected to be fabricated? ANNs can often tolerate a few unprogrammable weights but only up to a point.
CMOS Compatibility	Does the memory require a special fabrication process, post-processing of a standard CMOS process or entirely compatible with standard CMOS without further processing?
Programming Speed	How fast can the memories be programmed to target weights? Again this impacts upon the practicality of chip-in-the-loop training.
Programming Linearity	Non-linear programming substantially complicates the programming procedure and may also slow it down.
Read/Write Synchronicity	Can weights be written to whilst they are being (correctly) read? Important for continuously adaptive systems or non-linear programming with feedback.
Cell Size / Power	Standard VLSI circuit requirements: must not be too large or draw too much power.

Table 3.1: Selection of criteria for evaluation of a non-volatile analogue memory technology

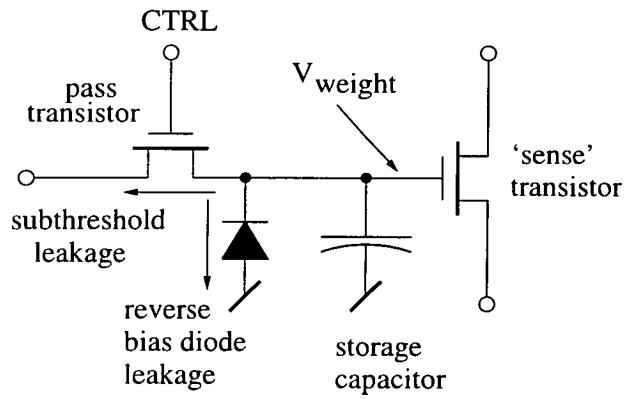


Figure 3.1: Analogue weight storage on capacitor with charge leakage paths

One issue which may be of contention is *power-up non-volatility*. This may seem an oxymoron using the traditional definition of non-volatility: retention of data in the absence of power. However it is argued here that while a strict partitioning of volatile and non-volatile digital technologies is reasonable since there is an implicit storage of data in digital logic circuits (using basic elements such as flip-flops, registers and static RAM cells) so system states can exist indefinitely in the presence of power. The same is not true of analogue parameters in an analogue circuit - data held on switched capacitors degrades *immediately* even in the presence of power. Retention of analogue data even in the presence of power then becomes problematic and may aptly be labelled a problem of analogue volatility. Therefore it has been decided to categorise analogue memories into three classes:

1. **Volatile:** Charge storage on a (leaky) switched capacitor.
2. **Power-Up Non-Volatile:** Storage of analogue parameter in the presence of continuous power.
3. **Power-Down Non-Volatile:** Storage of analogue parameter in the absence of continuous power.

Volatile capacitor methods have already been mentioned. Memories of the other two classes will be described in the following two sections.

3.3 Power-Up Non-Volatile Technologies

Most *circuit-based* approaches to non-volatility directly tackle the problem of charge leakage from capacitors. This requires either *compensation* or an attempt to limit the leakage, or alternatively a refreshing or regeneration of the weight value.

3.3.1 Leakage Limitation and Compensation

The hold time of the capacitor can be improved by limiting the leakage current or compensating for its effect. Two proposed designs are:

- **Improved sample-and-hold:** The circuit of figure 3.1 is a basic sample-and-hold; Vittoz *et al* [166] proposes the use of a two amplifier sample-and-hold which attempts to ensure zero volts across both the leakage paths of junction diode and channel. Because of inevitable imperfections leakage will still occur, but hold times can be increased to a number of seconds.

- **Differential capacitor:** Instead of a single capacitor, V_{weight} is stored as the difference in voltages between two capacitors, V_{diff}. Since both capacitors decay in a similar way, the decay in V_{weight} is less severe. (Horio *et al* [82] propose periodically refreshing the memory by resetting the voltage on one capacitor to $V_{diff} + V_{ref}$ and the other to V_{ref}. However sensing error of V_{diff} gradually degrades V_{weight}).

Increasing hold times beyond more than a few seconds (or possibly minutes if run at low temperatures) requires a more explicit approach to tackling capacitor charge leakage. This requires some form of weight refreshing or weight regeneration.

3.3.2 Weight Refresh and Regeneration

Several approaches can be taken to weight refresh or regeneration. Most are general techniques for storing analogue values in VLSI, others are distinctly 'neural' and rely on continuous learning in the ANN. These may be categorised as follows

- Local digital storage
- Global refresh
- Local refresh
- Continuously adaptive circuits

3.3.3 Local digital storage

Digital SRAM cells use two loop-connected inverters to ensure that the binary value will not decay. Values held locally at neural cells may be used to set weights through some form of simple analogue-digital conversion.

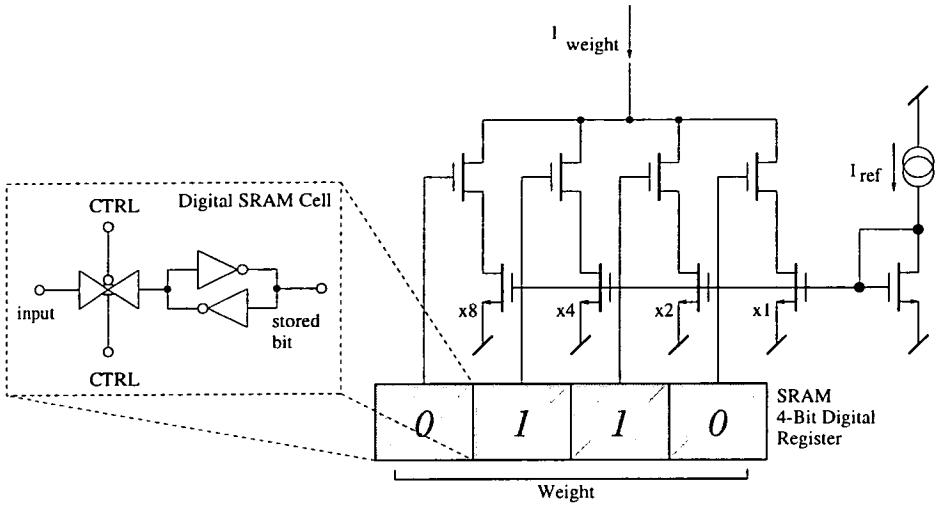


Figure 3.2: Multiplying (Current) Digital-Analogue Converter with SRAM cell inset

A simple example of such system is shown in figure 3.2. This employs a multiplying digital-analogue converter (MDAC) which multiplies the digitally held weight into an analogue current, I_{weight} according to

$$I_{weight} = \sum_{j=0}^{N-1} 2^j I_{ref} d_j \tag{3.2}$$

where d_j is the digital bit held in register cell j and N is the register length. Analogue VLSI ANN applications of this memory are described in [79, 85, 158]. An early pulse-stream ANN used a similar technique where a digital register was used to gate pulse trains [118].

Although this technique could be expanded to any reasonable required precision, its principal disadvantage is the large area required.

3.3.4 Global Refresh

An alternative technique is to lump all the digital storage together (often in an external RAM chip but it could be on-chip). Weights are now stored as analogue voltages on capacitors but leakage is handled by a *refresh cycle*. The RAM is clocked through and each weight is digital-to-analogue converted and (re-)stored on the relevant capacitor. Provided that the refresh cycle is fast enough, intermediate degeneration of weights is not sufficient to adversely affect the ANN operation. This may be a problem for large networks where many weights must be refreshed; here the refresh rate and/or the storage capacitor size must be increased. This technique is also particularly power con-

suming (particularly in the D/A conversion), requires a large area, especially when the interconnect is considered and the fast analogue busses can be a considerable source of noise. However the concept is simple and it is very easy to change weights, making this technique one of the most popular for analogue VLSI ANNs, including those designed in the Edinburgh neural group.

3.3.5 Local Regeneration

An alternative approach which avoids the problems associated with high speed refresh busses is to regenerate the analogue weights *locally*. This may be done using *(i)* staircase regeneration, or *(ii)* with an A/D/A loop.

Staircase Regeneration

A possible circuit proposed by Vittoz *et al* [166] is shown in figure 3.3. Globally distributed V_{stair} is continually ramped up in quantised steps as shown. When $V_{stair} > V_{weight}$, the switch closes and V_{weight} is reset to the nearest quantization level above its current value. After this, the logic ensures that the switch remains open until the system resets with the return to zero of V_{stair}. The period of V_{stair} must be less than the time it takes for V_{weight} to decay more than a quantization level[2]. The use of a low leakage sample-and-hold may help. This approach can be scaled to any size of network since no cycling through of RAM addresses is required. Castello *et al* [22] and Hochet [76] describe similar approaches (Hochet using a linear ramp instead of a staircase).

A/D/A Loop

Another local regeneration scheme involves an iterative procedure of quantization and weight refresh using an A/D/A loop (analogue-digital-analogue conversion) to restore the analogue voltage on a capacitor. The quantization rounds up to allow for intermediate leakage from the capacitor [16, 104]. Cauwenberghs and Yariv [25] adapt this scheme to use an *incremental* refresh rather than overwriting the stored V_{weight} such to avoid unrecoverable quantization errors. Variants on this design include a current-mode approach (using a current latch [45]) and a frequency-mode approach (with a voltage controlled oscillator (VCO) and frequency locked loop (FLL) [82]).

While no comparison exists in the literature, it would seem likely that global refresh techniques using data fixed digitally in RAM would offer greater robustness than

[2]It must be ensured that V_{weight} will *decay* (fall towards zero) rather than *degenerate* (diverge randomly from its initial value).

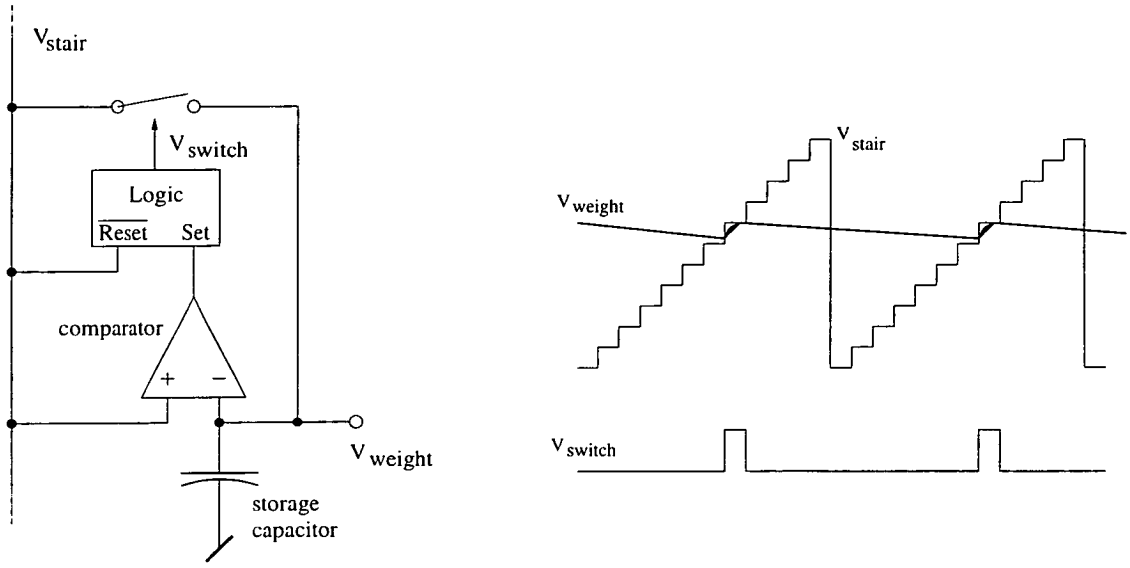


Figure 3.3: Staircase refresh circuit

local regeneration where the data representation is transient and for much of the time exists solely in an analogue representation which may be prone to gradual or abrupt corruption.

Unfortunately these schemes are also area intensive and are susceptible to noise which may flip V_{weight} irrecoverably to the wrong quantization level.

3.3.6 Continually Adaptive Circuits

The preceding circuits are appropriate, not only to ANNs, but wherever an analogue memory is required in VLSI. But in ANN-specific circuits which incorporate *on-chip learning* (OCL), V_{weight} may be continually updated by the learning system such that it need not be held constant. V_{weight} is held on a capacitor and may be incremented or decremented by learning circuits.

V_{weight} can be maintained in three ways:

1. With the continuous application of external stimulus such that learning is always on-going (weight decay in the absence of stimulus, but recovery upon the reintroduction of stimulus) [56, 101, 102, 148].

2. With sufficiently rapid learning, weights can be refreshed by periodically relearning from a fixed training set between applications of novel data [5, 20].

3. After the learning phase V_{weight} is copied into a storage cell such as an A/D/A loop [24]. Montalvo *et al* have proposed - although not implemented - the use of

floating gates (see section 3.5) as secondary non-volatile storage in an on-chip learning capacitor weight system [113].

On-chip learning is a difficult problem, demanding complicated designs. The high precision needed for back-propagation-type algorithms [17, 101, 102, 178] requires that less-mathematical, more technology-led designs are necessary. These issues are also important in the reduction of circuit size and complexity.

3.3.7 Battery-Backed RAM

Dallas Semiconductor and Benchmarq have introduced non-volatile SRAM chips (batRAMs) which contain a lithium battery which provides sufficient power for over ~ 5 years data retention when the external supply is removed. These comprise SRAM, battery and controller on a small PCB which is epoxied into the packaging. Sizes of up to 4Mbits [40] are available. Although the cost is higher than for a comparably sized EEPROM (see section 3.5.4), the SRAM advantages of fast write time and unlimited reprogramming are retained.

It is possible to use battery-backed RAM in global refresh ANN weight storage to achieve a form of pseudo-power-down non-volatility. However, this system is not fully integrated as desired and requires a rather large battery which must be protected from soldering by complex packaging. It is not suitable for local regeneration techniques because of their higher power demands. Unlike EEPROMs, batRAMs cannot be reprogrammed after their rated retention period but must be discarded; it is also worth noting that lithium is a well recognised environmental toxin [41].

A few research groups are developing solid-state thin-film rechargeable batteries using various lithium compounds as electrodes. Layers of these batteries are deposited using techniques compatible with the eventual integration with circuits on VLSI wafers although this does not yet appear to have occurred. A typical thin-film battery area is of the order of a square centimetre and so is comparable with large microprocessor dies. A recent lithium-ion thin-film battery provides ~ 100 milliamp-hours at 3.6V [2] (achieving a higher energy density than lithium coin batteries for a number of technological reasons). In the near future there is then the potential for circuits with on-chip power supplies (although at present manufacturability is a problem)[3]. These batteries might drive miniature ultra-low power circuits or provide pseudo-power-down retention using very low power RAM cells. The benefit of integrating the power source on-chip is the packaging size reduction and greater protection for fragile battery contacts.

[3]This paragraph is based on a correspondence with Te-Yang Liu, a research associate at the Electro-Optics Technology Center of Tufts University, USA.

3.4 Power-Down Non-Volatile Memories

All the previously described memories are fully compatible with SLV-CMOS since they are simply circuit designs. This means that they also should have high endurance and yield. However all require some input of power to maintain their non-volatility. To minimise power consumption and battery size, as little power as possible must be devoted to maintaining the weights and certain applications may even permit long shut-down periods to preserve battery charge. This is particularly important if it is impractical, expensive or disruptive to have to replace the batteries (such as in a remote sensing station) and re-establish lost weight values. It may also not be convenient to maintain power between establishing ANN weights and fixing it in operation position (ie. shipping requirements). Furthermore, sensors may need to be removed from power sources and carried to new locations. In all these cases it would be required that the system would start up with the same weight set regardless of disruption to power.

Another consideration is power consumption *during* operation. By definition power-down non-volatile memories require zero external power to *preserve* their value although all will require some power to actually *sense* it during operation (eg. current in a resistor). However the support circuitry for power-down non-volatile memories is generally far simpler than that of a power-up non-volatile memory cell leading to a considerable power saving during power-up operation.

A final consideration is the reduction in switching noise and system complexity if the need for continually operating refresh circuits is removed.

Until recently, establishing weights by setting geometries of circuit elements at fabrication time was the only way of achieving this in a standard CMOS process. But this is extremely inflexible, precluding even trimming of analogue offsets let alone customising weight sets and subject to diverse behaviours due to fabrication mismatch (laser-trimming of on-chip resistors could help but this is inconvenient, non-standard and expensive).

Therefore non-volatile technologies were used which either required special fabrication processes or post-processing of standard CMOS wafers. Some such technologies include

1. **Silicon Nitride:** MNOS (metal-nitride-oxide-silicon) technology and its descendent, SONOS (silicon-oxide-nitride-oxide-silicon) – which has better scalability with shrinking process geometries, lower control voltages, better retention and endurance [171] – store charge in discrete traps in a silicon nitride layer above a transistor channel, thus modifying the transistor threshold as viewed from an upper control gate (a similar principle to floating gates described in section 3.5).

2. **Ferroelectrics:** a ferroelectric film has a highly non-linear dielectric composed of irregular *domains* which are polarised by applied electric fields. These may be used as capacitors in ferroelectric DRAM cells which retain their charge when power is removed. Polarisation is not abrupt, and intermediate characteristics can be exploited in an analogue manner [37].

3. **Programmable Resistors:** modifiable resistors (memistors) have been constructed for ANNs using *(i)* tungsten oxide, *(ii)* bismuth oxide, and *(iii)* copper sulphate electrolytes (electrochemical synapse).

4. **Amorphous Silicon:** amorphous silicon deposited between two metal electrodes forms an electrically programmable resistor due to a narrow conducting filament. The resistance may be varied bidirectionally over a range determined by the electrode metal by the application of positive or negative pulses following a high voltage *forming pulse*.

More detailed description of these special technologies is given in [80] and [119] so will not be replicated here. None of these memory devices was considered suitable for the applications described in chapter 1 due to the need for special fabrication processes, or at least post-processing of a SLV-CMOS wafer.

Floating gate memories offer power-down non-volatility but also require special fabrication processes. However, as will be shown in chapter 4 these have the potential to be fabricated, although less ideally, in SLV-CMOS which makes them the most promising devices to meet the specification of chapter 1. Before describing *analogue, SLV-CMOS* floating gates, their direct antecedents, *digital, special process* floating gates will be described here.

3.5 Floating Gate Devices

A floating gate is an electrically isolated capacitance directly above the channel of a 'sense' transistor. The generic form of a floating gate memory cell is shown in figure 3.4. The stored floating gate charge modulates the threshold voltage of the sense transistor as seen from the control gate by $dV_t \sim Q_{fg}/C_{fg}$ where Q_{fg} is the charge stored on the floating gate.

Programming is accomplished by electron injection onto the floating gate through the insulating oxide and several mechanisms are suitable for this, although it is also necessary to provide a suitable means for removing the electrons, so that the device

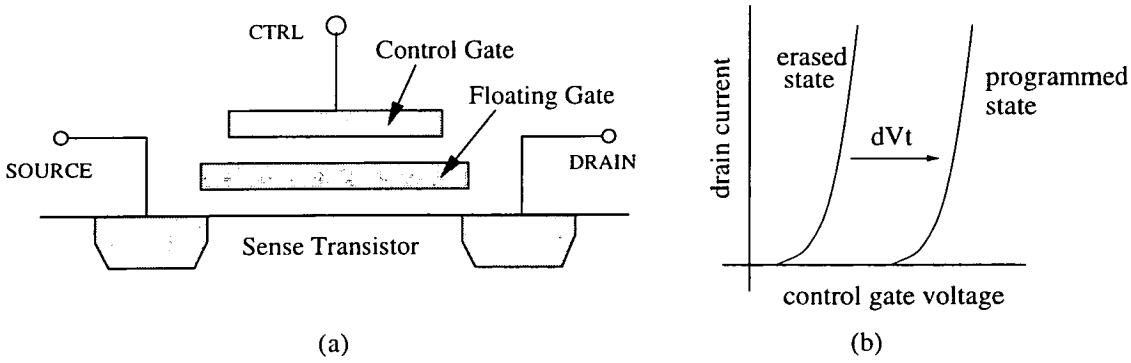


Figure 3.4: Generic floating gate memory cell: (a) Cross-section, (b) Memory action

can be initialised or re-programmed[4].

Physical processes which can inject (and, in some cases, remove) electrons onto the floating gate will be discussed in the following order:

- Ultraviolet Irradiation
- Hot Electron Injection
- Avalanche Injection
- Fowler-Nordheim Tunnelling

3.5.1 Ultraviolet (UV) Illumination

Ultraviolet light has a frequency of petaHertz (10^{15} Hz) to exaHertz (10^{18} Hz) [186]; typical mercury UV lamps emit light at 1.182×10^{15} Hz. This means that bombardment of electrons with UV photons adds sufficient energy that they can surmount the potential barrier imposed by the Si-SiO_2 boundary. This effectively means that the normally insulating oxide becomes conductive although only slightly: it still has a resistance measurable in GΩs, so programming is slow (seconds or minutes). Controlled programming of the floating gate then simply requires that circuit nodes are appropriately biased so that the desired voltage is established on the floating gate as shown in

[4]A "programmed" floating gate can be defined variously as one with a negative charge or one with a positive charge. No consistent rule as been adopted in the literature or amongst semiconductor manufacturers. Therefore in this thesis, the terms WRITE and ERASE have been avoided where they might cause confusion (ie. where the direction of electron transport is relevant); the more direct terms, electron INJECTION (onto the floating gate) and electron REMOVAL (from the floating gate), have been used in their place (regardless of the physical mechanism used to achieve this). The term "programming" has been taken to have the more general meaning of INJECTION *or* REMOVAL.

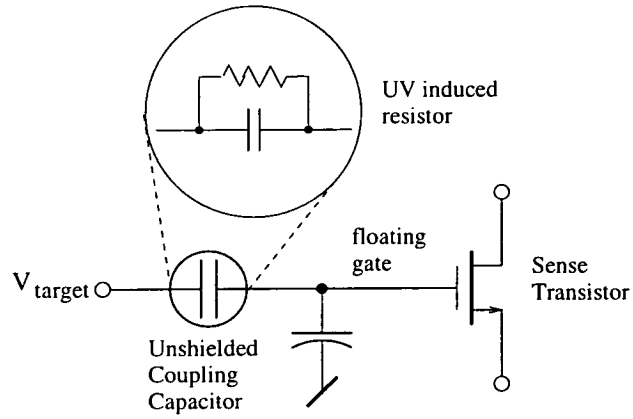


Figure 3.5: UV programming of floating gate

figure 3.5 . Normally *shielding* is required such that only the floating gate insulating oxide is exposed to UV rather than the oxide separating control and signal lines.

Åbusland and Lande [1] propose a scheme for UV programming of analogue voltage references.

3.5.2 Fowler-Nordheim Tunnelling (FNT)

Fowler and Nordheim described the emission of electrons in a high electric field as early as 1928 [61]. It is a quantum mechanical phenomenon described by Schrödinger's Wave Equation in terms of a variable ψ which is known as the *probability amplitude* or *the wave function* of an electron. $\mid \psi \mid^2$ represents the probability of finding an electron at a given location.

Schrödinger's equation states that

$$\nabla^2\psi + \frac{8\pi m}{h}\left(E_{total} - E_{potential}\right)\psi = 0 \tag{3.3}$$

where m is the mass of the electron, h is Planck's constant and E refers to the energy of the electron.

This yields a description of an electron within the *potential well* imposed by the boundary conditions of MOS capacitor structure. Here the vertical barrier height is typically taken to be around $\phi_B \simeq 3$eV (defined as the bottom of the silicon conduction band to the bottom of the oxide conduction band).

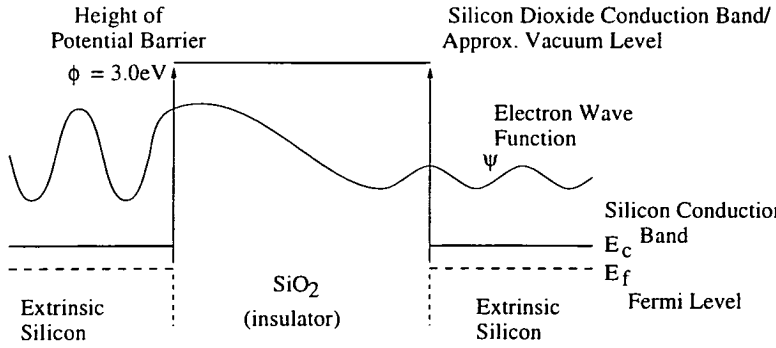


Figure 3.6: Potential barrier due to silicon dioxide potential barrier neglecting band bending

The solution of Schrödinger's Equation takes the form:

$$\psi = A\exp\left(\frac{i\sqrt{2m\left(E_{total}-E_{potential}\right)}\,x}{\hbar}\right) + B\exp\left(\frac{-i\sqrt{2m\left(E_{total}-E_{potential}\right)}\,x}{\hbar}\right) \tag{3.4}$$

where x is the distance into the potential barrier and $\hbar = h/2\pi$.

Hence, solving for a typical potential barrier situation such as that shown in figure 3.6, and substituting in the boundary conditions results in the remarkable conclusion that although the wave function decays exponentially through the barrier, it is still finite at the other end. Hence electrons have a small, but finite probability of appearing on the opposite side of a barrier of potential energy which is actually larger than their own total energy.

In a MOS capacitor structure, such as a MOSFET cross-section, when a bias is applied to the gate, the shape of the potential barrier is distorted due to the induced electric field. This is illustrated in figure 3.7 which also shows the band bending at the interface with the p-type silicon substrate.

Figure 3.7 illustrates how the tunnelling distance is narrowed with applied electric field: with no field then it is approximately the entire width of the oxide, but *with a field applied, electrons have to tunnel a much shorter distance before they enter the conduction band of the SiO_2 and are swept along by the electric field.* This effectively narrows the barrier, and since the tunnelling current is exponentially dependent upon barrier width, this can lead to a dramatic increase in tunnelling current.

Holes can also tunnel but this behaviour is markedly less significant as they must overcome a larger potential barrier.

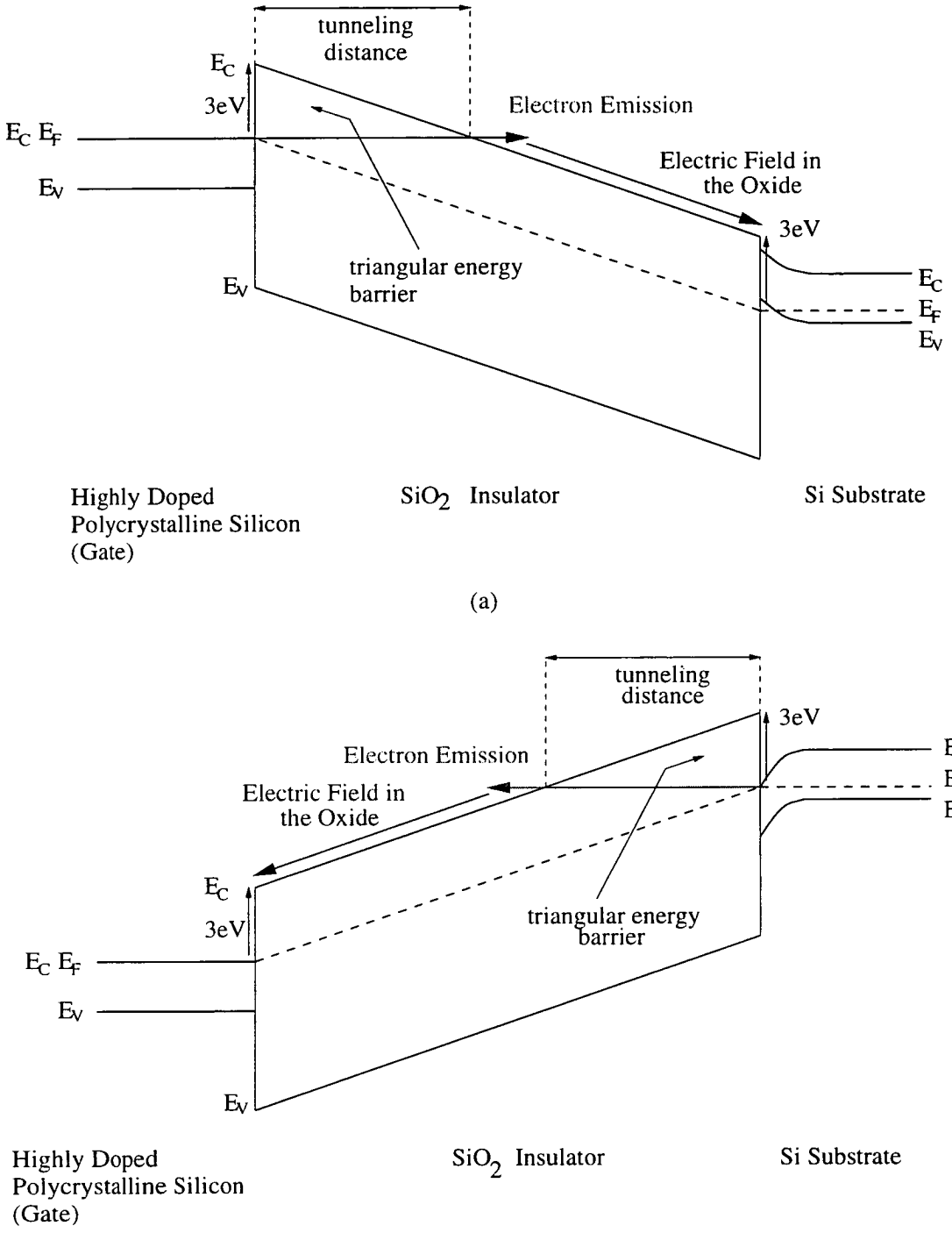


Figure 3.7: Energy band diagram of MOS structure with (a) a large *negative* bias on the heavily doped polysilicon gate, and (b) a large *positive* bias on the heavily doped polysilicon gate

In the case of the triangular potential barrier, $E_{potential}$ in Schrödinger's equation (3.3) is no longer a constant across the width of the barrier; the equation itself is therefore no longer a linear differential equation. In 1928 Fowler and Nordheim [61] provided a mathematical solution and this emission current has since become identified as Fowler-Nordheim tunnelling (FNT). The particular form of the solution is beyond the scope of this introduction; it will be simply noted that the transmission coefficient, T, of electrons into the oxide barrier becomes:

$$T = \exp\left(\frac{-4}{3}\frac{\sqrt{2m_{ox}}}{q\hbar F}\phi_B^{\frac{3}{2}}\right) \tag{3.5}$$

where F is the electric field and m_{ox} accounts for distortions of the electron wave due to atomic centres in the oxide. m_{ox} is usually taken to be about $\frac{1}{2}m$.

An expression for the Fowler-Nordheim tunnelling current can then be developed by integrating T over the occupied states of the semiconductor conduction band. The result is that the tunnelling current density, J, takes the form:

$$\frac{J}{F^2} = C\exp(T) \tag{3.6}$$

where

$$C = \frac{q^3 m}{16\pi^2\hbar m_{ox}\phi_B} \tag{3.7}$$

Thus the Fowler-Nordheim equation (3.6) shows a strong dependency on the electric field F.

$$F = \frac{V_{applied}}{t_{ox}} \tag{3.8}$$

where t_{ox} is the thickness of the oxide in the MOS structure, and $V_{applied}$ is the potential across it. More generally, the Fowler-Nordheim relation can be expressed as

$$I = \chi_1 V_{applied}^2 \exp\left(-\frac{\chi_2}{V_{applied}}\right) \tag{3.9}$$

where χ_1 and χ_2 are fitting parameters which accommodate the tunnelling relation and device characteristics such as oxide thickness and tunnelling area. In practice fitting these parameters normally takes place in a special large capacitor test structure which has a voltage applied across the plates with the resultant tunnelling current measured directly.

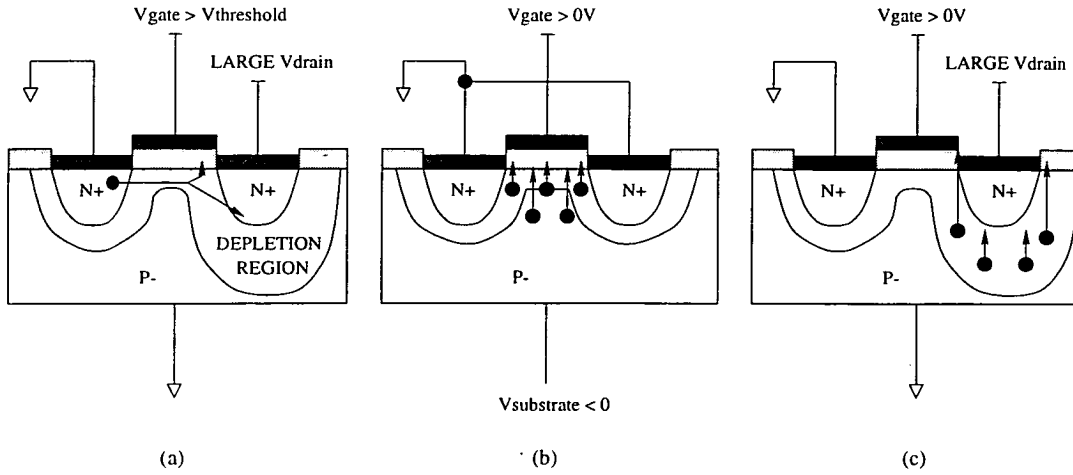


Figure 3.8: (a) Channel hot electrons, (b) Substrate hot electrons, (c) Avalanche hot electrons

3.5.3 Hot Electron Injection

Hot electrons are defined as electrons which have energy such that they are not in thermal equilibrium with the surrounding crystal lattice and are more than a few kT above the Fermi Level [186]. Such electrons may have sufficient energy to surmount the potential barrier associated with the SiO_2-Si interface and thus enter the insulating oxide. (This is in contrast to tunnelling in which electrons do *not* have sufficient energy to surmount the barrier and instead tunnel *through* it).

Channel hot electrons (CHE) (figure 3.8 (a)) are electrons flowing from source to drain which gain energy by acceleration in the high field region *around the drain.* Those approaching the Si-SiO_2 interface with sufficient energy surmount the barrier and enter the oxide.

Substrate hot electrons (SHE) (figure 3.8(b)) and avalanche hot electrons (AHE) (figure 3.8(c)) are closely related phenomena: here electrons are accelerated in the high field in the *surface depletion region* (SHE) or *drain depletion region* (AHE), and again those reaching the interface barrier with sufficient energy may enter the oxide.

Hot electrons only occur when the transistor is in, or near to, its 'snapback' regime [139] (see also section D.5.2).

Hot electrons which become trapped in the gate oxide can introduce a threshold shift which can age MOSFETs. This is particularly important in state-of-the-art submicron processes where the electric fields around the drain can become very intense due to the short channel length. Techniques to counteract this include dropping voltages across cascode arrangements or using smaller voltages (eg. 3.3V logic). Rather than being trapped, however, most hot electrons either return to the channel or are swept

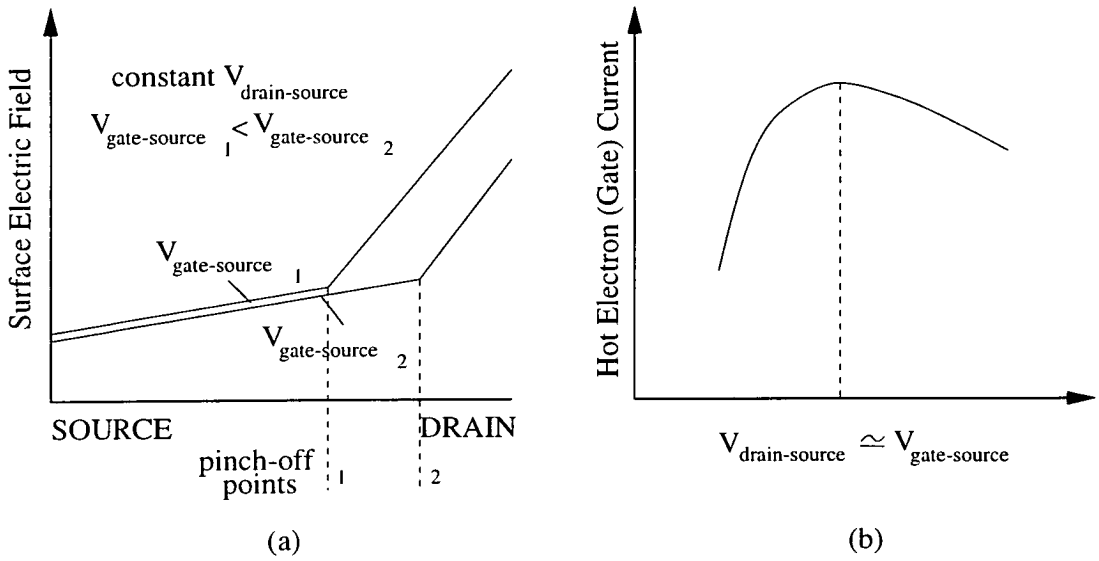


Figure 3.9: (a) Surface electric field distribution in relation to the pinch-off point of a transistor in saturation, (b) Sketch of typical CHE gate current characteristics showing a peak about $V_{gs} \simeq V_{ds}$

onto the gate.

SHE is enhanced in devices with heavily doped substrates where the hot-electron emission probability is very large [126], and is therefore likely to be more pronounced for transistors in the well. Mann [106] reports floating gate memories in MOSIS using AHE injection by means of a *gated diode* which is essentially a source-less transistor. The drain had to be pulsed at values close to the *pn*-junction breakdown voltage.

Maximum Gate Current

I_{gate} due to CHE peaks when the high lateral field in the substrate needed to 'heat' the electrons coincides with a suitable vertical field which modulates the barrier height and allows hot electrons to enter the oxide [139].

In fact this peak occurs when $V_{gate} \simeq V_{drain}$. This is because *(i)* when $V_{gate} < V_{drain}$, the field in the oxide reverses at some point in the channel, so that electrons emitted near the drain are attracted back towards it, and *(ii)* when $V_{gate} > V_{drain}$, the channel tends towards the linear region and the increased channel conductance will reduce the peak electric field in the substrate and so reduce the electron emission [142, 184]. This is illustrated in figure 3.9.

3.5.4 Digital Floating Gate Memories

The use of floating gate semiconductor memories for digital data storage applications is well established and dates from the early 1970s. These devices will be described here briefly so that analogies can be drawn with analogue floating gate memories later. Also such devices can replace the SRAM cells required by previously described analogue refreshed memories such as to achieve power-down non-volatility.

EPROM

The FAMOS (*f*loating gate *a*valanche injection *MOS*) EPROM, (*e*lectrically *p*rogrammable *r*ead *o*nly *m*emory), was introduced by Intel in 1970 [63] based on the structure shown in figure 3.4 [90].

Writing of a selected cell involves hot electron injection onto the floating gate by application of high gate and drain voltages through the addressing scheme. This modifies the threshold of the cell such that unwritten cells pass current in read mode whilst written cells do not. The current in the sense transistor then defines a logic 0 or 1 (thus the actual level of stored charge is unimportant provided a distinction can be made between the two logic states). The chip is bulk reset by UV exposure with a special erase lamp for several minutes. (EPROMs may typically be re-programmed 10^2 to 10^3 times [136] but are commonly used as one-time programmable (OTP) memories). EPROMs require a range of 12V to 25V supplies for programming and draw several tens of milliamps during injection [144].

Once a logic value is established on the floating gate it remains there for a long time $\sim$ decades. Thus power-down non-volatility is attained (see appendix E for discussion of floating gate non-volatility testing).

EEPROM

The obvious disadvantages of the EPROM are the requirement for a UV lamp for resetting the memory and the slow speed of this process. This led in the early 1980s to the development of electrically programmable non-volatile memories called EEPROMs (*e*lectrically *e*rasable *p*rogrammable *r*ead *o*nly *m*emories). Instead of CHE and UV, these used Fowler-Nordheim tunnelling for both electron injection and removal from the floating gate.

The most popular EEPROM structure is the FLOTOX (*flo*ating gate *t*unnel *ox*ide) cell illustrated in figure 3.10 [88, 184]. An alternative, FETMOS, (*f*loating gate *e*lectron *t*unnelling *MOS*) is like FLOTOX but without gate oxide (tunnel oxide extends the length of the channel) [94]. In both cases a select transistor is added in series with

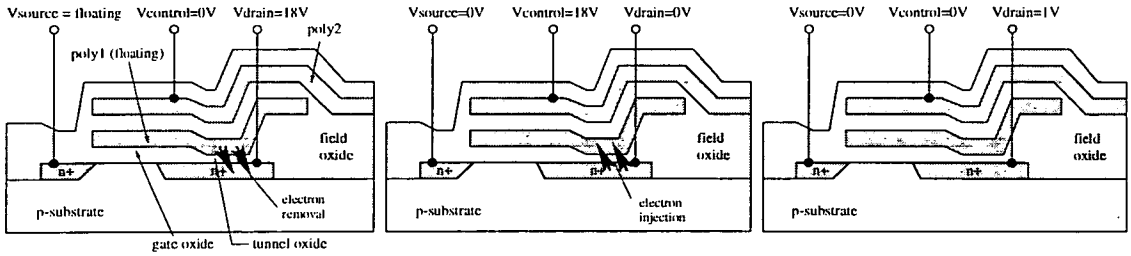


Figure 3.10: Cross sectional structure and operating modes of FLOTOX device

the floating gate transistor. This provides cell isolation during readout and allows programming of individual digital words.

There are three distinct modes of operation in both devices as shown in the figure. Removal mode applies a high drain voltage pulse ($\sim 5-20$ms – much slower than CHE injection) to cause tunnelling of electrons off the floating gate due to the high electric field in the thin tunnelling oxide (the source is floating to prevent CHE injection); in injection mode a high voltage pulse is capacitively coupled onto the floating gate to cause tunnelling of electrons onto it; and in read-out mode a small voltage is applied to the drain (the lowest possible voltage is used to prevent read-disturbance due to low level electron removal). As for EPROM, sensing of current flow (eg. with a threshold of ~ -4V) or its absence (eg. with a threshold of $\sim +8$V) then indicates a logic 1 or 0 [153]. The high voltages required for programming may be supplied from off-chip but in modern EEPROMs are generated on-chip from 3.3V or 5V supplies using charge-pumping circuits. Transistor doping profiles are designed such that the voltages can be easily handled on chip.

The difference between the programmed thresholds (eg. ~ -4V to $\sim +8$V) is known as the 'threshold window'. The actual width of this window depends on cell-to-cell and process variations but provided the on/off nature of the programmed threshold is maintained this variation is not critical. However, some of the electrons travelling through the gate oxide during programming become trapped during each programming cycle (see section D.2). These trapped electrons distort the electric field in the tunnelling oxide reducing its effectiveness in inducing tunnelling. Thus, in each cycle, a reduced number of electrons is transported for each fixed programming pulse and this manifests itself in a gradual decay of the difference between the programmed and erased thresholds (this is known as 'window closure'). Eventually this behaviour becomes so severe that the sense circuitry can no longer distinguish whether a cell is programmed or erased and the EEPROM fails. Because of the high electric fields it is common that TDDB (time dependent dielectric breakdown) occurs before complete

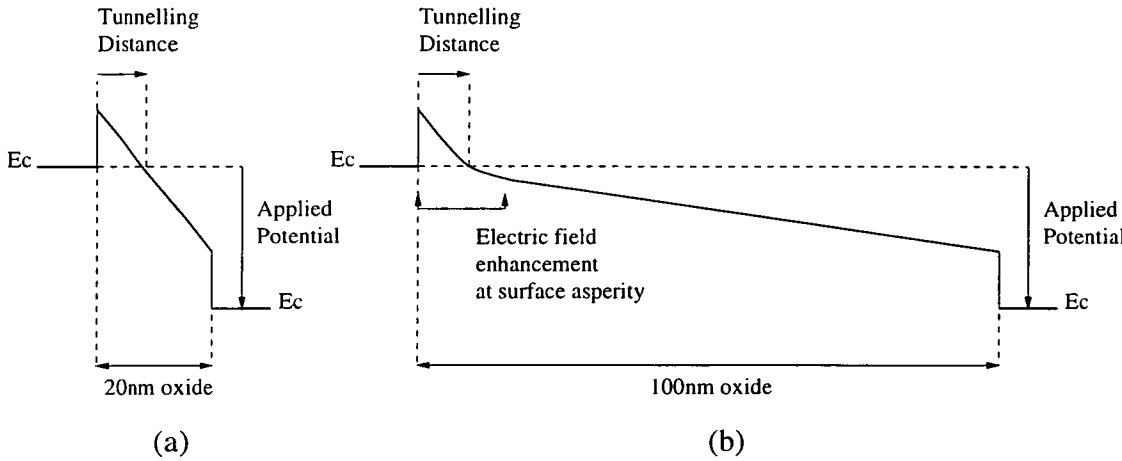


Figure 3.11: (a) Conduction bands in tunnelling oxide EEPROM structure, (b) Conduction bands in textured polysilicon EEPROM structure

window closure[5]. Typically EEPROM failure occurs after about 10^5 to 10^7 cycles [30] depending on the manufacturer and part with specified retention times of about 10 years. Shadow EEPROM (or Non-Volatile RAM) provides SRAM and EEPROM on a single package giving fast writing times in the presence of power but non-volatility for power-loss.

Textured Poly EEPROM

To avoid the oxide breakdown problems associated with thin tunnelling oxides, TPFG (*t*extured *p*olysilicon *f*loating *g*ate) EEPROMs were developed. Instead of a thin oxide, these used specially textured polysilicon surfaces which leads to sharp surface points known as *asperities*. Field lines are concentrated at these asperities resulting in a localised enhancement of the electric field (by a factor of about 3-5 [95]).

This behaviour is illustrated by the conduction band behaviour in figure 3.11. Because of the enhanced electric field around surface asperities, the required tunnelling distance (and so tunnelling current) is the same in both cases for the same applied electric field despite the use of a thicker oxide (lower *bulk* oxide electric field). This design provides two reliability gains:

1. The thicker oxide is less susceptible to microscopic defects.

[5]EEPROMs tend to have a higher reliability than EPROMs despite the use of higher electric fields. This is because these high fields accelerate defect related breakdown of oxides during pre-shipping screening; the same defects would remain latent in EPROMs [182].

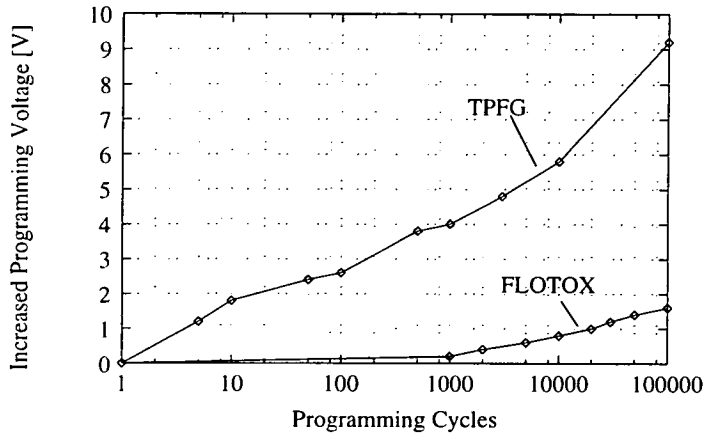


Figure 3.12: Reproduction of data from [112]: The increase in programming voltage required to maintain the original threshold window width for TPFG and FLOTOX EEPROMs

2. The bulk of the oxide has to withstand a lower electric field reducing field accelerated breakdown.

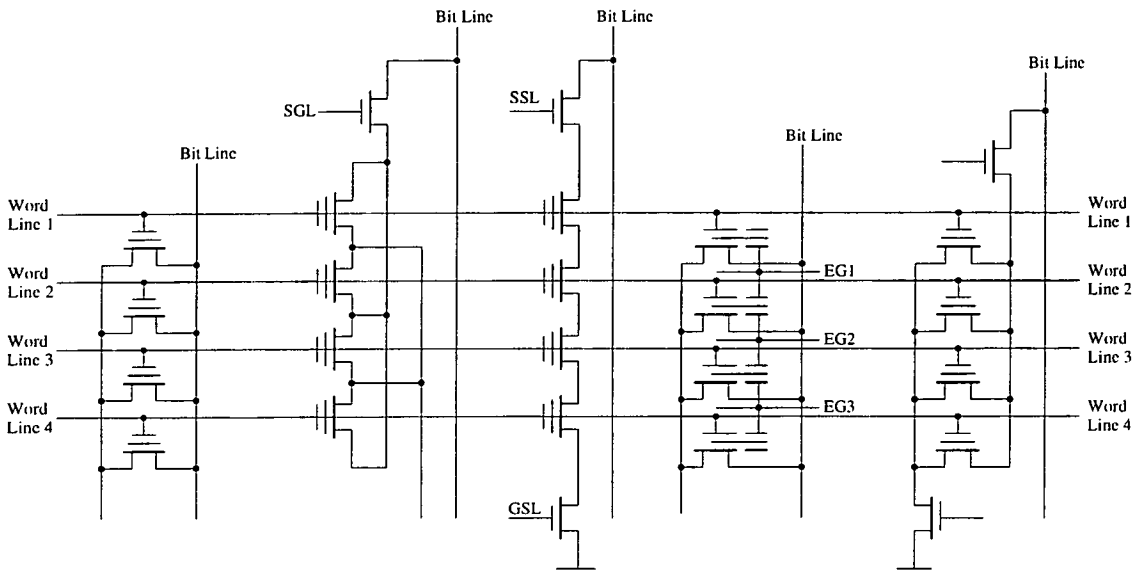
The distribution of initial threshold window widths of TPFG EEPROMs is typically three times that of FLOTOX EEPROMs because of the poor control over the number, size, distribution and shape of surface asperities [112]. Additionally, window closure is more severe than for FLOTOX for three reasons:

1. The oxide is thicker, so there is more trap-up (in fact, trap-up is proportional to the square of the oxide thickness [112]).

2. The effect of trapped charge close to the interface causes disruption of the asperity electric field enhancement to an extent greater than trapped charge would impact on a planar electric field. This is called trap acceleration.

3. The lower electric field means that there is less field induced de-trapping.

Figure 3.12 shows an experiment in [112] in which the programming pulse magnitude in each cycle is increased to maintain the same threshold window width (ie. by application of sufficient voltage to overcome the field degradation due to trapped charge). The figure clearly shows virtually no trap-up in FLOTOX cells below 1000 cycles and in all cases much less than the trap-up in TPFG cells. As a consequence TPFG EEPROMs tend to fail primarily due to window closure, whereas tunnelling oxide EEPROMs fail primarily due to TDDB (retention faults). Thus it is argued by TPFG EEPROM manufacturers [182] that this is a better trade-off for high densities where a single defect may destroy a FLOTOX chip.



NOR	DiNOR	NAND	T-Poly	AND
Double Poly	Triple Poly	Double Poly	Triple Poly	Triple Poly
Double Metal	Double Metal	Single Metal	Single Metal	Double Metal
CHE Write	FNT Write	FNT Write	CHE Write	FNT Write
FNT Flash	FNT Flash	FNT Flash	FNT Flash	FNT Flash

Figure 3.13: Major flash cell technologies [124]

Flash EEPROM

Flash EEPROMs (or simply Flash memories) were designed to fill the niche between the low bit cost and high data density of EPROM and the higher bit cost, lower data density but electrical-programmability convenience of EEPROM.

Most flash memories (eg. Intel, AMD) use fast CHE ($< 10\mu s$) for electron injection and FNT for electron removal; bi-directional FNT has also been pursued (eg. Atmel) since, although this leads to larger cells, devices can be programmed at lower voltages and currents and also demand less precise control (However both types are expected to converge, with the use of charge pumps, on supply voltages of less than 3V by 1998/9 [32]). To achieve the higher data density the select transistor has been removed and with it selective byte programming: flash memories must first be bulk (or *flash*) erased by page or sector. A typical flash cell is therefore about 2.5 times smaller than a typical EEPROM cell [136] (and even 1.5 times smaller than a DRAM cell). A charge-pump may be used to generate the programming voltages or these may be provided from off-chip. Endurance performance typically lags EEPROMs by about one order of magnitude.

Some common flash memory configurations are shown in figure 3.13. Each has its

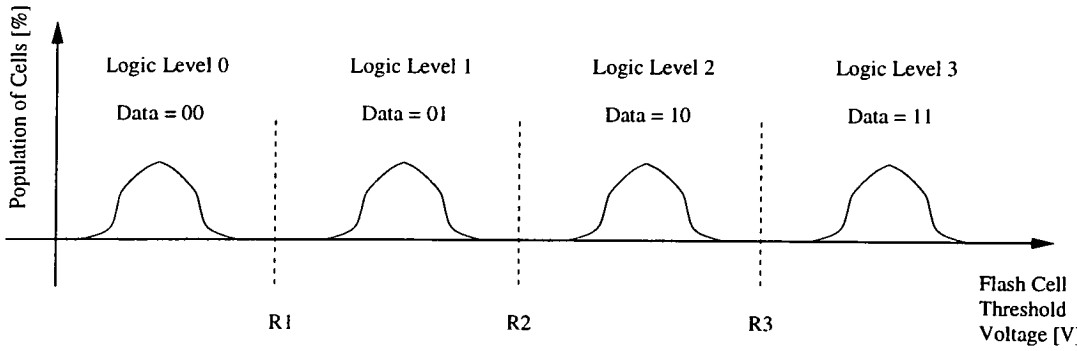


Figure 3.14: 4-level (2 bit) MLC Program Thresholds [43]

own advantages: Intel and AMD promote the NOR configuration but higher densities can be achieved with the NAND configuration. However NAND is less expandable and has a higher initial access time making it less suitable for multiple random access applications (although appropriate to sequential data, such as in disk-drive replacements) [173].

MultiLevel Flash

The mass market for non-volatile digital memory in emerging digital equipment such as cellular telephones, memory cards, updatable BIOS storage (and also personal digital assistants, PDAs, which are expected to emerge from 1998 onwards [31]) is demanding a growth in digital non-volatile density which may not easily be achieved through process scaling alone.

This has prompted recent interest in storing more than a single bit (0,1) on a single floating gate by use of multiple threshold voltages (MultiLevel Cells - MLCs) as illustrated in figure 3.14. However, the difficulty in reliably fabricating consistent thin oxides has brought practical difficulties to this approach – while the large window between 0 and 1 thresholds in standard Flash memories is tolerant to a wide process variation, the window is sharply reduced by every extra bit to be stored. Programming thresholds reliably within tightly defined representative regions then becomes a serious practical difficulty.

Ohkawa *et al* [129] propose a 4-level FN-NOR-MLC design (ie. NOR configuration but with drain-gate tunnelling rather than CHE injection). They use a Drain Voltage Controlled MultiLevel Programming (DCMP) scheme based on the characteristics shown in figure 3.15. In this scheme 0,4,5 or 6V is applied to the drain (bit-line) for 1ms leading to a different shift in the threshold from the initial bulk-erased state. As the flow chart shows, the resultant programmed data is verified against latched input and programming repeated until successful, thus covering the range of process

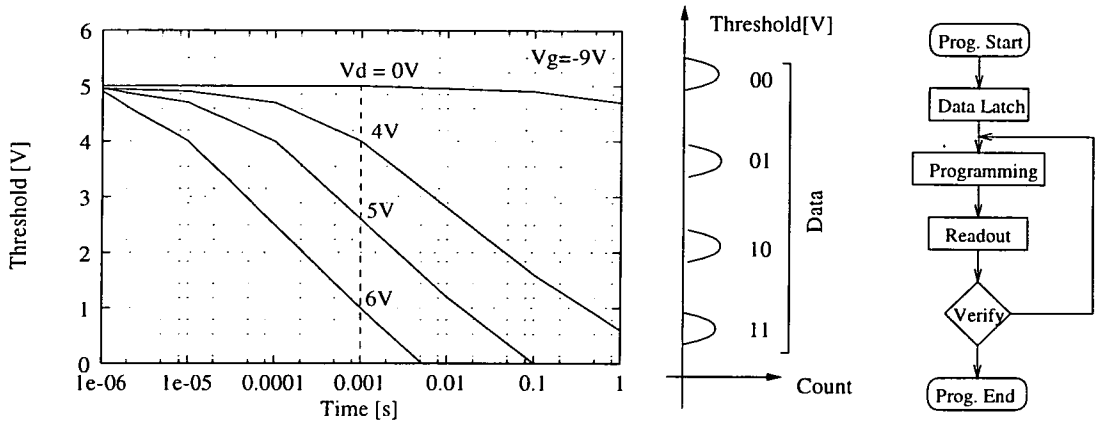


Figure 3.15: Programming characteristics and flow chart of DCMP Scheme

variations.

Read-out proceeds by applying voltage $R2$ (figure 3.14) to the word line to determine the MSB, and subsequently applying $R1$ or $R3$ to determine the LSB. This technique is known as wordline sweeping read (WSR).

Jung *et al* [89] propose a NAND-MLC scheme using incremental voltage pulse trains for programming with interpulse evaluation of the stored data and WSR read-out.

Intel launched commercial MultiLevel Flash products (StrataFlash^{TM} 32Mbit and 64Mbit densities) in September 1997 [44] . These devices use the 4-level (CHE)-NOR-MLC configuration. CHE programming proceeds using a train of fixed magnitude drain voltage pulses until the correct threshold is established within the specified ranges.

Intel's read-out approach is to directly compare the sense current of the cell with those of transistors with thresholds $R1$,$R2$ and $R3$ (figure 3.14) and using decoding logic to determine the data stored (see figure 3.16). This read-out approach is much more area intensive than WSR since analogue sense circuitry must be distributed by page buffer, but it allows evaluation of stored data in a single clock cycle. Some compromises - a reduced specified endurance and a 5V rather than 3.3V power supply - were required to ensure correct performance. ISD and SanDisk also produce similar components [134].

Montanari *et al* [115] propose another read-out scheme using analogue Euclidean distance calculator circuits to determine the distance between the Flash sense transistor current and currents produced by cells with thresholds at the exact centres of the desired level distributions. A winner-takes-all (WTA) circuit then determines the measured data as the nearest level to the measured current.

Most analysts believe that flash memories will eliminate the EPROM market by

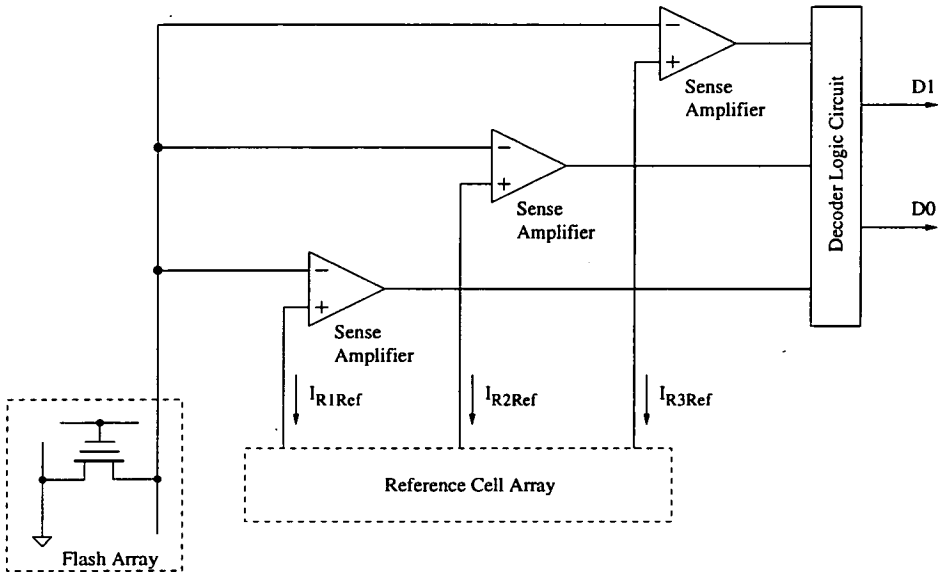


Figure 3.16: Intel StrataFlash^{TM} read scheme [43]

	EPROM	Flash	EEPROM
Typical Storage Use	Fixed programs (eg. embedded systems, BIOS)	In-system modifiable programs	Frequently updated programs and data
Typical Number of Writes	OTP	up to a million times	up to 10 million times
Available Densities	256kbit - 8Mbit	256kbit - 16Mbit	1kbit - 64kbit (serial) 65kbit - 4Mbit (parallel)

Table 3.2: Comparison of EPROM, Flash and EEPROM adapted from table in [31]

the millennium [31] and many manufacturers have already halted or reduced EPROM development and production. While EPROM currently still has a cost advantage, this is largely annulled by the ceramic packaging required for UV-erase and so the bulk of EPROMs are now sold as OTP memories with no erase window. In fact UV will not be able to penetrate all the cells to assure total erasure if EPROMs were to become much denser and so the technology is also reaching a natural limit.

3.5.5 Analogue Floating Gate Memories

The digital floating gate memories described so far all have the potential for analogue operation. Multilevel flash memories, especially those with analogue sense circuitry blur the distinction between digital and analogue devices; here but for the final quantization stage, multilevel flash can be accurately described as an analogue floating gate memory.

The last decade has seem some interest in the *explicit* storage of analogue values on EEPROM-type devices which may be exploited in several forms:

- Floating gate voltage, $\phi_{fg} = Q_{fg}/C_{fg}$.
- Floating gate *threshold* voltage, as viewed from the control gate.
- Current in floating gate sense transistor.

Non-Linear Programming

A significant problem with *analogue* programming is non-linearity. Put simply, floating gate programming is self-limiting: each injected electron increases the electric field *opposing* the injection of further electrons by FNT. Using a train of fixed magnitude and duration programming pulses, the packet of charge injected (or removed), ΔQ_{fg}, is progressively smaller than for the preceding pulse. The injection of charge reduces the tunnelling potential in the tunnelling oxide, which, as equation 3.9 describes, results in a strong reduction in the tunnelling current. This means that ϕ_{fg} will change $\sim$ logarithmically with the number of identical applied pulses.

Fujita *et al* [65, 66] propose that the tunnelling oxide and floating gate storage capacitor be separated by a large resistor (figure 3.17) through which injected charge diffuses slowly. Provided the resistance is sufficiently large, the change in stored charge in the *total* device (ie. $C_i + C_g$) is very small while C_i charges sufficiently to halt tunnelling. An analysis of ΔV_g shows it to have a very small (C_i/C_g) dependency on V_g, compared to a direct dependency in a conventional EEPROM; this increases programming linearity so that roughly equal packets of charge are injected by each

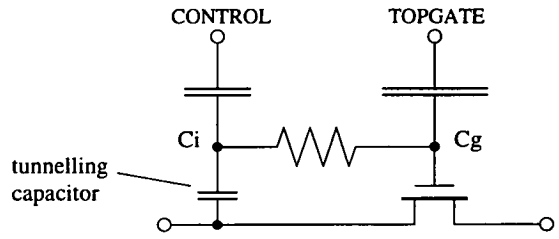


Figure 3.17: High resistance floating gate structure: bipolar FNT occurs in the indicated tunnelling capacitor

Application	Example	Programming Technique
ANN weight storage		see later for description
Fuzzy logic membership function	[108]	Not stated
Analogue amplifier trimming	[21]	Fixed $\{V_{PP}, t_{PP}\}$ pulse train with interpulse evaluation by latching comparator
Analogue storage of speech signals	[179]	Coarse linear mapping
Removal of fixed pattern noise in CMOS imager	[50]	Continuous tunnel voltage application, attenuated in continuous-time by flipping of comparator

Table 3.3: Applications of analogue floating gate memories and analogue floating gate programming techniques.

consecutive pulse[6] (A variation on this idea involving the replacement of the resistor with a switched thin-film transistor (TFT) has also been proposed to speed programming [122]) . Shima and Rinnert [151] propose an alternative linearising method in which a sample-and-hold circuit measures the value of ϕ_{fg} between pulses and compensates the V_{PP} magnitude accordingly to account for the change in tunnelling potential.

However, despite improved linearity, Sin *et al* [152] argue that even for a special fabrication process, the effects of intercell mis-match, non-idealities and the effects of trap-up make a direct mapping infeasible for reasonable levels of precision, preferring instead a feedback-based method using a train of fixed $\{V_{PP}, t_{PP}\}$ pulses with interpulse evaluation. An 8-bit equivalent precision in less than 20μs was claimed.

[6]And also allows simultaneous valid readout of the stored analogue value *during* the application of programming pulses.

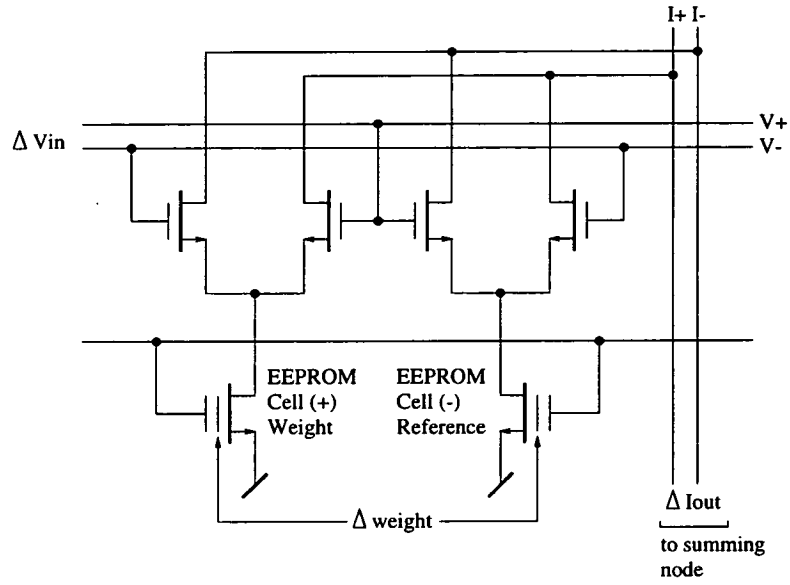


Figure 3.18: ETANN synapse implementation

The applications of analogue floating gates fall into a number of categories[7] given in table 3.3. Further discussion of on-chip floating gate programming schemes is given in section 7.2.

Commercial Analogue Floating Gate Products

Analogue floating gate memories have seen far less commercial exploitation than their digital equivalents; it *has* however found a niche for storing short duration audio recordings without degradation by quantization. A significant contributor to this market is Information Storage Devices, Inc. of San Jose, California [84, 163], whose ChipCorder^{TM} has been incorporated into such products as talking keyrings, cameras, photographs, telephones, alarms, warning signs and even a talking flower arrangement.

Analogue Floating Gate ANNs

Of particular interest here is the use of floating gates for weight storage in analogue VLSI ANNs of which there have been a number of examples in recent years.

Perhaps the best known is probably the ETANN[8] chip [77, 42] which was for a

[7] An additional category are analogue arithmetic elements using multiple input floating gates (MIFGs) [140, 147] - however these exploit capacitive summation operation of floating gate, rather than their non-volatile properties (and as such, isolation by a pass transistor is sufficient). Therefore these applications are not included in the list.

[8] *E*lectrically *T*rainable *A*nalog *N*eural *N*etwork.

while marketed by Intel Corporation[9] but is no longer produced.

The ETANN chip contains 10240 synapses based on 4-quadrant Gilbert multipliers suitable for MLP or Hopfield ANN implementation. The floating gate devices establish the tail current as shown in figure 3.18.

ETANN was bundled with a training software suite called the Neural Network Training System (iNNTS). Network weights were established by simulation of the chip using this software and then serially downloaded to the chip. Analogue programming of the weights used an iterative algorithm which modified the bipolar magnitude of a fixed-width FNT programming pulse until the correct weight was measured. iNNTS then provided chip-in-the-loop (CIL) trimming by using the actual chip for the feedforward path of the simulation; this was to allow some of the hardware imperfections to be 'trained out' to optimise performance. Bake retention (see appendix E) tests suggested a weight precision of 4-bit could be achieved for 10 years [42]. Data D/A and A/D conversion bottle-necks and a rather inflexible architecture resulted in ETANN failing to become the generic ANN accelerator for desk PCs for which it was originally intended.

ETANN, and some more recent floating gate ANN designs are summarised in the following table:

Analogue-Value Floating Gate ANNs

Authors	Holler, Tam, Castro, Benson, (Intel) [77, 42] (1989,1991)
Process	Intel CHMOS III EEPROM
Neural Operation	Perceptron (Optional feedback connectivity), allows MLP, Hopfield Net configurations
Description	2 floating gates form differential weight input to 4-quadrant Gilbert multiplier
Programming Mechanism	Bipolar FNT
Programming Scheme	Download: Software controlled iterative programming algorithm to establish weights
Authors	Lee, Sheu and Yang [99] (1991)
Process	'Standard' CMOS reported (but 26V handling transistors used are highly rated for a standard process)
Neural Operation	Perceptron allows MLP and Hopfield Net configurations
Description	Differential transconductor amplifier synapse with floating gate at one side of the differential pair; input in the form of a tail current
Programming Mechanism	UV-erase or Bipolar FNT (method of V_{PP}^{-} application not stated)
Programming Scheme	Not stated

[9]Part No.: 80170NX

Analogue-Value Floating Gate ANNs

Authors	Sin, Kramer, Hu, Chu and Ko [152] (1992)
Process	EEPROM
Neural Operation	Loosely linear 1-quadrant multiplier
Description	High tunnelling capacitance between gate and drain make I_{ds} current of sense transistor strongly dependent on V_{ds} (approximately linear above an offset)
Programming Mechanism	Bipolar FNT
Programming Scheme	Download: Iteration using fixed magnitude pulses with inter-pulse evaluation of drain current

Authors	Kosaka, Shibata, Ishii and Ohmi [93, 150] (1995)
Process	EEPROM
Neural Operation	'Neuron-MOS' with hardware backpropagation (HBP) on-chip learning (learns XOR)
Description	Programming linearisation by use of local latched source-follower arrangement to ensure tunnelling potential is the same for each pulse
Programming Mechanism	Bipolar FNT
Programming Scheme	Hardware backpropagation

Authors	Berg, Sigvartsen, Lande and Abusland [13] (1996)
Process	SLV-CMOS
Neural Operation	MLP with on-chip learning (learns XOR)
Description	Floating gates form differential pair gate voltages in analogue multipliers
Programming Mechanism	UV activated conductances
Programming Scheme	Backpropagation with weight changes determined by the time a training pattern is presented and the UV light intensity

Authors	Marshall and Collins [107] (1996)
Process	SLV-CMOS
Neural Operation	RBF
Description	see section 5.3 for operation of Euclidean distance calculator in hidden layer; modified Gilbert multiplier in output layer - floating gates set the differential pair voltage inputs rather than the tail currents as in ETANN
Programming Mechanism	Refresh through switch; non-volatile operation demonstrated separately: UV preset and unipolar FNT program
Programming Scheme	Download: Direct weight programming with switch; use of current comparator with feedback proposed for non-volatile operation

Authors	Morie, Fujita, Uchimura [117] (1997)
Process	SRAM/EEPROM (Shadow) (thin-oxide and high ohmic poly)

Analogue-Value Floating Gate ANNs

Neural Operation	MLP, Boltzmann Machine and Hopfield Net with on-chip learning (learns XOR)
Description	Neural circuits not described
Programming Mechanism	Bipolar FNT
Programming Scheme	Use of resistor floating gates [65, 66] to linearize programming for on-chip backpropagation

It is worth highlighting that multilevel *digital* floating gates impose more stringent demands on the memories since slight level errors generally lead to graceful decay in the performance of the analogue circuits but would lead to completely incorrect digital byte readout and therefore unacceptable storage performance. For this reason most MLC Flash designs use only 2^2 discrete levels. Up to 2^4 levels may be introduced but only with the incorporation of error correcting circuitry [124]. This should be compared with the quoted 2^8 levels of recent ISD analogue chips [163].

3.6 Summary

In this chapter the various properties of non-volatile analogue technologies have been described and methods classified into power-up and power-down. Power-up methods are widely applicable since they involve circuit designs but demand a continuous application of power to retain their analogue parameter. Power-down methods generally involve modified or special fabrication processes but afford non-volatility which is immune to the removal of power. The latter class is superior for the sensor applications pursued. One particular device - the floating gate - shows promise for SLV-CMOS adaption and its programming mechanisms and various guises have been described in some detail.

In addition to the non-linearities, mis-match and trap-up effects which plague analogue floating gate memories and so complicate the programming process, further complications arise if an attempt is made to implement these devices in SLV-CMOS due to the lack of native thin oxides, and high voltage transistors. These specific problems – and proposed work-arounds – are the subject of the next chapter.

Chapter 4

Standard CMOS Floating Gates

4.1 Introduction

Floating gate memories have been introduced in the preceding review of non-volatile VLSI technologies. This chapter describes some of the problems involved with mapping floating gates from special EEPROM fabrication processes into the SLV-CMOS which is available for this work, and discusses the related literature which formed the basis of the research detailed in the rest of the thesis.

4.2 Floating Gate Requirements

Two specific requirements for developing floating gate memories for RBF circuits for use in integrated sensor applications make them distinct from conventional EEPROM memories:

- Analogue storage must be possible for compact compatibility with the analogue / pulse-stream RBF implementation pertinent to the sensor application.
- All circuits and memory elements must be fabricated on the same wafer in the same readily-available SLV-CMOS manufacturing run without recourse to post-processing steps. Manufacturing must be totally transparent between tape-out and delivery.

4.3 Limitations of SLV-CMOS

As noted in section 3.5.2, tunnelling is strongly dependent on the electric field in the oxide. Electric field may be enhanced globally in a special ultra-thin tunnelling oxide, or locally enhanced on a specially textured emission surface. Such oxides may be difficult to fabricate reliably and so result in a more costly non-volatile processes. They are therefore absent from SLV-CMOS where the thinnest oxide (the gate oxide, GOX) may typically be about four times thicker than a tunnelling oxide.

Even with the use of tunnelling oxides and textured emission surfaces, the *field emission threshold*, V_{fe}, the point at which tunnelling current becomes significant (analogous to the forward threshold of a junction diode due to the exponential characteristic of the tunnelling equation) may often be higher than conventional supply voltages (eg. +5V,+3.3V). Certain breakdown characteristics are associated with such high voltages (see appendix D). Much of these can be controlled by appropriate choice of doping densities and profiles, and thus care is taken to design a fabrication process in which tunnelling control voltages can easily be switched by transistors. Again, no such safeguards exist for SLV-CMOS (in which the high voltage problem is of course exacerbated by the absence of tunnelling oxides/surfaces).

4.4 Review of Preceding Research in Standard Low-Voltage CMOS Floating Gate Memory Cells

Several research efforts have been aimed at the integration of floating gate memory elements in SLV-CMOS, either for use in trimming of analogue circuits or for analogue VLSI ANN synaptic weight storage. This work will be described here, to provide the basis for the first test chip designed in this project, and also to provide a context of the available knowledge on the subject when the work was instigated late in 1994.

4.4.1 Features Under Control: VLSI Layout

In using an 'off-the-shelf' SLV-CMOS, only lithographic features are actually under the control of the VLSI designer - these are determined entirely by the drawn mask layout. However, preceding research into the physics of semiconductor devices suggested that it *was* possible to influence the characteristics of electric field (and thus tunnelling) by purely lithographic means. The following is a review of the work based on this idea. Note that a prime objective of much of this work was to lower the field emission threshold, in an attempt to control programming with voltages low enough that they did not damage the memory or other circuits on the chip.

4.4.2 Corner Effects

In 1989, Carley [21] pursued the idea of localised electric field enhancement using purely lithographic features. The electric field in structures is enhanced at corners and

sharp edges, using the device shown in figure 4.1(a)[1], the principle being that it is the electric field at the interface rather than in the oxide as a whole which determines the potential barrier width, and hence the tunnelling current (figure 4.1(b))[2].

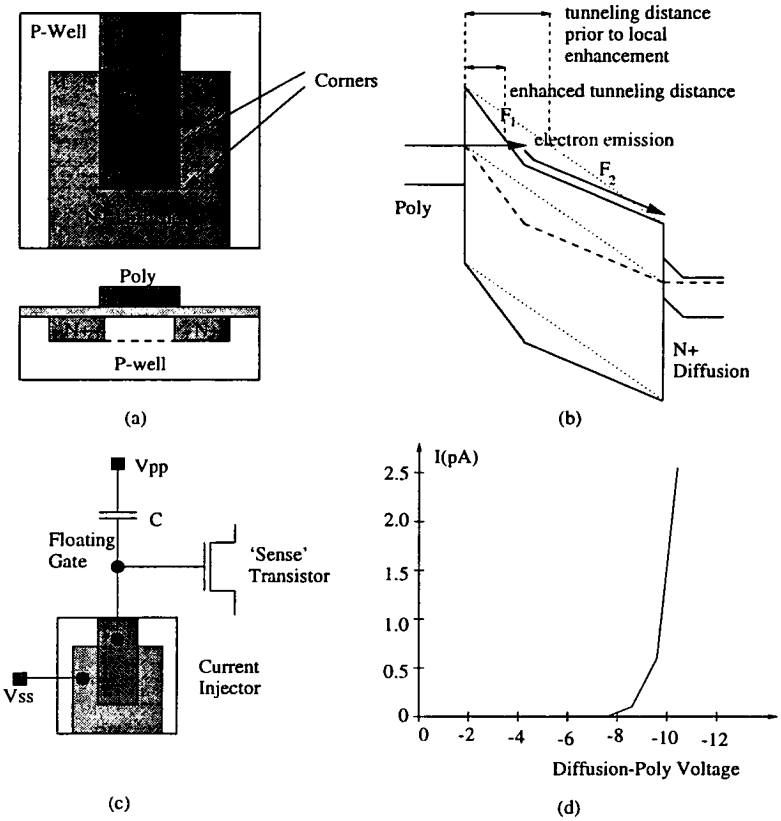


Figure 4.1: (a) Top and cross-sectional sketches of current injector, (b) Principle of potential barrier narrowing with localised electric field enhancement, (c) Floating gate system with current injector, and (d) Current injection characteristics of Carley's device

The high programming voltages, V_{PP}, are coupled in through capacitor, C, since the floating gate must remain electrically isolated (figure 4.1(c)). The injected current

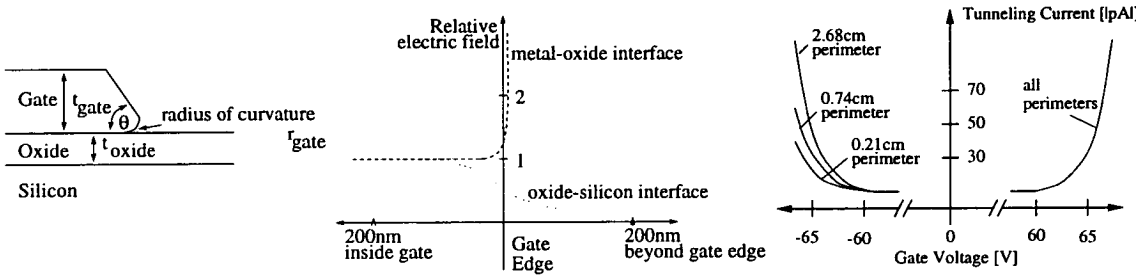
[1]Carley describes his design as a *current injector*. However such structures are also commonly referred to as *tunnelling injectors* because the current is a tunnelling one. However, no one term has become standard in the literature. In this thesis an alternative term, *tunnelling capacitor* is proposed, as this highlights not only the tunnelling current aspect of the device but also the parasitic capacitance due to the presence of two overlapping plates.

[2]Strictly speaking, Fowler-Nordheim tunnelling refers exclusively to the triangular potential barrier as considered by Fowler and Nordheim in their 1928 paper. However it is common practice to refer to this phenomenon as Fowler-Nordheim tunnelling regardless of the specific shape of the potential barrier which may be prone to many forms of electric field distortion; this convention will be followed here. It should be noted, however, that the Fowler-Nordheim current equation holds exactly *only* if the potential barrier is triangular.

therefore sets up a voltage on the gate of the sense transistor. Current characteristics of the device are shown in figure 4.1(d). Carley reported that the p-well potential had no effect on the tunnelling current. This suggested that tunnelling was entirely between the active area and the polysilicon.

4.4.3 Edge Effects

Fringing effects result in an increase of dielectric flux which means that the actual field strength across the oxide is enhanced at the edges [170]. Enhancement is particularly strong if an abrupt sharp edge exists at the edge of capacitor plate.



(a) (b) (c)

Figure 4.2: (a) Geometrical description of MOS gate edge, (b) Electric field at two interfaces as a function of the distance from the edge of the gate electrode, (c) Sketch of Hook and Ma's measurement of tunnelling current in $2.2^{-3}cm^2$ GOX capacitors with various perimeters (aluminium gate, r_{gate} = 6nm, t_{GOX} = 100nm). Note the absence of perimeter effects for tunnelling from the substrate (positive gate voltages) due to the absence of an edge.

In Hook and Ma's theoretical analysis [81], a large field enhancement around the gate edge is apparent (see figure 4.2(b) and (c)). Due to the the exponential dependence of current density on the cathode field, this suggests that a large gate injection current would flow through this small area (some experimental results from their work are sketched in figure 4.2(c)).

Hook and Ma also reported that the enhanced perimeter current is more pronounced at low fields (such as those to be found within a floating-gate device). This was consistent with previous work [8] which reported significant deviations from the Fowler-Nordheim equation at low fields when edge effects were not included. Hook and Ma's model emphasises the field enhancement effect of the microscopic radius of curvature at the bottom of the gate (r_{gate} in figure 4.2(a)).

In modern MOS structures, the gate is polysilicon, not aluminium. The high temperatures involved in fabrication tend to result in a much larger microscopic radius of

curvature than for metal. Muto *et al* [121] have looked at such structures in the context of floating gate flash memory cells ($t_{ox} \simeq$ 10nm, $r_{gate} \simeq$ 20nm). Although still present, such relatively large radii tend to reduce the electric field enhancement effect of edges. Of course, r_{gate} is not a parameter under the control of the VLSI designer.

4.4.4 Thinning

Thomsen and Brooke [161] took an alternative approach which used the interpoly oxide in a double polysilicon process, rather than the gate oxide, and attempted to make the oxide *thinner* along its edges. It is important to make the distinction between the two types of edge effects: fringing and thinning. Fringing might also be expected to occur along thinned edges.

A sharp, abrupt edge is generally produced by the etch of the poly1 and poly2 layers. Thus it is difficult to reliably cover this edge with thin interpoly oxide (grown by thermal oxidation of poly1 immediately prior to poly2 deposition) because this may result in a break in the oxide and/or a shorting of the poly1 and poly2. Thus the crossing of a poly1 boundary with poly2 is often not permitted [68].

Thomsen and Brooke laid out a poly-poly capacitor with the top plate (poly2) overlapping the bottom plate (poly1) (figure 4.3(a)). Their hope was that, at the boundaries, the poly2 and poly1 would not actually short but that significant thinning at the edges would occur. This effect could be observed using a Scanning Electron Microscope (SEM) (sketched in figure 4.3(b)). In addition to thinning, sharp edges were observed on the poly 1, leading also to localised field enhancement. For convenience, this will be referred to as an *overlapped* poly-poly tunnelling capacitor, as opposed to an *underlapped* poly-poly tunnelling capacitor, in which the design rules are obeyed (and thus a normal interpoly capacitor is formed and no thinning is expected).

Figure 4.3(c) shows how the system was set up, and figure 4.3(d) shows how the threshold voltage of the sense MOSFET (as viewed from the control gate) varied with time during programming (the traces are labelled with the voltage applied between the V_{CTRL} and V_{TUNN} terminals).

Although this design breaks the design rules for most fabrication processes, Thomsen and Brooke reported only 1 failed device out of 100 in their experiments.

4.4.5 Survey of SLV-CMOS Research

Table 4.1 is a summary of papers published prior to January 1995 about the behaviour of SLV-CMOS floating gate devices. As the first test chip (described in the next chapter) was submitted to fabrication at this time, the work published in these papers forms the basis of its design. FNT is particularly important as it is used in all floating

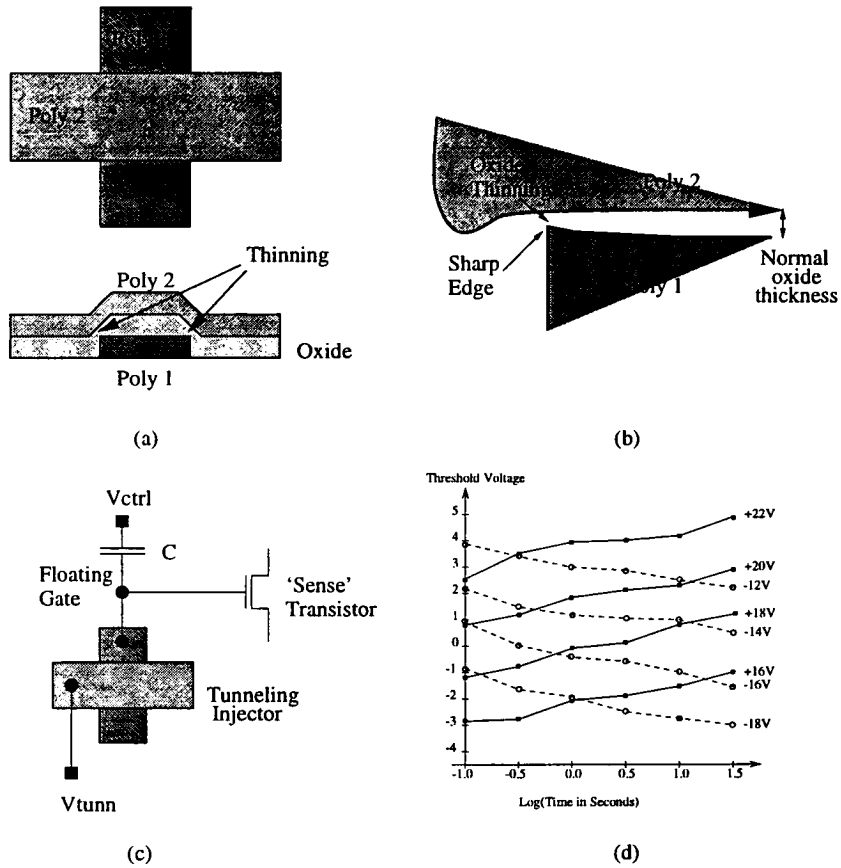


Figure 4.3: (a) Top and cross-sectional sketches of tunnelling injector, (b) Sketch of SEM cross-section showing oxide thinning at edges of poly1, (c) Floating gate system with tunnelling injector, and (d) Threshold shift within the floating gate system

Authors	Process Description	Programming Method	Structure	Suggested Tunnelling Enhancement Mechanism
Carley [21] (1989)	2μm p-well CMOS GOX=40nm	FNT	poly1-diffn	corner effect
Mann [106] (1990)	MOSIS 2μm n-well double-poly CMOS GOX=40nm POX=60-70nm	(a) CHE inject	sense transistor channel	
		(b) CHE/AHE inject	sense transistor channel (depending on length)	
		(c) AHE inject	source-less transistor (gated diode)	
		(a)-(c) FNT remove	poly1-poly2 (underlapped)	
Thomsen & Brooke [161] (1991)	MOSIS 2μm p-well double poly CMOS POX=75nm	FNT	poly1-poly2 (overlapped)	perimeter oxide thinning / sharp surface tips at edges
Durfee & Shoucair [57] (1992)	MOSIS-ORBIT 2μm p-well double poly CMOS GOX=40nm POX=84nm	FNT	poly1-diffn	*no* corner/edge effect
			poly1-poly2 (underlapped)	surface asperities / oxide contaminants
Durfee & Shoucair [58] (1992)	MOSIS 2μm p-well double poly CMOS GOX=40nm POX=69nm	FNT	poly1-poly2 (underlapped)	
			poly1-poly2 (overlapped)	perimeter oxide thinning / corner effect
Montalvo & Paulos [114] (1993)	ORBIT 2μm double poly n-well CMOS	CHE inject	sense transistor channel	
		FNT remove	poly1-poly2 (overlapped)	
Gao & Snelgrove [67] (1994)	1.2μm n-well CMOS (CMOS4S) GOX=24nm POX= 40nm	FNT	poly1-diffn	corner effect
Chai & Johnson [26] (1994)	2μm n-well double poly CMOS POX=75nm	FNT	poly1-poly2 (overlapped)	corner effect (thin tips)

Table 4.1: Summary of SLV-CMOS research papers published before January 1995

gates listed (either bi-directionally or only for electron removal). Authors consistently reported tunnelling potentials much lower than those predicted by the Fowler-Nordheim equation, and their suggestions as to why this should be (where applicable) are listed in the final column; there was clearly no common consensus. Additionally, even where different groups agreed on corner enhancement, they would disagree as to its extent.

In 1992, in two related papers, Durfee and Shoucair [57, 58] attempted to address the same issues; in the first paper they reported significantly lower programming voltages for poly-poly underlapped tunnelling capacitors and negligible corner effects. In the second paper they reported lower programming voltages for overlapped tunnelling capacitors compared to underlapped tunnelling capacitors, with even lower for tunnelling capacitors with many corners.

Structure	Oxide Type	Inject Area	Perimeter	No. of Corners	Capacitance Ratio
		μm^2	μm		C_1/C_2
A	poly-diffn	64	128	32	4/1
B	poly-diffn	64	64	8	4/1
C	poly-poly	180	196	28	4/3
D	poly-poly	182	58	4	2/1

Structure	Overlapping Polysilicon	Inject Area	Perimeter	No. of Corners	Capacitance Ratio
		μm^2	μm		C_1/C_2
1	poly2	150	42	4	4.5/1
2	poly2	208	130	28	3.7/1
3	poly1	144	40	4	5.2/1

Table 4.2: Key Parameters for Durfee and Shoucair's Floating Gate Structures (upper table [57], lower table [58])

The pertinent parameters of their test devices are listed in table 4.2. It would appear that the injection area, the perimeter, the numbers of corners, the capacitance ratios between C and the tunnelling capacitor and the tunnelling capacitor type varies simultaneously between the test structures in the study thus making it difficult to identify the contributions of the various factors; the conclusions of these papers were hence treated with caution. Hence, a need was identified in similar experiments to establish good controls.

Montalvo and Paulos [114] adapted Thomsen and Brooke's idea by replacing the tunnelling injection phase by a CHE injection phase and found this to require lower voltages and shorter programming periods for their given process. For their process, they measured a five magnitude speed increase in programming over Thomsen & Brooke's tunnelling-programmed structure for a 16V gate voltage.

Gao and Snelgrove [67] reported tunnelling enhancement due to corners in poly-diffusion tunnelling capacitors but that it is not proportional to the number of corners. It is worth stating, however, that their gate oxide was only 250 Å wide, whereas Durfee and Shoucair's was 400 Å.

The device geometries were not published and it was difficult to see from the lay-out diagrams if other factors (area, perimeter, capacitance ratios) had been suitably controlled. Interpoly tunnelling capacitors had not been investigated.

Authors	Process Description	Programming Method	Structure	Suggested Tunnelling Enhancement Mechanism
Brown, Collins & Marshall [15] (1995)	Orbit 2μm double poly CMOS	FNT	poly1-poly2 (overlapped)	surface asperities
Diorio, Mahajan, Hasler, Minch & Mead [54] (1995)	2μm Orbit n-well BiCMOS GOX=34.2nm	CHE inject	NMOS transistor with Bi-CMOS layer implant to channel	
		FNT remove	poly1-diffn.	tunnelling at edges†
Lande, Ranjbar, Ismail & Berg [96] (1996)	2μm single poly CMOS	FNT	sense transistor gate with stacked contact	edge effect, *no* corner effect
Diorio, Hasler, Minch & Mead [55] (1997)	2μm Orbit n-well BiCMOS GOX=35nm	CHE inject	NMOS transistor with Bi-CMOS layer implant to channel	
			PMOS transistor	
		FNT remove	poly1-diffn	tunnelling at edges†

Table 4.3: Summary of SLV-CMOS papers published after January 1995. †Diorio *et al* report that tunnelling occurs along the length of gate-to-n^+ edge since the self-aligned n^- region beneath the floating gate is depleted, lowering the effective oxide potential. This is a description of the locus of tunnelling in the structure as opposed to a suggested mechanism for enhancement.

Chai and Johnson [26] attempted to make even sharper corners by defining them as less than the width of their process minimum feature size. This, they found, reduced the required V_{PP} by a couple of volts. However it is unclear if this technique is transferable - these features may simply disappear from other process masks.

Work published from 1995 onwards is summarised in table 4.3[3] Some of this work will be discussed later when the experimental results of the test chip are put into context.

4.5 Discussion

A number of VLSI research groups have taken an interest in SLV-CMOS and have generated almost as many hypotheses about the nature of the floating gate programming behaviour. Tunnelling, hot electron injection and UV-irradiation have been used for programming and both gate and interpoly oxides have been used as isolating mediums. Corners, fringing, thinning, surface asperities and oxide contaminants have been used to explain observed behaviours but the conclusions have been contentious. Of significant note has been the emphasis on programming characteristics. Some authors have attempted bake retention tests but most have taken long analogue storage times in SLV-CMOS devices for granted. There is clearly scope for investigation at this basic device level regarding these properties before considering going forward to build full floating gate parameterised circuits.

[3]Notice that follow-up papers by the same groups of authors about applications, programming schemes or modifications to their floating gate designs are not included in this summary as it is the initial behavioural findings which are of interest here. Some related papers are discussed in the chapter 7.

Chapter 5

TARDIS: A Floating Gate Test Chip

5.1 Introduction

As the first step towards building a non-volatile neural network it was decided to build a floating gate test chip, *TARDIS*[1], which would comprise very simple analogue-mode floating gates. No neural circuitry would be included on this first chip so to allow quick and simple design and test.

5.1.1 Design Motivations

The motivations for the design of the test chip were severalfold:

- Evaluate the appropriateness of the available fabrication technology (Mietec 2.4μm N-well double-polysilicon) for the implementation of floating gates. Previous reported work has only been in MOSIS compatible technologies.
- Gain practical design, test and programming experience in a simple context prior to applying floating gates to more complicated integrated neural systems where individual factors may be difficult to isolate.
- Parameterise the floating gate and programming behaviour for use in developing models for the design of subsequent systems.
- Investigate issues of contention in the literature identified in tables 4.1 and 4.3: the best oxide to use for tunnelling, individual geometric factors (ie. the influence of corners and perimeter effects and edge thinning), and contribute to the body of knowledge regarding the behaviour of these devices by developing test structures with proper controls.

A further aim was to isolate the most efficient mechanism for programming of those discussed in section 3.5.

[1] *TARDIS*: Tunnelling Analogue Reference Devices In Silicon

5.1.2 Design Constraints

As *TARDIS* would rely on undocumented features of the fabrication process it was not possible to know at design time the likelihood of success. It was therefore undesirable to dedicate a large amount of the silicon budget to this investigation. Additionally since nothing was assumed about the device operation (particularly programming voltages) it was necessary to avoid an addressing scheme and to have each test element directly pinned out in isolation from one another (special pads were designed for this purpose which did not contain any other circuits which may interfere with device operation at high voltages[2]).

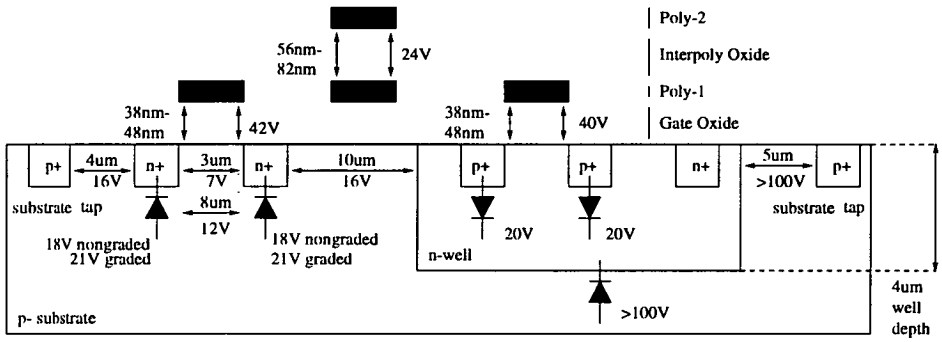


Figure 5.1: Pertinent electrical parameters of the Mietec 2.4μm process [46]. In Alcatel-Mietec 2.4μm CMOS layout is drawn to 3μm rules and then scaled by 0.8 during mask making. (For example a minimum length transistor channel is drawn as 3.0μm but is scaled to 2.4μm). The convention in this thesis is to describe lengths and widths of devices in their drawn (ie.3μm) dimensions but to use the post-scaled values for capacitances.

Thus the number of test structures had to be kept small. For this reason they were designed to accommodate three types of experiment:

1. Interpoly Tunnelling - tunnelling capacitor structures were required. Since the body of research preceding this work (table 4.1) strongly favoured interpoly tunnelling capacitors to allow the lowest tunnelling voltages, the *TARDIS* test structures were strongly weighted in preference of investigating the edge/corners properties of these types of structure.

2. Gate Oxide Tunnelling - instead of a separate tunnelling capacitor, experiments with gate oxide used the oxide of the sense transistor. Since interpoly tunnelling was the principal subject of investigation, multiple gate oxide structures were not desired.

[2]Voltage handling capabilities of the Mietec fabrication process are shown in figure 5.1.

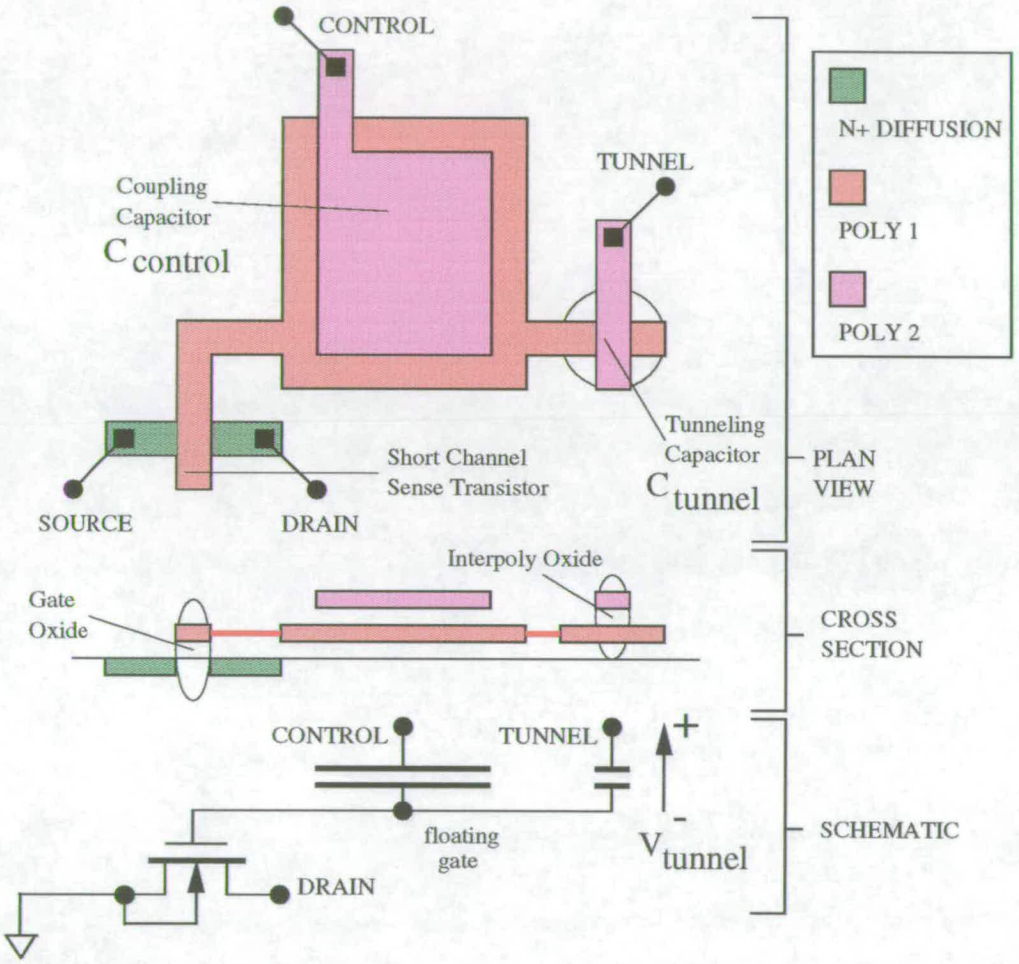


Figure 5.2: Generic structure (plan, cross section and schematic) of floating gate cell

3. Hot Electron Injection - to facilitate the high drain-source electric field required by CHE, the sense transistor was made to be minimum length (2.4μm). The sense transistor as part of a system is unlikely to be of minimum length.

To further minimise the size and scope of *TARDIS* two programming mechanisms were entirely discounted at design time:

1. Ultra-Violet: The principal disadvantage with UV is the need for a special lamp; clearly this limits integration and a wholly electrical system was desired. Additionally programming can be slow and addressing can be problematic since a conducting path will be formed across *every* exposed oxide insulation not just the floating gate, therefore requiring the use of shielding.

2. Avalanche Injection: Previous work [139] has shown CHE to be the dominant hot electron process at low temperatures, and therefore the best candidate for hot electron based programming.

The chosen generic structure of a floating gate is therefore that shown in figure 5.2. The tunnelling potential (across the tunnelling capacitor), V_{tunnel}, is defined as:

$$V_{tunnel} = V_{PP_{tunnel}} - V_{fg} \tag{5.1}$$

The definition of the field-emission tunnelling threshold, V_{fe}, is an ambiguous one since the tunnelling currents are an extremely non-linear function of the applied tunnelling voltages and the current can vary over several orders of magnitude. Therefore defining the tunnelling threshold the onset of 'significant tunnelling current' is unsatisfactory. For this project we define V_{fe} as *the initial value of* V_{tunnel} *required to cause a shift in* V_{fg} *above a precision boundary,* γ*, during a period of 100ms.* The *initial* value of V_{tunnel} is specified since V_{tunnel} will decay during the 100ms period as electrons are injected/removed. The precision boundary constraint, γ, is used simply to ensure that the movement in V_{fg} observed is due to re-programming, not noise. This limit is therefore set (provided it is above the precision boundary it is fairly arbitrary) in the context of the relevant experiment.

5.1.3 Layout of Tunnelling Capacitors

The various layout designs for the tunnelling capacitors implemented on TARDIS are shown in figure 5.1.3. The control terminal was coupled in through a capacitor identified as $C_{control}$ and the capacitance of the tunnelling capacitor was identified as C_{tunnel}.

The geometric properties of the test structures are shown in table 5.1. The figures in this table, and henceforth, shall refer to the the post-shrink geometries. The capacitance ratio is of $C_{control}$:C_{tunnel} layout ratio.

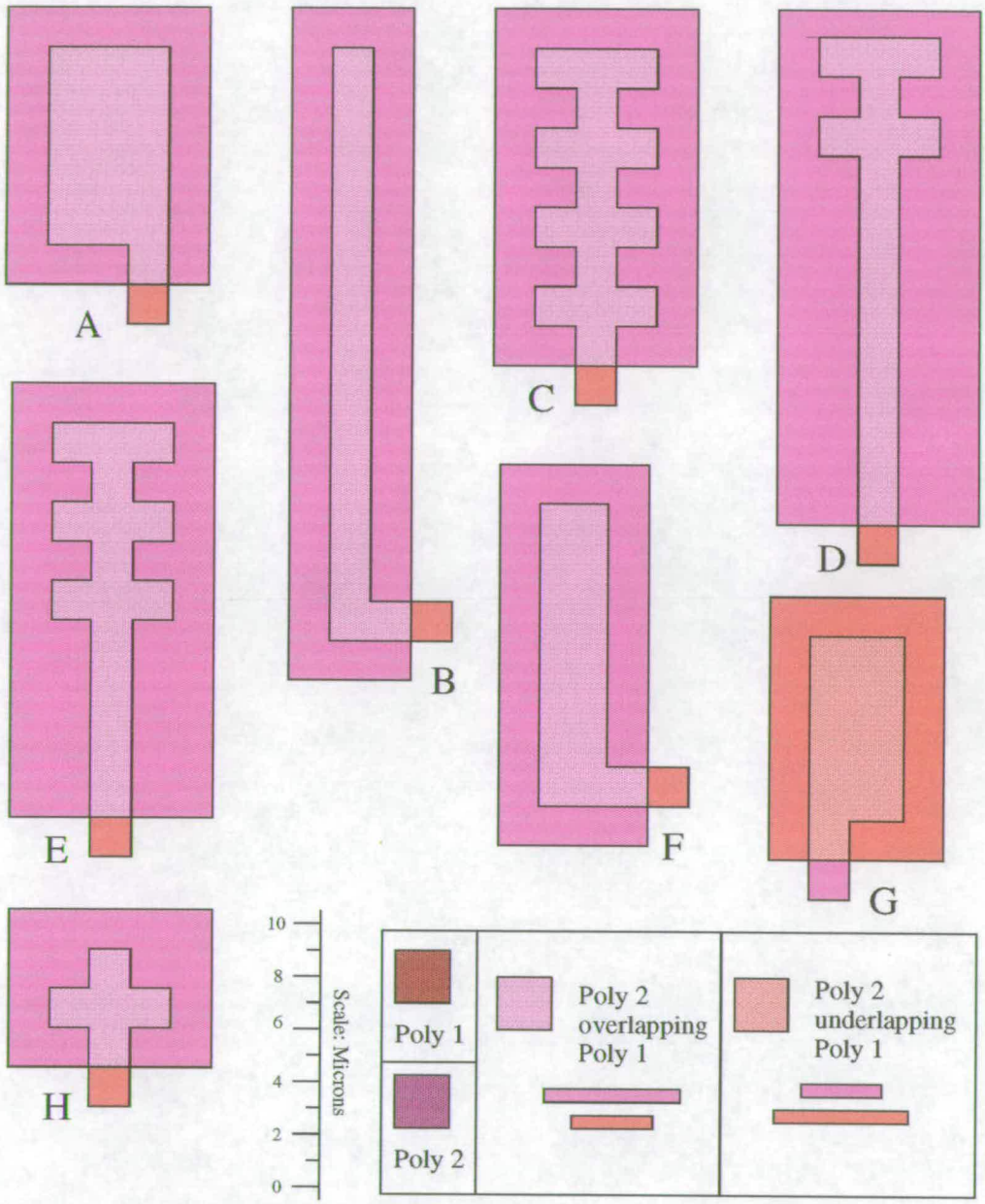


Figure 5.3: Layout of tunnelling capacitor structures. In Alcatel-Mietec 2.4μm CMOS, layout is drawn to 3μm rules and then scaled by 0.8 during mask making. (For example a minimum length transistor channel is drawn as 3.0μm but is scaled to 2.4μm). The layouts shown here are the drawn (pre-shrink) sizes.

Device Name	poly2 overlap	area of overlap	perimeter of overlap †	no. of corners ††	capacitor ratio
A	yes	92.16μm^2	40.8μm	3	10:1
B	yes	92.16μm^2	79.2μm	3	10:1
C	yes	92.16μm^2	79.2μm	16	10:1
D	yes	92.16μm^2	79.2μm	8	10:1
E	yes	92.16μm^2	79.2μm	12	10:1
F	yes	92.16μm^2	50.4μm	3	10:1
G	no	92.16μm^2	40.8μm	3	10:1
H	yes	28.80μm^2	26.4μm	6	32.2:1

Table 5.1: Geometric properties of the test devices. (†Does not include the 2.4μm perimeter at the point where poly1 (poly2 in G) is connected into the structure, ††Indicates only sharp convex corners). Area and perimeter values are post-shrink.

The floating gate capacitors are naturally large in *TARDIS* because of the large feature size of the Alcatel-Mietec 2.4μm process. To incorporate all the perimeter and corner variations detailed in table 5.1, this results in a minimum tunnelling capacitor of 46fF according to ideal layout/oxide characteristics. To implement a reasonable capacitive coupling ratio of 10:1, this requires $C_{control} \sim$ 460fF. Although such a large capacitor is layout area inefficient, it acts as a large storage capacitor, mitigating some of the undesirable second order parasitic effects associated with floating gates as will be seen later.

Type G tunnelling capacitors are the control underlapped devices. In Alcatel-Mietec 2.4μm CMOS, it is not allowed to contact poly2 over interpoly oxide[3]. Therefore a poly2-poly1 overlap is inevitable since poly2 must run off the capacitor to be contacted; this overlap has been minimised (as shown in figure 5.1.3) but cannot be eliminated[4].

Type H devices did not form part of the test set but were merely simple, compact, multi-cornered devices which, on the basis of SLV-CMOS floating gate literature, were proposed as a good compromise between size and injection efficiency.

[3]This is to prevent additional stresses and contact etch damage to poly2 and to prevent metallisation consuming the poly. Aluminium reacts with silicon and some silicon from the contacts migrates out into the aluminium during thermal processing.

[4]The reliability of this overlap must be addressed by the process engineer rather than the layout designer although, anecdotally, modern processes hardly tend to discourage any structures because of step coverage issues. Some do offer the option of thick oxide *between* poly1 and poly2 and allow the contact to be drawn on top of that area. This is not permitted in Alcatel-Mietec 2.4μm CMOS.

5.2 Experiments with Tunnelling

When *TARDIS* was returned from fabrication, the $I_d - V_d - V_{CONTROL}$ characteristics of various devices were examined. Between measurements, V_{TUNNEL} was set to various large (+/-) values while $V_{CONTROL} = 0$V. As can be seen in figure 5.4 this had the influence of changing the I_d characteristics of the floating gate cells. This suggested that the floating gates had been successfully fabricated and were programmable [5].

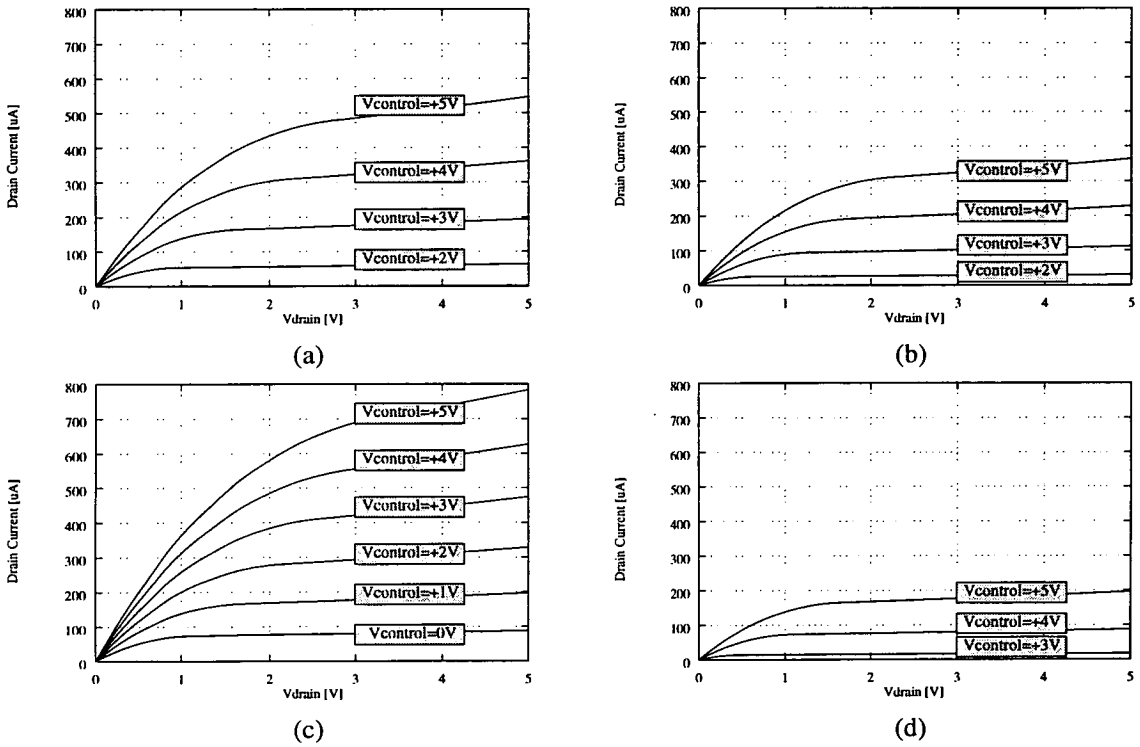


Figure 5.4: (a) Measured $I_d - V_d - V_{gate}$ characteristics for an NMOS reference transistor of equivalent layout to floating gate sense transistor. Measured $I_d - V_d - V_{CONTROL}$ characteristics for a floating gate transistor with Vso (b) set approximately equal to V_{T_n} (exhibiting the capacitive division of $V_{CONTROL}$ onto the gate), (c) set arbitrarily negative and (d) set arbitrarily positive. $V_{TUNNEL} = 0$V during measurement.

[5]Initial concern that charge build up on the isolated gate *during* fabrication would damage the oxide appeared to be unfounded. EEPROM fabrication processes tend to use radiation shields in plasma etch and deposition equipment to prevent this. All floating gates appeared to have encountered a build up of holes (rather than electrons) during fabrication with about 2-3V on the floating gate when initially turned on.

Measurement of Switch-On Voltage

Here we define the *switch-on* voltage, V_{SO}, of the sense transistor as the $V_{CONTROL}$ required to establish $I_d = 1.0\mu$A; this is closely analogous to the threshold, V_{T_n} of a normal transistor (but easier to measure[6]). In these experiments V_{SO} is measured with $V_d = 2.0$V; measurement is by a modified binary search algorithm: $V_{CONTROL}$ is initially set to range between two small seed values, $V^-_{CONTROL} < 0$ and $V^+_{CONTROL} > 0$. If $I_d\left(V^-_{CONTROL}\right) \leq 1.0\mu$A and $I_d\left(V^+_{CONTROL}\right) \geq 1.0\mu$A, resolution of $V_{CONTROL}$ proceeds by binary search, otherwise the appropriate seed is incremented/decremented until the range is correct. In this way the entire V_{SO} range can be measured without inadvertently reprogramming the floating gate by applying a large wrong-polarity signal to the `control` terminal.

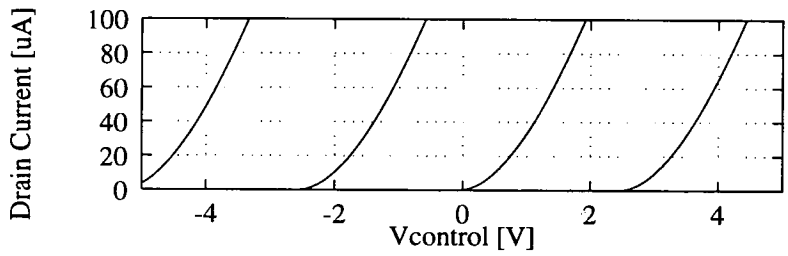


Figure 5.5: Measured $I_d - V_{CONTROL}$ characteristics for a range of programmed V_{SO}s. $V_d = 2$V.

Figure 5.5 shows the $I_d - V_{CONTROL}$ characteristics of a device programmed to a range of different V_{SO} values, illustrating how the programming process can be considered closely analogous to changing the threshold voltage of a conventional transistor.

Initial experiments involved re-measurement of the characteristics after devices had been left unpowered overnight and the changes in I_d characteristics appeared to be unaltered. It was therefore concluded that it *was* possible to program floating gates in the Alcatel-Mietec process non-destructively, and to a first examination they were non-volatile. This was a significant outcome.

Controllability of Programming

A second issue of concern was that of controllability; in previous work with amorphous silicon (a-Si:H) resistors [80], the response to a series of programming pulses was often

[6]One empirical definition of threshold involves calculation of the x-axis crossing of the least square linear regression of the $I_d - V_{gs}$ curve with V_{drain} = 100mV [46].

found to be erratic with jumps of resistance, sometimes in the "wrong" direction, in response to a single polarity series of programming ramps.

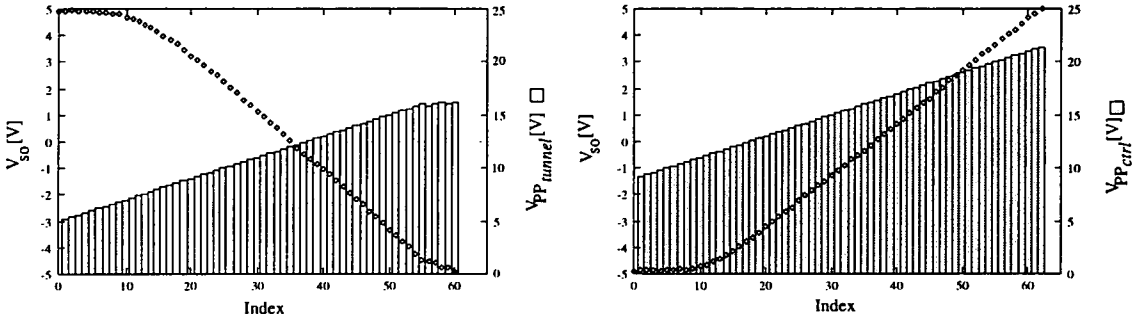


Figure 5.6: (a) Initialising V_{SO} to -5V with positive pulses applied to the tunnel terminal, and (b) Measurement of V_{SO} while applying a constantly increasing pulse stream to the control terminal (NB. Equivalentally, a negative pulse stream could be applied to the `tunnel` terminal). Duration of programming pulses = 100ms.

However programming of *TARDIS* floating gates was always found to be consistent. A positive potential across the tunnelling capacitor either produced no change in V_{SO} ($0 < V_{tunnel} < V_{fe}^{+}$) or reduced it by removing electrons from the floating gate. Conversely, a negative potential across the tunnelling capacitor either produced no change in V_{SO} ($V_{fe}^{-} < V_{tunnel} < 0$) or increased it by injecting electrons on the floating gate. This is illustrated by a typical experiment shown in figure 5.6 (Here boxes represent programming pulses of 100ms duration, points represent measured values of V_{SO}). In this experiment programming pulses were ramped up by 200mV after each measurement (unless the distance to the initialisation target value was less than ΔV_{SO} induced by the preceding pulse, in which case pulses were ramped down by 200mV instead; this was to slow programming near the target value and so prevent overshoot). There were no reset pulses between the programming pulses. A simple *RC* filter was connected to the programming input lines to smooth the pulse roll-on and roll-off edges, so to reduce detrimental oxide stress [185]. Rise/fall times were consequently $\sim$ 1ms.

Note in figure 5.6(b), an alternative approach to generating negative V_{tunnel} potentials is used: instead of driving V_{TUNNEL} negative ($V_{PP_{tunnel}}$), V_{TUNNEL} = 0V and a large potential is coupled onto the floating gate by driving $V_{CONTROL}$ high ($V_{PP_{control}}$). This permits programming using only positive signals.

5.2.1 Voltage and Time Programming Dependency of Gate and Interpoly Oxides

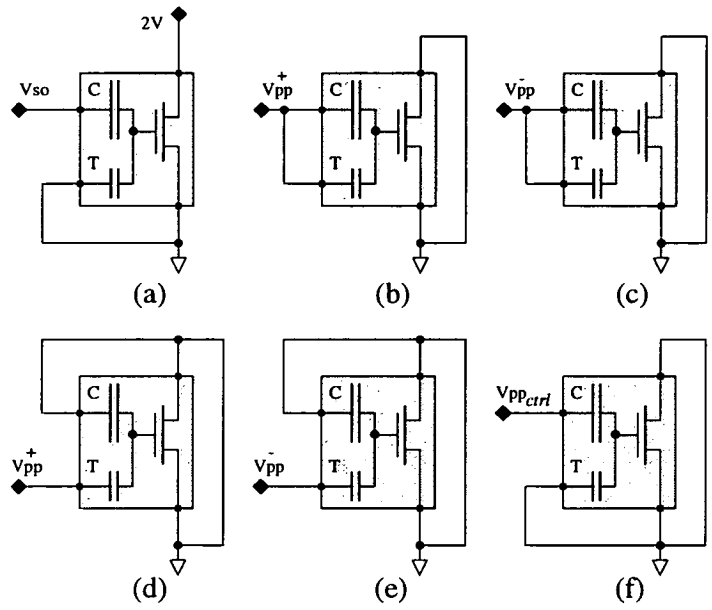


Figure 5.7: Fowler-Nordheim programming configurations for floating gate device: (a) READ mode - measurement of V_{SO}, (b) Electron INJECTION mode through gate oxide of sense transistor by coupling high positive voltage onto floating gate, (c) Electron REMOVAL mode through gate oxide of sense transistor by coupling high negative voltage onto floating gate, (d) Electron REMOVAL mode through interpoly oxide by direct application of high positive V_{tunnel}, (e) Electron INJECTION mode *(i)* through interpoly oxide by direct application of high negative V_{tunnel}, (f) Electron INJECTION mode *(ii)* though interpoly oxide by coupling high voltage onto the floating gate from CONTROL terminal.

Figure 5.7 shows the various signal configurations required to program the floating gate, either through the gate oxide or through the tunnelling capacitor. These configurations were established using a set of PC-controlled relays (to handle the high voltages) and programming pulses generated by a pulse generator connected to amplifiers. The test board for *TARDIS* is described in more detail in section B.2.

The next set of experiments were designed to investigate the necessary programming voltage magnitudes ($|V_{PP}|$) for the two oxide types.

The first experiment was on the tunnelling in the sense transistor gate oxide. Prior to a positive programming signal (figure 5.7(b)), V_{SO} was programmed iteratively to -4V, and prior to a negative programming signal (figure 5.7(c)), V_{SO} was programmed iteratively to +4V. Programming signals in the ranges +10V $\rightarrow$ +34V and -10V $\rightarrow$ -34V

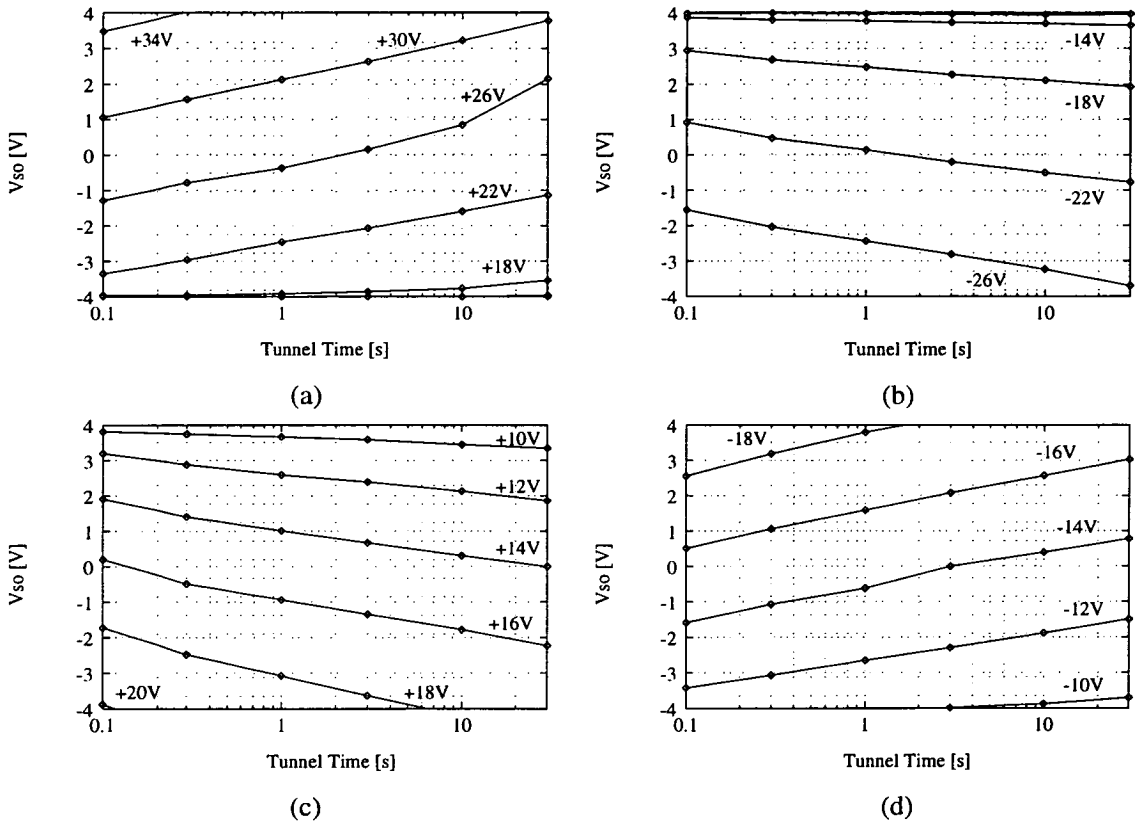


Figure 5.8: Top: Typical V_{SO} programming characteristics for tunnelling through the gate oxide of the sense transistor. (a) $V_{SO}(0) = -4$V (positive pulses), (b) $V_{SO}(0) = +4$V (negative pulses). Bottom:Typical V_{SO} programming characteristics for tunnelling through the interpoly oxide of the tunnelling capacitor. (c) $V_{SO}(0) = +4$V (positive pulses), (d) $V_{SO}(0) = -4$V (negative pulses). The experiments were run in increasing $|V_{PP}|$ order with a reset to $V_{SO}(0)$ between each experiment. (Note that the reset voltage at $V_{SO}(0)$ is not shown on the graphs because of the logarithmic x-axis)

were applied to the COUPLE and TUNNEL terminals for a total period of 30 seconds; V_{SO} was evaluated at various instants within that period as shown (figure 5.7(a)). A typical experimental trace is shown in figures 5.8(a) and 5.8(b).

A similar set of experiments were carried out on the tunnelling capacitor structures (figure 5.7(d)-(e)). A typical experiment is shown in figures 5.8(c) and 5.8(d).

Two significant observations could be drawn from these experiments:

- ΔV_{SO} exhibits an approximate logarithmic time dependency in both cases. This effect was mentioned in section 3.5.5. The tunnelling potential, V_{tunnel}, depends on the potential difference between the TUNNEL terminal and the floating gate:

$$V_{tunnel} = V_{PP} - V_{fg} \tag{5.2}$$

  Assuming removal of electrons, V_{fg} rises and so V_{tunnel} falls, exponentially reducing the rate of electron removal (due to the FNT dependency of tunnelling current on tunnelling potential). In this way programming is self-limiting; and $\Delta\phi_{fg}$ becomes logarithmic with time. The theoretical form of this behaviour is derived in section 5.5.1.

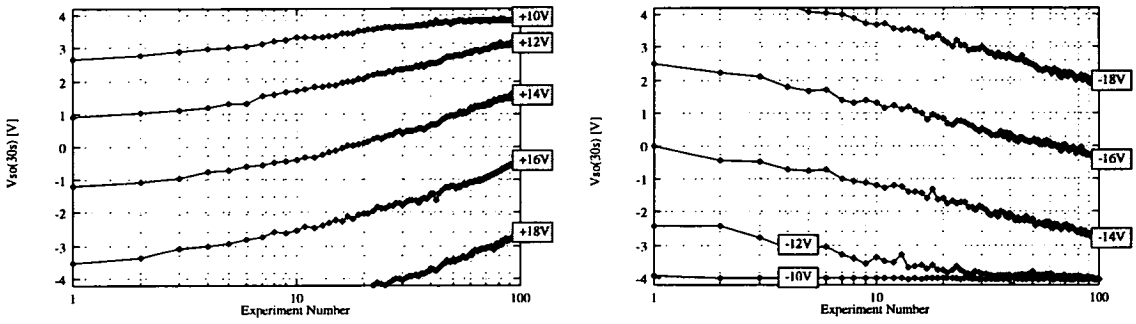
- Programming (hence tunnelling) occurs at significantly lower voltages in the interpoly oxide than in the gate oxide despite its greater thickness.

The lower $V_{fe}^{\pm}$ of the interpoly oxide gate oxide is quite a striking result although it was expected from the reported experiments on MOSIS-compatible structures. Despite the relative thickness of the interpoly oxide, the tunnelling threshold *has* been reduced below that of the gate oxide by some feature of the layout or physical process.

5.2.2 Reprogrammability and Device Aging

Oxide "aging" is a significant problem in floating gate devices. Electrons travelling through the oxide can become trapped and cause localised distortions of the electric field (see section D.2). These trapped electrons can introduce a repulsive field which counteracts the programming field. This is not in itself a failure mechanism, since the programming voltage can be increased to recover the field required for programming. However, wear-out limits the extent to which this can be repeated without damaging the oxide, and so devices can eventually become unprogrammable.

Figure 5.9 shows the non-reproducibility of repeated cycling of the magnitude and duration experiment on a typical interpoly tunnelling capacitor. The figure shows that $\Delta V_{SO}(30s)$, ie. $V_{SO}(30s) - V_{SO}(0s)$, declines logarithmically with programming cycles, indicative of early 'aging' followed by a decline in the aging rate. All experiments with the *TARDIS* floating gates were intrusive and non-reproducible for this reason.



(a) (b)

Figure 5.9: V_{SO} after 30s tunnelling time for (a) $V_{SO}(0) = +4$V and positive pulses, (b) $V_{SO}(0) = -4$V and negative pulses

5.2.3 Interpoly Tunnelling Capacitor Design Comparison

Having confirmed the reported lower tunnelling potentials in interpoly tunnelling capacitors, the next stage was to determine the influence of poly2 overlaps, corners and extended perimeters using the various interpoly tunnelling structures shown in figure 5.1.3. This was approached by applying a series of experiments to determine the field-emission tunnelling threshold, V_{fe}, for each of the tunnelling capacitor designs.

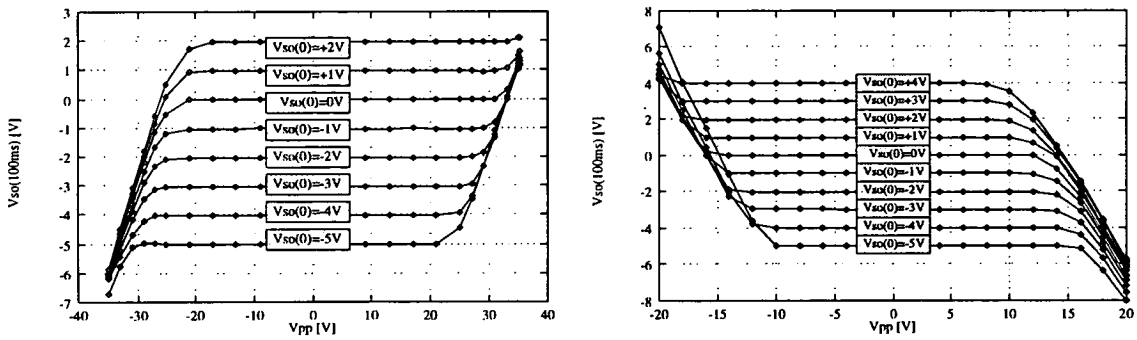
In these experiments, each tunnelling capacitor was characterised by initialisation of V_{SO} (to $V_{SO}(0)$), followed by the application of a single 100ms pulse and re-measuring V_{SO}. This experiment was repeated for a range of $V_{SO}(0)$ and V_{PP} values until a picture of programming behaviour was built up for each test device (typical results shown in figure 5.10(a) for gate oxide tunnelling and (b) for interpoly oxide tunnelling; some device aging is inevitable during the experiment and this is visible as irregularities within the graph).

To convert these data sets into V_{fe} measurements, a simple lumped capacitor model was developed.

5.2.4 Capacitor Network Model

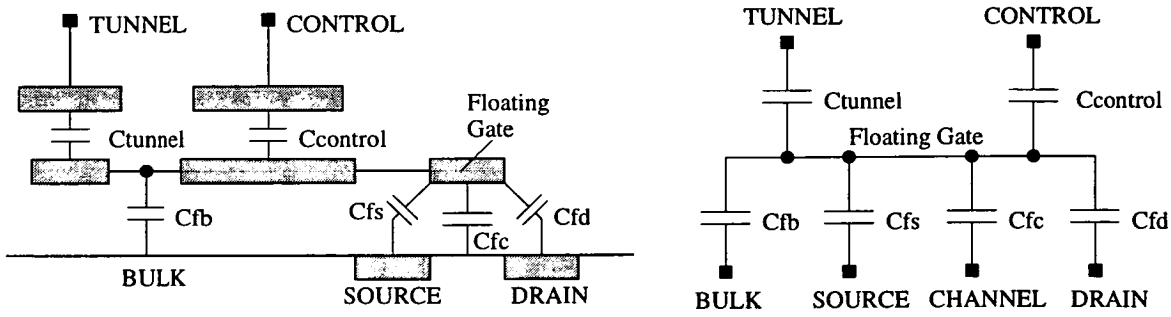
Figure 5.11 shows the capacitor network structure for *TARDIS*. Although exact capacitances were not determined, typical values can be derived from published capacitance per areas and device layout geometries. Since the resistivity of the heavily doped polysilicon floating gate is small, it can be treated as an equipotential surface.

Typical plate capacitances are then C_{tunnel} = 46fF, $C_{control}$ = 464fF and C_{fb} = 60fF. C_{fs},C_{fc} and C_{fd} are voltage dependent capacitances but sum, in the worst case, to about 10fF. For simplicity, then, it was chosen simply to add 10fF to C_{fb} and ignore the MOSFET capacitances due to their comparatively small size. The bulk and source



(a) (b)

Figure 5.10: Typical programming characterisation experiment on (a) Tunnelling in the gate oxide of the sense transistor, (b) Tunnelling in the interpoly oxide of the tunnelling capacitor



(a) (b)

Figure 5.11: (a) Schematic structure of floating gate cell illustrating the lumped capacitors to be used in the capacitor network model. (b) The derived capacitor network model

voltages were always zero. Fringing capacitances were ignored as they were about two orders of magnitude less than the plate capacitances. Cross-chip variations in oxide thicknesses and geometrical deviations have also been ignored. Finally, the relation between the capacitance of overlapped tunnelling capacitors and equivalent area interpoly capacitors is unknown and has been assumed unity.

Given these assumptions the derived model must be treated only qualitatively; fitting parameters must account not only for tunnelling behaviour but also for deviations from typical capacitances.

The total capacitance of the floating gate, C_{fg} is the sum of all capacitances, and so is calculated as 580fF.

Data Transformation

V_{fe} is the minimum V_{tunnel} such that

$$|V_{SO}(0) - V_{SO}(t_{PP})| > \gamma \tag{5.3}$$

For this experiment, γ was set at 50mV.

To determine V_{fe} it was first necessary to reduce $(V_{SO}(0), V_{PP})$ pairs into the direct parameter, V_{tunnel}.

By superposition of the input voltages and the trapped charge:

$$V_{fg} = \left(\sum_{j}^{capacitors} \frac{C_j}{C_{fg}} V_j \right) + \phi_{fg} \tag{5.4}$$

Therefore during readout, when the switch-on condition is met, $V_{CONTROL} = V_{SO}$ and $V_{TUNNEL} = 0V$ equation 5.4 evaluates as

$$V_{fg}(ON) = \frac{C_{control}}{C_{fg}} V_{SO} + \phi_{fg} \tag{5.5}$$

where $V_{fg}(ON)$ is the voltage required on the floating gate such that the switch-on condition is exactly met. Defining the capacitor ratio as $m_c = C_{control}/C_{fg}$, this can be rewritten as

$$\phi_{fg} = V_{fg}(ON) - m_c V_{SO} \tag{5.6}$$

From experiments, it is easy to produce a database of items of the format:

$$\{V_{SO}(0), V_{PP}, t_{PP}, V_{SO}(V_{PP}, t_{PP})\} \tag{5.7}$$

which define the switch-on condition of a device before and after a programming pulse of duration t_{PP}. Using equation 5.6, this can be mapped into the form:

$$\{\phi_{fg}(0), V_{PP}, t_{PP}, \phi_{fg}(V_{PP}, t_{PP})\} \tag{5.8}$$

Now, during programming, when $V_{TUNNEL} = V_{PP}$ and $V_{CONTROL} = 0V$, equation 5.4 evaluates as

$$V_{fg} = \frac{C_{tunnel}}{C_{fg}} V_{PP} + \phi_{fg} \tag{5.9}$$

This may be expanded to incorporate the time dependency of V_{fg} and ϕ_{fg} during programming (with the capacitor ratio, (C_{tunnel}/C_{fg}) defined as m_t):

$$V_{fg}(t) = m_t V_{PP} + \phi_{fg}(t) \tag{5.10}$$

Note that V_{PP} is assumed to be constant for the duration of programming since during experiments the $\sim$1ms rise and fall times of the pulses are typically $\ll t_{PP}$. The model therefore assumes the rise and fall times to be instantaneous.

Now, we can write

$$\begin{aligned} V_{tunnel}(t) &= V_{PP} - (m_t V_{PP} + \phi_{fg}(t)) \\ &= V_{PP}(1 - m_t) - \phi_{fg}(t) \\ &= V_{PP}(1 - m_t) - V_{fg}(ON) + m_c V_{SO}(t) \end{aligned} \tag{5.11}$$

And we can therefore map the triples of form 5.8 into the final form

$$\{V_{tunnel}(0), t_{PP}, V_{tunnel}(V_{PP}, t_{PP})\} \tag{5.12}$$

The element V_{PP} has been omitted as it is implicit in the calculation of V_{tunnel}. t_{PP} is fixed at 100ms. Equation 5.3 can now be applied to determine both V_{fe}^{+} and V_{fe}^{-}.

Processing of Experimental Data

Some typical data processed using the above capacitor network model is shown in figure 5.12; values of $V_{fe}^{\pm}$ were calculated from this plot as the threshold of the ΔV_{SO} characteristic.

The results of these experiments are summarised in table 5.2 with some typical ΔV_{SO} curves plotted in figure 5.13. The following are some observations made during these experiments.

- **Tunnelling Thresholds:** Using the ideal layout values and the Fowler-Nordheim

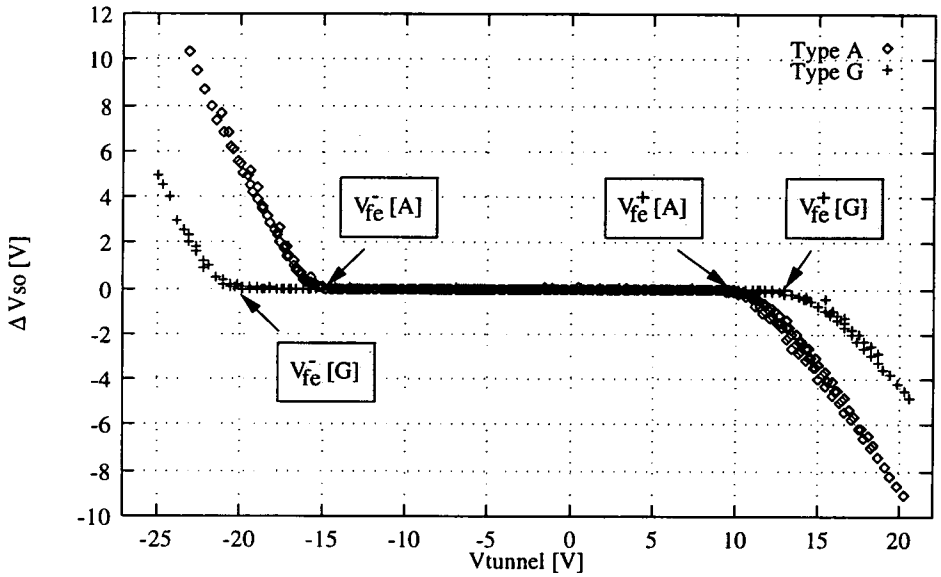
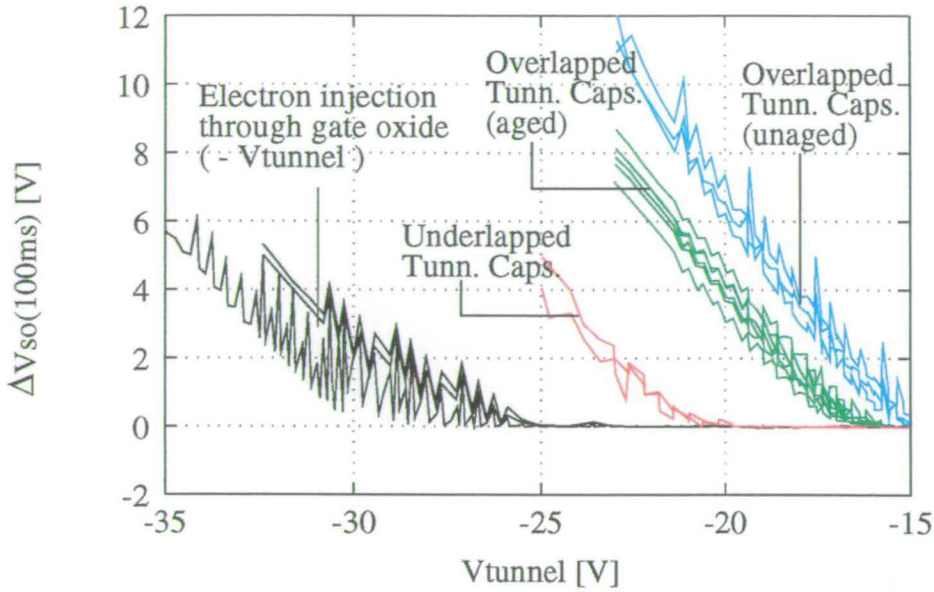


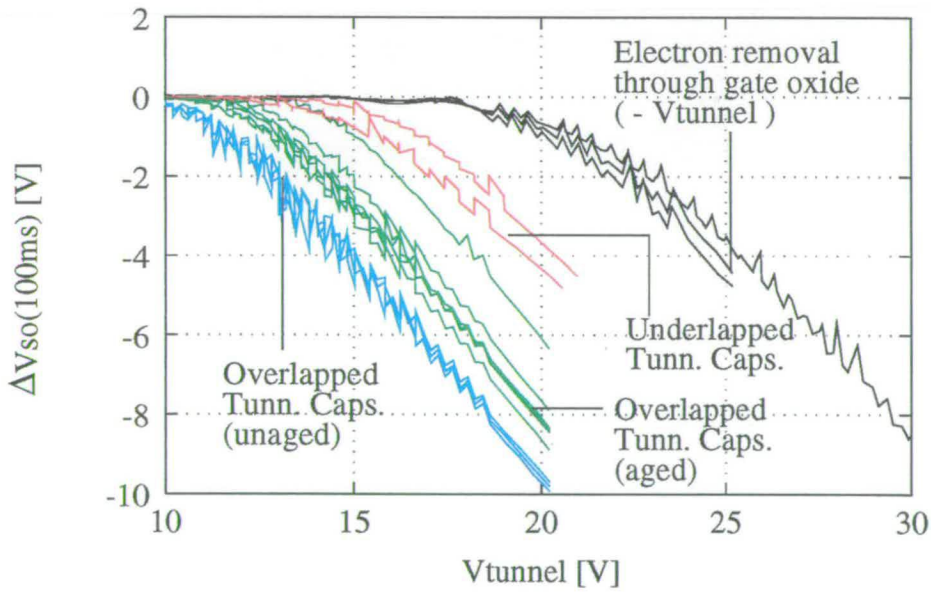
Figure 5.12: Typical ΔV_{SO} versus V_{tunnel} graphs using the capacitor network model to transform experimental data, for an overlapped tunnelling capacitor (layout type A) and geometrically equivalent underlapped tunnelling capacitor (layout type G).

Tunn. Cap. Design	no. samples	Initial Experiment				Third Experiment			
		V_{fe}^{+}	$s_{V_{fe}^{+}}$	V_{fe}^{-}	$s_{V_{fe}^{-}}$	V_{fe}^{+}	$s_{V_{fe}^{+}}$	V_{fe}^{-}	$s_{V_{fe}^{-}}$
A	9	11.6V	0.74V	-14.9V	0.91V	13.4V	0.54V	-16.6V	0.76V
F	10	11.3V	0.59V	-15.8V	1.83V	13.2V	0.47V	-17.3V	1.80V
B	9	11.4V	0.73V	-15.5V	1.41V	13.2V	0.28V	-16.9V	1.19V
D	8	11.3V	0.77V	-15.2V	1.08V	13.2V	0.36V	-16.4V	0.92V
E	9	11.2V	0.93V	-15.5V	1.38V	13.4V	0.89V	-16.8V	1.19V
C	7	11.2V	0.61V	-14.8V	1.08V	13.0V	0.45V	-16.4V	0.92V
G	8	13.3V	0.98V	-17.8V	2.61V	15.3V	0.99V	-21.3V	1.02V
Gate Oxide	6	17.7V	2.05V	-24.8V	1.31V	19.0V	1.50V	-24.5V	1.50V

Table 5.2: Mean, $\overline{V_{fe}}$ and standard deviation, $s_{V_{fe}}$, of field emission thresholds for V_{SO} programming experiments for t_{PP} = 100ms. The initial experiment refers to unaged devices (the first trial of the V_{SO} programming experiment); the third experiment refers to devices which have been aged to a fixed extent (the third iteration of the V_{SO} programming experiment). Devices A,F and B represent increasing perimeters: 40.8μm, 50.4μm and 79.2μm. Devices A,D,E and C represent increasing numbers of corners: 3,8,12 and 16. Device G is the underlapped tunnelling capacitor.



(a)



(b)

Figure 5.13: Selection of typical results for ΔV_{SO} versus V_{tunnel} for (a) V_{PP}^{-}, and (b) V_{PP}^{+}. Tunnelling capacitors of types A1-A3 and B1-B4 have been grouped together as the class of 'overlapped tunnelling capacitors'. Only a representative subset of experiments is shown for clarity. Injection through the gate oxide has been displayed with a reversed x-axis (ie. - V_{tunnel}) so that the polarity of all ΔV_{SO}'s are in agreement; notice that this disparity is due to interpoly tunnelling being referenced from the TUNNEL terminal and gate oxide tunnelling being referenced from the floating gate.

coefficients published in [184], then - for a simple analysis - a $V_{fe}(interpoly) > 46\text{V}$[7] and $V_{fe}(gateox) > 33\text{V}$ should be expected (in both cases symmetrical tunnelling): it is clear that the tunnelling threshold has been reduced by some effect in both cases, especially for the interpoly oxide.

- **Matching:** Tunnelling was not only asymmetric: $|V_{fe}^{+}| \neq |V_{fe}^{-}|$ but there were large variations between notionally matched devices.

- **Overlapped vs. Underlapped Tunnelling Capacitors:** Underlapped tunnelling capacitors were found to have higher V_{fe} than overlapped tunnelling capacitors, particularly where electrons are emitted from poly2 (electron injection *onto* the floating gate - thus the enhancement is asymmetric.

- **Interpoly Oxide vs. Gate Oxide:** All interpoly tunnelling capacitors had considerably lower V_{fe} than for tunnelling through the gate oxide of the sense transistor.

- **Oxide aging:** From table 5.2 it can be seen that the gate oxide ages slightly less than the interpoly oxide over the course of the first three experiments. The gate oxide is a high quality thermal oxide compared to the interpoly oxide on the irregular polysilicon surface. The irregularity of the surface in particular may be reason for a higher density of interface traps in the interpoly oxide and subsequently faster aging.

- **Lithographic Enhancement in Tunnelling Capacitors:** *no feature* – periphery or number of corners – was found to be markedly significant, other than the consistently higher $|V_{PP}^{-}|$ required for underlapped tunnelling capacitors. Variations in V_{fe} between different instances of the same tunnelling capacitor were found to be as severe as between different designs, and aging was found to result in a considerable shift in V_{fe} after cycling.

The last point is perhaps the most unexpected. It shows that tunnelling is *not* being enhanced by lithographic features of periphery or number of corners. Additionally it strongly suggests that field enhancement along edges and at corners is *not* the reason for the observed field emission threshold reduction. The high inter-device variations would point rather to surface asperity-dominated enhancements (ie. electric field enhancement at sharp peaks on a rough surface[8]) as proposed by Brown, Collins and

[7] This is above the dielectric breakdown voltage.

[8] Although the interpoly oxide, like the gate oxide, is thermally grown, the random orientation of the grains in polycrystalline silicon causes a surface roughening to occur [156].

Marshall [15] from their study of continuous-time trapping behaviour[9]. The asymmetry between tunnelling polarities could also be explained in this way since the poly1 top plate will not have identical surface topology to the poly2 bottom plate.

Other oxide irregularities (dust particles, crystalline defects or localised thin regions) or metal contamination (causing energy barrier lowering) can also lead to an observed electric field acceleration by effective oxide thinning [100] at the 'weak spots'. Such mechanisms could also lead to the observed enhancement behaviour with influence also degraded by oxide trapping. While the asperity explanation would appear the simplest and most consistent, and would relate to the generally accepted irregular surface topology of polysilicon, this explanation can by no means be said to be proven by the results of *TARDIS* characterisation.

There is a clear V_{fe} reduction for *overlapped* tunnelling capacitors and accepting Thomsen & Brooke's explanation of thinning along the edges would be somewhat in conflict with the asperities interpretation. Marshall believes that the oxidation process may lead to asperities along the edges rather than wholesale thinning [10,11,12].

Of specific interest is the fact that whilst the perimeter length in overlapped tunnelling capacitors would appear insignificant, the minimum length overlap in the underlapped tunnelling capacitors where poly2 leaves the capacitor is insufficient to cause an equivalent $|V_{fe}|$ reduction. One possibility is that sharp edge asperities or cusps are actually quite rare and the perimeter must be above a certain threshold length to attain a good probability of one existing.

Unfortunately the data from *TARDIS* would appear to raise as many questions that it answers; the edge thinning results are particularly difficult to interpret. Perhaps a more promising approach to this problem might be the examination of a cross-section with a focused ion beam (FIB). However it was not possible to include such an investigation here.

Topological Tunnelling Analysis

The irregularity of an asperity topology does not yield well to analysis especially if the surface is not well characterised. Roy *et al* [143] have approximated asperities by

[9] In this study a single trapping event was observed to make a significant impact on tunnelling current. This is not compatible with a current equally distributed along a large area such as an edge or planar surface.

[10] Private Correspondence with Gillian Marshall, September 1995.

[11] The doubling of edge length between tunnelling capacitor designs A and B would not lead to a lower V_{fe}, since V_{fe} would be determined by the sharpest asperity.

[12] Rinaldi *et al* [141] observed a small number of pronounced 'cusps' of polysilicon at plate edges in a TPFG process which project outwards more than the larger number of 'legitimate' injecting tips. It is conceivable that something similar may be occurring here.

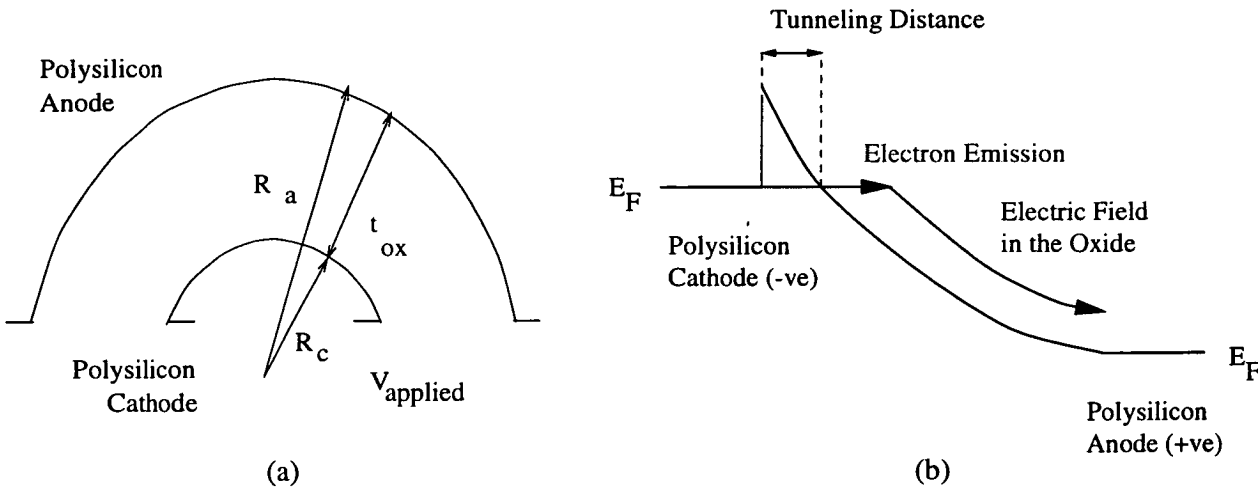


Figure 5.14: (a) Concentric polysilicon emission plates (b) Profile of resultant potential barrier (excluding image force lowering).

a curving emission plate (shown in figure 5.14). In this analysis the geometric field enhancement becomes

$$F_{enhanced} = \left(1 + \frac{t_{ox}}{R_c}\right) \frac{V_{applied}}{t_{ox}} \tag{5.13}$$

where R_c is the radius of curvature of the plate, the authors derived an expression for the electron tunnelling probability, equivalent to $\exp(T)$ in the Fowler Nordheim equation. For simplicity, the form neglecting the image force effect is given here:

$$T_c = exp\left[\left(-\frac{2\sqrt{(2m_{ox}F_{enhanced}R_c^3)}}{\hbar}\right)\left(\frac{\sin^{-1}\sqrt{\eta}}{\sqrt{(1-\eta)}} - \sqrt{\eta}\right)\right] \tag{5.14}$$

where $\eta = \frac{\phi}{F_{enhanced}R_c}$.

For t_{ox} = 60nm, the tunnelling probability is plotted against applied electric field (ie. $V_{applied}/t_{ox}$) in figure 5.15. Also plotted is the Fowler-Nordheim tunnelling probability; it can be seen how the probability defined by equation 5.14 decays towards the Fowler-Nordheim expression as the radius of curvature increases (At the limit $R_c \rightarrow \infty$, the plate becomes planar, and the potential barrier becomes triangular as in the Fowler-Nordheim analysis). As can be seen, a sharply curving surface allows much higher probability of electrons tunnelling for the same applied voltage across the oxide.

In practice it is not the plates of the polysilicon capacitors which would curve but that regions of the surface would be non-planar. Clearly the radius of curvature of

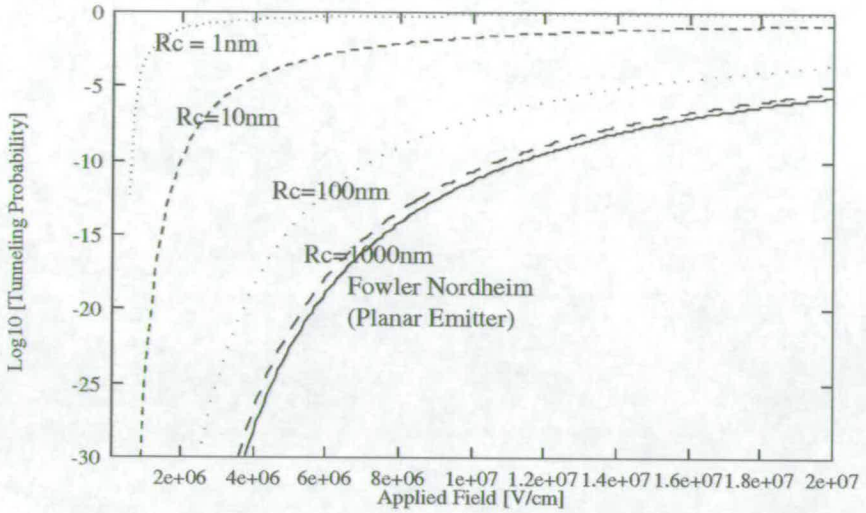
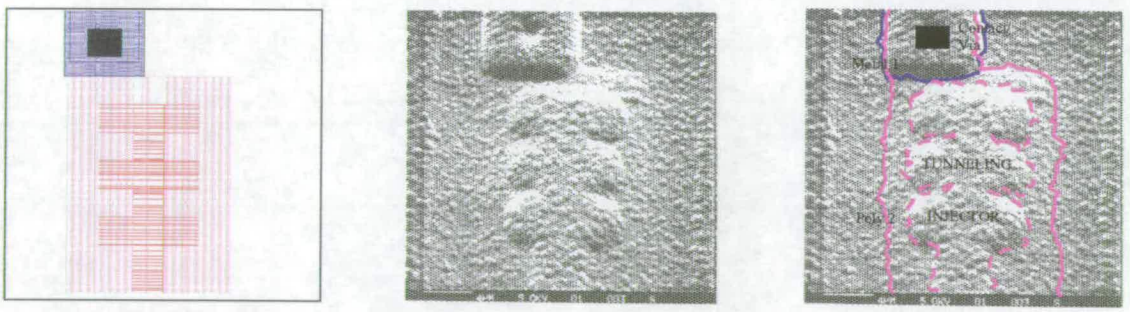


Figure 5.15: Electron emission probability from a polysilicon surface of radius of curvature R_c

such asperities must be much less than that of oxide thickness. At such levels, the requirement of concentricity may be relaxed, since t_{ox} is comparatively large.

Asperities with very small radii of curvature obviously cover very small areas. Therefore, if tunnelling *is* asperity-dominated as believed, then current will tunnel more easily through very small regions of the plate, rather than equally across the plate or along edges as initially believed.



(a) (b) (c)

Figure 5.16: (a) Layout of type E tunnelling capacitor, (b) SEM surface photomicrograph, (c) Annotated SEM surface photomicrograph.

Some *TARDIS* devices were examined under a scanning electron microscope (SEM). A typical observation is shown in figure 5.16. Since this is the view from the chip surface and the passivation layer had to be gold-coated to allow SEM observation,

it would be unreliable to draw any conclusions on the surface topology. However it is indicative that significant rounding has occurred at the corners which would eliminate any significant localised electric field enhancement (and so is in agreement with the V_{SO} experiments).

It is important to note at this point that these conclusions *only apply to the experiments with the available Alcatel-Mietec 2.4μm CMOS process.* There is no real justification here to disagree with the findings of others for other fabrication processes on the effect of edges and corners (especially when tunnelling is through the gate oxide, not extensively examined here, since the thinner, better-quality oxide may have different characteristics) however corner-enhancement does seem *unlikely* since a general consensus appears to exist that corners are inevitably rounded in all processes. This is due to the finite wavelength of light which precludes the ability to generate a singularity. It would appear most likely that tunnelling characteristic are strongly process specific - and possibly run specific and depend on such criteria as oxide quality, surface smoothness and layer 3D geometries.

Gate Oxide Injection Curve Irregularities

The gate oxide electron injection curves for tunnelling in the n-channel sense transistor are shown in figure 5.13 show an extreme irregularity in the processed data compared with the other curves that cannot simply be attributed to aging during the experiment.

A closer examination revealed an initial-conditions dependency not found in the interpoly tunnelling or in electron removal through the gate oxide (shown for one device in figure 5.17). The reason for this is unknown. One possible hypothesis is that while there is a plentiful supply of electrons for tunnelling *from* the polysilicon since it is degenerate. However if it is supposed that injection onto the floating gate through the gate oxide that the main source of electrons is the channel inversion region, then if there are insufficient electrons in this region then tunnelling is reduced; the electrons can only be replenished under the constraint of finite carrier mobility. So, for a low V_{SO} (high ϕ_{fg}) there is already an inversion layer in existence before programming starts, compared to a high V_{SO} (low ϕ_{fg}) for which the inversion layer only comes into existence when V_{PP} is coupled onto the floating gate. The more inverted the channel is prior to programming, then the more tunnelling occurs. This might also contribute to the asymmetry of $V_{fe}(gateox)$[13].

It might be possible to explore this hypothesis further by altering t_{PP}. Unfortunately this anomaly was only encountered when processing data after the test board had

[13]Note too that during *injection* the floating gate is biased such that it may stimulate SHE injection in addition to FNT. It is not straightforward to discriminate between the two injection mechanisms.

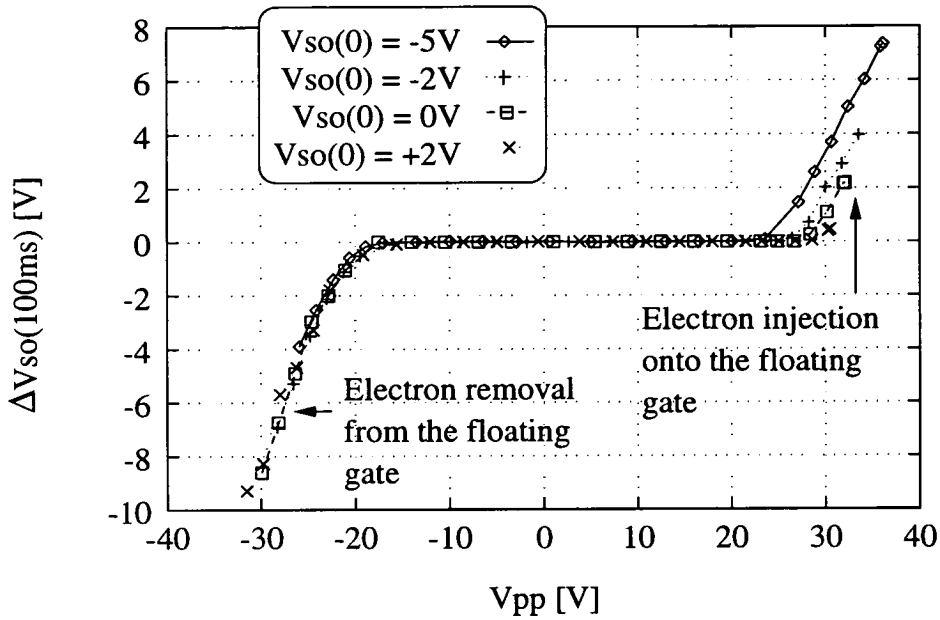


Figure 5.17: Typical ΔV_{SO} versus V_{tunnel} graphs using the capacitor network model to transform experimental data, for injection/removal through the gate oxide of the sense transistor.

been disassembled for parts[14] and it was not possible to do such an experiment. Notice that if this hypothesis is true then it would require that tunnelling electrons come from the *channel*, not the underdiffusion regions suggested by Carley. This would mean that, in this case, injection in the gate oxide is *not* along the edges.

Yield

Not all test structures were included in table 5.2. One entire chip and several type H devices were used for the previously described experiments and were thus intrusively aged and so could not be included in the table. In fact, of the 80 floating gate devices (10 chips of 8 devices) fabricated 3 were found to be non-operational at first examination [15]. However, subsequently, during the course of this work several more of the devices failed. Most of these failures occurred during experiments involving the application of high voltages or currents and are attributable to normal breakdown

[14]Initially only interpoly data had been processed.

[15]The faulty devices were device A on chip 4, device E on chip 5 and device H on chip 9. These devices passed zero current and could not be programmed to do otherwise. There appeared to be a short circuit from the floating gate to the bulk.

behaviours[16]. The others failed more mysteriously between experiments whilst in storage. It is now believed that most of these failures are due to electrostatic zapping due to poor handling and storage conditions. Since the programming characteristics were considered completely unknown at onset, all ESD (see section D.9) protection circuits had been removed. ESD damage may also account for premature wear-out observed in some devices.

5.3 Analogue Memory Cell Applications

Having demonstrated the analogue programmability of the *TARDIS* floating gate devices, two simple applications were investigated: (*i*) programmable current reference, and (*ii*) programmable Euclidean distance calculator.

DC Current Reference

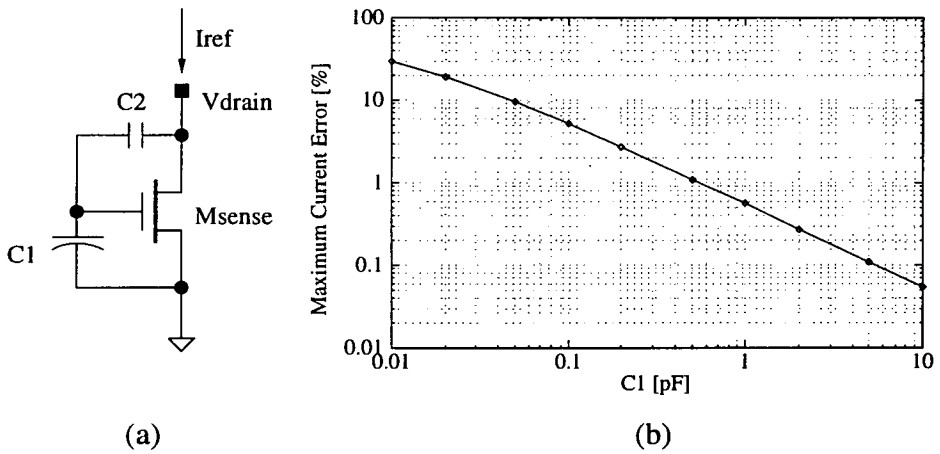


Figure 5.18: (a) Schematic of *TARDIS* floating gate acting as a current reference. $C_{control}$ and C_{tunnel} are grounded to form C_1. (b) Maximum absolute error in replicating behaviour of a driven current source in the range $V_{drain}, V_{fg} \leq 5$V (Hspice simulation results).

Floating-gate devices are considered to make poor current references because of the drain-coupling effect. Consider the situation shown in figure 5.18(a) where a floating gate sense transistor is acting as a constant current sink. $C1$ is the floating gate

[16]Several wear-out behaviours were also noted in very aged devices which, whilst they could be programmed would leak their charge over the course of a few seconds or minutes

capacitance and $C2$ is the parasitic capacitance due to lateral underdiffusion of the drain. If we assume the M_{sense} is operating in saturation, then:

$$I_{ds} = \frac{\beta}{2}(V_{gs} - V_{T_n})^2(1 + \lambda V_{ds}) \tag{5.15}$$

$$= \frac{\beta}{2}\left(\frac{Q_{fg}}{C_1 + C_2} + \frac{C_2 V_{ds}}{C_1 + C_2} - V_{T_n}\right)^2 (1 + \lambda V_{ds}) \tag{5.16}$$

where λ is the modelling term for channel length modulation. Calculating the small-signal drain-source transconductance, g_{ds} (which as a metric of current reference behaviour should approach zero):

$$g_{ds} = \left.\frac{\partial I_{ds}}{\partial V_{ds}}\right|_{q-point} \tag{5.17}$$

$$= \beta\frac{C_2}{C_1 + C_2}\left(\frac{Q_{fg}}{C_1 + C_2} + \frac{C_2 V_{ds}}{C_1 + C_2} - V_{T_n}\right)(1 + \lambda V_{ds}) \tag{5.18}$$

$$+\frac{\beta}{2}\left(\frac{Q_{fg}}{C_1 + C_2} + \frac{C_2 V_{ds}}{C_1 + C_2} - V_{T_n}\right)^2 \lambda \tag{5.19}$$

The first term on the right-hand side of equation 5.17 is the transconductance due to parasitic coupling, whereas the second term is simply due to channel length modulation and would be present in any transistor. Of the two terms, the parasitic coupling dominates - for typical values - for C_1 less than about two orders of magnitude larger than C_2. Figure 5.18(b) plots the absolute maximum current error (as a percentage of the expected current) of a floating gate current reference compared to a driven transistor in the range $V_{drain}, V_{fg} \leq 5\text{V}$.

Thus in practical terms *TARDIS* floating gates form perfectly adequate dc current sinks since when $C_{control}$ and C_{tunnel} are grounded, they form a C_1 almost three orders of magnitude larger than the drain capacitance. This is due to the large geometries of the process leading to naturally large capacitors, as described earlier, and also because tunnelling occurs in a specifically separate tunnelling capacitor. This is in contrast to special process EEPROMs, where the floating gate capacitance may simply be the nodal gate capacitance and where tunnelling is through a thin drain oxide, resulting in a much larger parasitic drain capacitor. Here it would be preferable to implement, say, a simple cascode arrangement, such that the drain of the sense transistor is held relatively constant.

However, since the sense transistors are minimum length to facilitate CHE experiments, traditional problems with channel length modulation g_{ds} *is* significant. For this reason, for experimental purposes only, V_{drain} was fixed at 2V for current reference experiments. This constraint can be eliminated simply by elongating the sense

transistor.

Programming Algorithm

Kolodny *et al* [92] demonstrated a coarse linearity between the change in an EEPROM cell threshold and the applied V_{PP}. Ong *et al* [130] used a standard EEPROM package to store analogue sound-waves in real time using a linear mapping between V_{PP} and cell current. However, as figure 5.19 shows, the V_{SO} response to the application of a 100ms programming pulses of magnitude V_{PP} demonstrates poor linearity in the *TARDIS* cells. More significantly the on-set of this poor linearity, and its gradient and offset varies widely from cell to cell (all cells tested had been equally aged; differential aging would worsen this condition). In addition, programming transitions of less than $\sim$ 2V (which would be typical) are extremely non-linear.

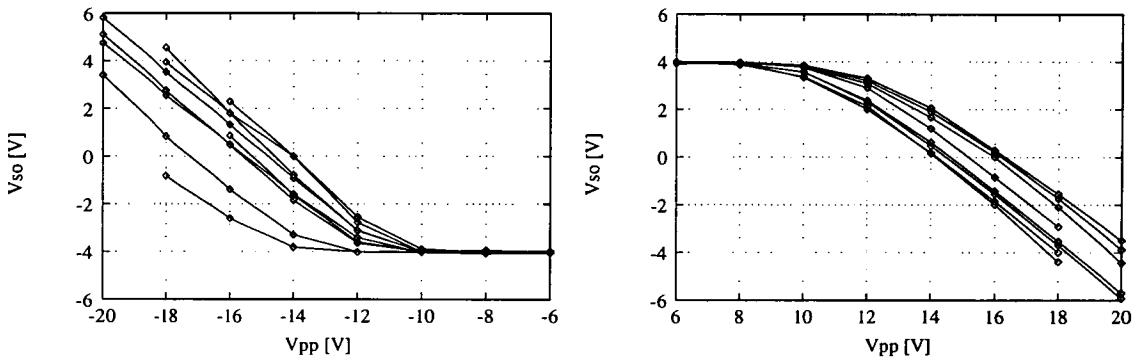


Figure 5.19: V_{SO} after application of 100ms programming pulses (for several different overlapped test cells). $V_{SO}(0) = -4$V for negative V_{PP} and $V_{SO}(0) = 4$V for positive V_{PP}. Similar results were obtained for gate oxide experiments.

Therefore, instead of trying to compensate for the poor linearity and matching characteristics which makes a direct mapping infeasible, a heuristic iterative programming algorithm was developed. Comparison of figures 5.8(c) and (d) with 5.19 demonstrates a much stronger voltage than time dependency (as would be expected from the FNT equation), so it was decided to base programming on a variable V_{PP} (as was done in the ETANN iterative programming algorithm). The aim here was not to provide an optimal algorithm in any respect but rather to show that a reasonably robust algorithm existed which would allow programming to take place over all the mismatched and aging device characteristics to a fair degree of precision. Improving on this algorithm (for example to minimise its time requirement) could be considered a potential future project (and its on-chip implementation; here the algorithm exists as C code on the controller PC).

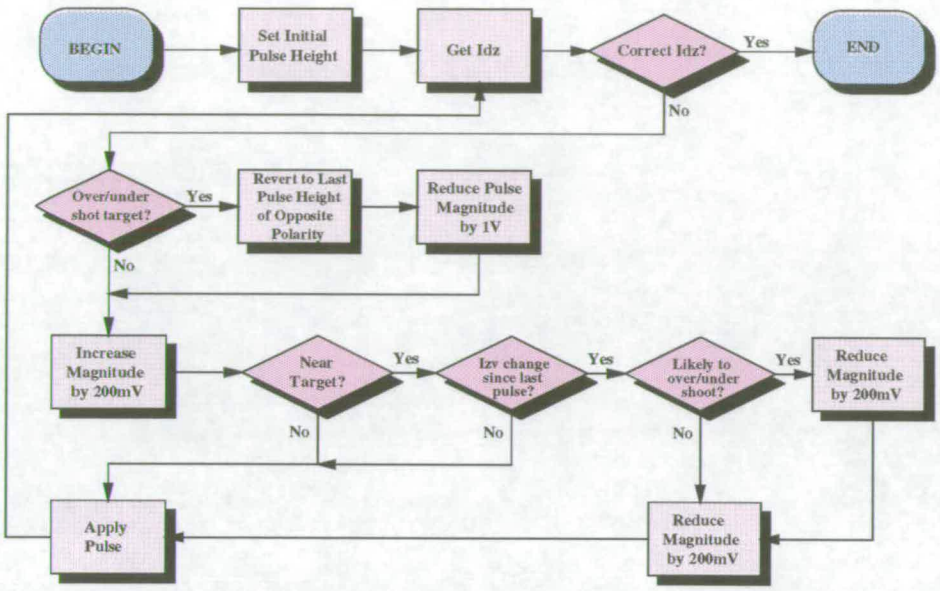


Figure 5.20: Flowchart of programming algorithm for *TARDIS* floating gate current sinks.

A simplified flowchart of the programming algorithm used for *TARDIS* floating gate current sinks is shown in figure 5.20[17]. (This is very similar to the algorithm was used for initialising V_{SO} in earlier experiments (I_{dz}[18] was effectively replaced by V_{SO})).

Programming progresses simply by incrementing the magnitude of the programming pulse in 200mV steps until the target, I_{dz}^{T}, is reached; if programming overshoots or undershoots the target, the pulse polarity is reversed, switching the direction of electron transport during programming pulses (The 1V magnitude reduction introduces hysteresis which prevents I_{dz} oscillating about the target value). The magnitude increment is annulled if the programming is close to the target to allow homing-in on I_{dz}^{T}, and the magnitude is actually decremented if there is likelihood of overshoot. Vague issues such as 'nearness' to target and 'likelihood' of overshoot can be quantified in terms of $|I_{dz}^{T} - I_{dz}|$ and δI_{dz} for the preceding pulse.

Included, but not shown in the flowcharts, are limits on the magnitude of V_{PP} such to provide a margin against likely oxide-breakdown potentials. Pulses above these magnitudes are clipped to the limits.

[17]This iterative algorithm is very much simpler than that used for ETANN [42] since it does not attempt to encapsulate the behaviour of tunnelling characteristics explicitly. Here simplicity is favoured over efficiency (although a direct comparison with the ETANN scheme was not possible) and provides the potential for future on-chip integration of programming.

[18]Drain current with zero voltage applied to TUNNEL and COUPLE terminals.

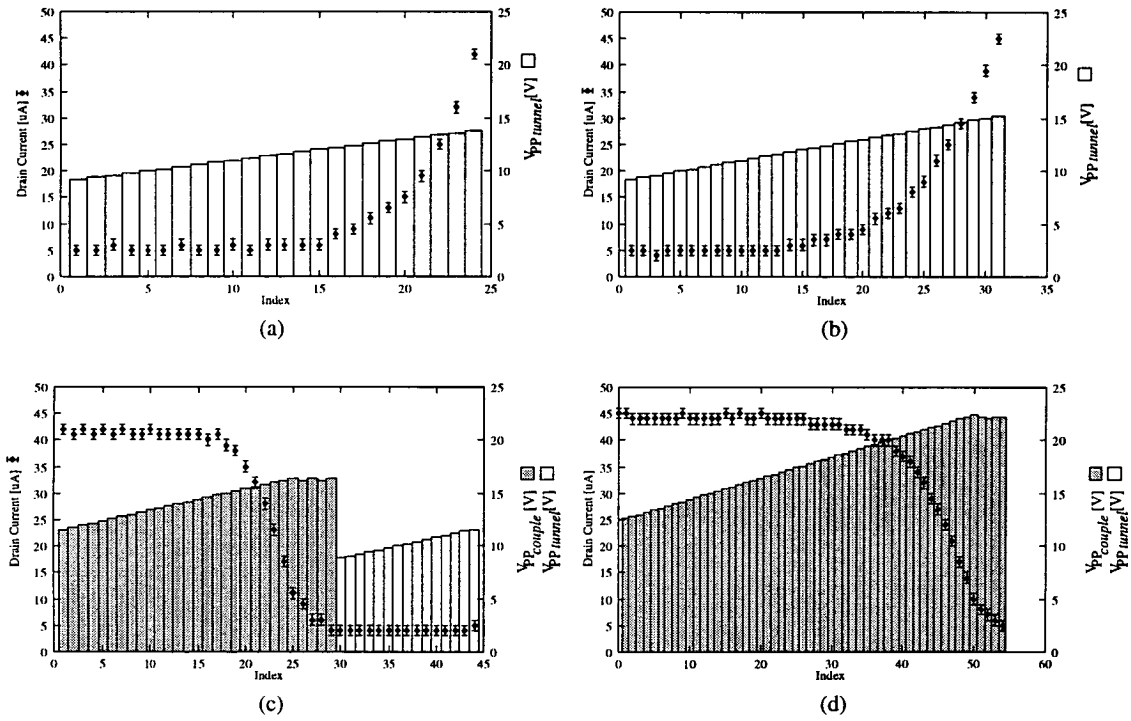
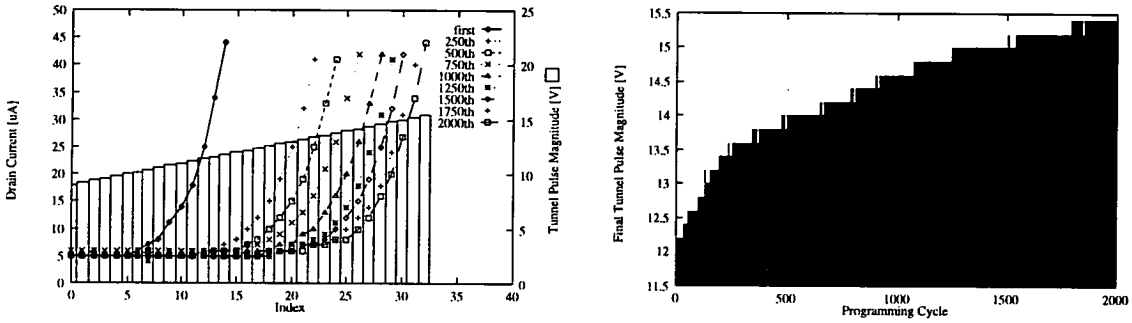


Figure 5.21: I_{dz} programming characteristics of (a) type B device - incrementing current (b) type G device - incrementing current (c) type B device - reset to 5μA (d) type G device - reset to 5μA

Figure 5.21 shows the I_{dz} programming characteristics for a couple of test devices, again illustrating the difference between overlapping and underlapping tunnelling capacitors. The configuration shown in figure 5.7(f) has been used for electron injection.



(a) (b)

Figure 5.22: (a) I_{dz} response to incremental $V_{PP_{tunnel}}$ pulse stream for a selection of programming cycles (b) Magnitude of final $V_{PP_{tunnel}}$ pulse in incremental pulse stream over 2000 programming cycles (programming terminates when $I_{dz} \geq 40\mu A$)

Aging was also apparent in the current references, as illustrated in figure 5.22. Note the logarithmic aging effect in figure 5.22(b). This can be explained by asperity-controlled tunnelling. Localised electric field enhancement at the sharper asperities dominate early cycles but become blocked by traps first, whereupon the greater number of blunter asperities take over.

Euclidean Distance Calculator

Collins, Marshall and Brown [39] proposed a two floating gate 1-dimensional Euclidean distance circuit for use in an analogue VLSI RBF [107]. The circuit is shown in figure 5.23(a). In their implementation the floating gates were programmed through a pass transistor activated by a `refresh` signal although it was their intention to migrate the design to non-volatile floating gates by removing the pass transistors and effecting programming by tunnelling or hot electron injection. An implementation similar to this can be formed by parallelising two *TARDIS* devices as shown in figure 5.23(b)[19].

In this design, both transistors are biased to operate in saturation and it is desired that I_{eucl} is minimised when $V_{in} = V_{centre}$. Thus, considering the left-hand transistor first, it is programmed such that $V_{fg} = V_{match}$ ($V_{match} \sim V_{T_n}$) when $V_{in} = V_{centre}$[20].

[19]For simplicity all parasitic floating gate shunt capacitance has been lumped with C_{tunnel}.

[20]In the original volatile version, this can be achieved simply by applying V_{centre} to the *in* terminal and pulsing the refresh control.

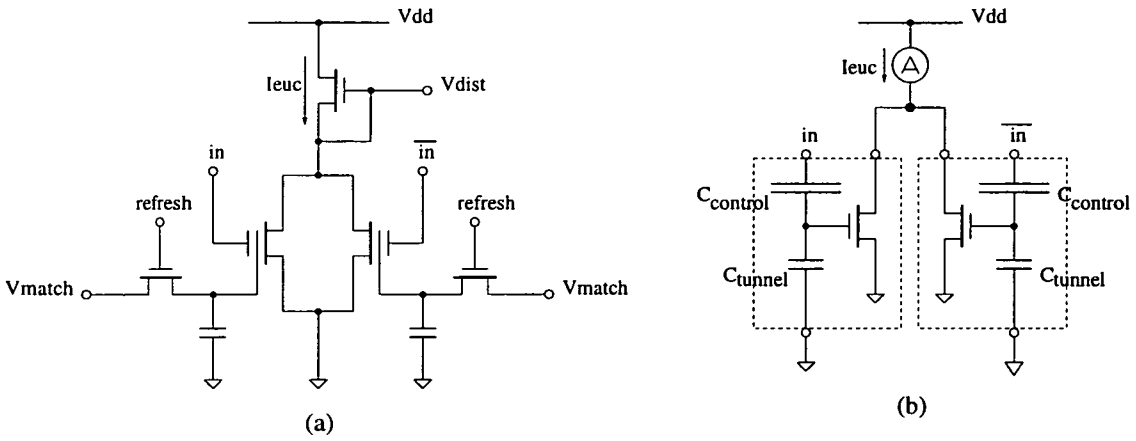


Figure 5.23: (a) Circuit diagram of one dimensional Euclidean distance cell with refresh, (b) Non-volatile floating gate implementation based on two parallelised *TARDIS* devices.

Hence

$$V_{match} = \frac{C_{control}}{(C_{control} + C_{tunnel})} V_{centre} + \phi_{fg} \tag{5.20}$$

Which can be re-arranged to find the necessary stored charge

$$\phi_{fg} = -\frac{C_{control}}{(C_{control} + C_{tunnel})} V_{centre} + V_{match} \tag{5.21}$$

Thus when an arbitrary input V_{in} is applied, the floating gate voltage can be calculated as

$$\begin{aligned} V_{fg} &= \frac{C_{control}}{C_{control} + C_{tunnel}} V_{in} + \phi_{fg} \\ &= \frac{C_{control}}{C_{control} + C_{tunnel}} (V_{in} - V_{centre}) + V_{match} \end{aligned} \tag{5.22}$$

Thus the current in the sense transistor can be calculated to a first order by

$$\begin{aligned} I_{ds} &= \frac{\beta_{left}}{2} (V_{fg} - V_{T_n})^2 \\ &= \frac{\beta_{left}}{2} \left(\frac{C_{control}}{C_{control} + C_{tunnel}} \right)^2 (V_{in} - V_{centre})^2 \end{aligned} \tag{5.23}$$

since $V_{match} \sim V_{T_n}$. Thus this transistor provides an approximation to one half of the

Euclidean calculation, ie. where $V_{in} > V_{centre}$. To include the cases where $V_{in} < V_{centre}$, the symmetrical right-hand transistor is required. It is programmed to V_{match} with $V_{\overline{centre}}$ on the control terminal where $V_{\overline{centre}} = V_{dd} - V_{centre}$. During normal operation $V_{\overline{in}} = V_{dd} - V_{in}$ is applied to the control terminal. This can be shown to generate a complementary current of the form

$$I_{ds} = \frac{\beta_{right}}{2} (V_{centre} - V_{in})^2 \tag{5.24}$$

Thus with the two drains connected together a complete Euclidean approximation across the V_{ss} to V_{dd} range is achieved.

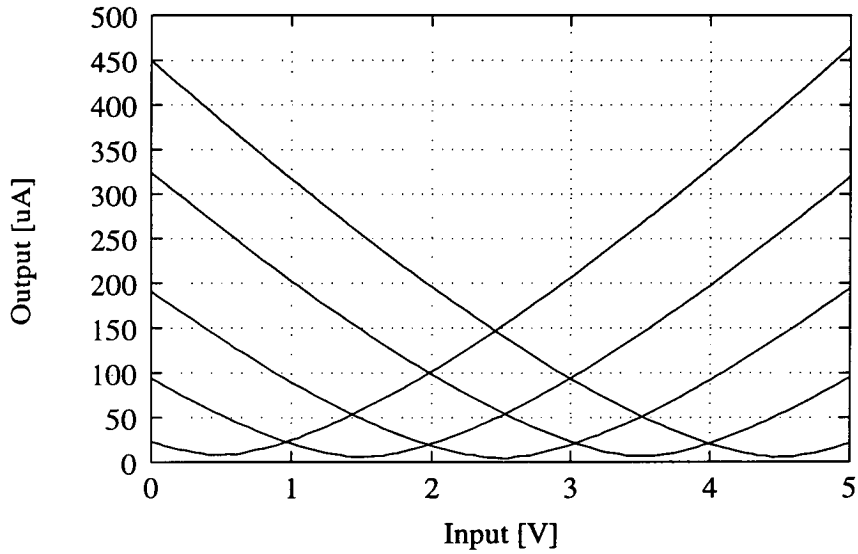


Figure 5.24: Experimental measurement of the response of 1-dimensional Euclidean distance calculator for centres at 0.5V,1.5V,2.5V,3.5V and 4.5V.

An experimental trial of this circuit is shown in figure 5.24. The width of the 'centre' was not a sharply defined point, but would vary depending on accuracy to which the two floating gates were programmed. If the programming 'overlapped', both transistors would be on simultaneously over some part of the range leading to a non-zero centre. If programming did not overlap at all, both transistors would be off simultaneously over some part of the range leading to a flat-bottomed curve. The extent of this non-ideality depends on how accurately programming can be achieved. I_{eucl} is larger than would be desirable in practice due to the minimum channel length of the sense transistors.

5.3.1 Thermal Trap Annealling

As noted in section D.2, trapped charge may become thermally liberated from its trap site back into the conduction band. It was considered possible then that a high temperature treatment could be used to mitigate aging by adding thermal energy and thus promoting the detrapping process. This is referred to as *annealling* of trapped charge[21].

A practical difficulty arises in the case of TARDIS in the use of a standard process; that of thermal damage by heating. By experimentation, the IC packaging clearly began to melt at temperatures in excess of about 80°C. Therefore annealling attempts had to be kept below this temperature, and a temperature of around 70°C was used.

Operation at elevated temperatures would change the characteristics of the memory as electrons would tend to be more energetic. Hence following annealling, chips were left to cool for 8 hours before being re-tested allowing time for them to return to room temperature.

Devices were operated as current references before and after the annealling experiment. As can be seen from figure 5.25(a), simply leaving the memory device at room temperature for 8 hours has no annealling influence[22]. However, as shown in figure 5.25(b-d) there is a definite lowering of programming voltage between the pre-annealling (trace 1) and post-annealing (trace 2) experiments. However, the annealing effect falls far short of returning the device to its virgin state (trace 0). Although various annealing times were tried, there seemed to be no strong time dependency at this temperature.

Perhaps the most significant feature of figure 5.25(b-d) is the rapid re-aging during the ten programming cycles between trace 2 and trace 3. In fact, the memories very quickly resorted to programming voltages equal to or above that of the pre-annealed state within only a handful of programming cycles.

This evidence would suggest that this relatively low annealling temperature is sufficient to liberate electrons from 'shallow' trap sites, but that these are the very same trap sites which are easy to fill again during programming. Hence the extreme nature of heat treatment would seem to little justify its small and short-term success in reversing aging.

[21]This is similar, but not to be confused with the high temperature annealling often performed to improve the quality of VLSI grade oxides. Annealling can out-diffuse impurities and also slightly move or re-orient dangling bond atoms allowing pairing up of dangling bonds, thus in both cases reducing the number of trap sites in the oxide. Thermal annealling usually requires temperatures several hundred degrees higher than room temperature.

[22]Strictly, a very slight annealling influence was observed in a couple of devices tested but this is to be expected from random detrapping events.

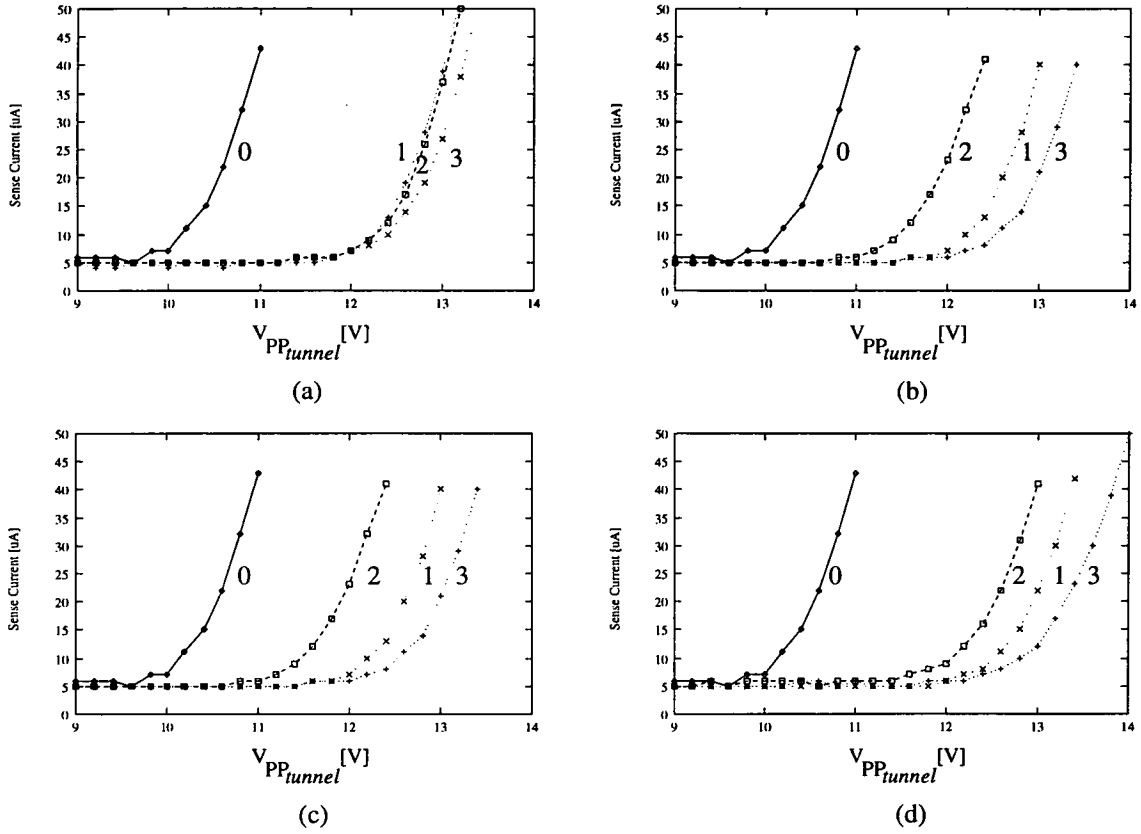


Figure 5.25: Heat treatment annealling for a highly aged memory: (a) Room temperature overnight, (b) 70°C for 4 minutes, (c) 70°C for 20 minutes, (c) 70°C for 30 minutes. In all cases, trace 0 shows the initial I_{zv} response to $V_{PP_{tunnel}}$ pulses immediately after TARDIS was delivered from fabrication, trace 1 shows the I_{zv} response to $V_{PP_{tunnel}}$ pulses immediately prior to annealing, trace 2 shows the I_{zv} response to $V_{PP_{tunnel}}$ pulses following annealing and following a cooling period of 8 hours, and trace 3 shows the I_{zv} response to $V_{PP_{tunnel}}$ pulses after 10 repetitions of the experiment subsequent to trace 2.

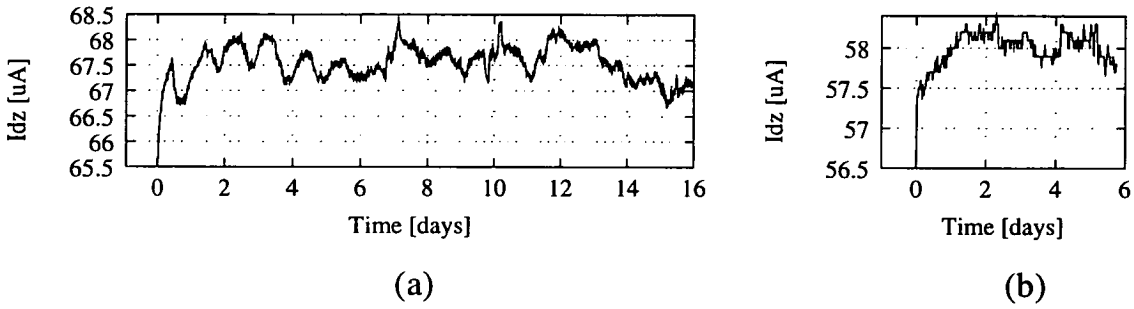


Figure 5.26: Continuous I_{dz} READ measurements commencing immediately after programmed floating gates and continued over a number of days.

5.4 Data Retention and Accuracy

In previous work with amorphous silicon resistors [80], non-volatility was tested by programming resistances and then powering these up for 30 minutes each day for a week. A more stringent test, it would seem, would be to lock the floating gates into a continuous READ cycle and then continuously monitor the behaviour; this would expose the floating gate to continuous operating stresses and so be closer to a practical situation. Since these tests were time consuming and tied up equipment in demand only a few runs were attempted: a couple of typical traces are shown in figure 5.26. In trace 5.26(a), the drift in I_{dz} represents a ϕ_{fg} drift of $\sim$ 5mV over one hour subsequent to programming, and $\sim$ 18mV over one day. Thereafter there is an oscillation of ϕ_{fg} within bounds of $\sim$ 18mV[23] but the trend of this trace is unclear. Trace 5.26(b) shows similar behaviour, with a ϕ_{fg} drift of $\sim$ 10mV over one hour and $\sim$ 17mV over one day followed again by an oscillation, here within bounds of $\sim$ 9mV.

This marked *initial drift* (or *relaxation*) has previously been observed by Carley [21] in gate-oxide structures, Säckinger & Guggenbühl [145], Sin *et al* [152] in special process floating gates, and Marshall & Collins[24] in interpoly structures, although it has not been widely reported in other SLV-CMOS literature. Drifts of several hundred microvolts to a few millivolts have been reported, so the observations here were consistent. Several explanations for this behaviour are possible: redistribution of trapped charge within the oxide and at the interface (both within the tunnelling capacitor and the sense transistor), slow detrapping, redistribution of ionic oxide contaminants, short-lived traps, dielectric depolarisation [159] (all also affected by noise in read-out). If this relaxation is predictable then it can be pre-compensated for dur-

[23]There appears to be a slight daily periodicity to the signal, possibly related to external influences within the laboratory.

[24]Gillian Marshall. Private Correspondence, September 1995.

ing programming. Säckinger & Guggenbühl believe that prediction to an accuracy of 15% is possible for their special process floating gates [145]. However the accuracy of prediction found with *TARDIS* seems very much poorer (although there is insufficient data to estimate it well). It is probable that SLV-CMOS floating gates relax less consistently than special process devices and this puts a bound on long-term programming precision. Claims of > 14 bit resolution [96, 54] must therefore be treated with a degree of caution particularly if it is not made clear if the readout is repeatable after some time delay[25].

A further issue related to high accuracy programming is temperature stability; a threshold shift of a few millivolts may be caused by a 1K temperature change. Whilst it was always intended to operate the *TARDIS* test devices under the 'normal operating conditions' afforded by the bench top and so naturally confront such problems, if a more detailed study of relaxation were to be pursued it must consider using an environmental chamber to eliminate this external influence.

5.4.1 Long Term Data Retention

Of a sample of fourteen specifically SLV-CMOS floating gate papers, [21, 161, 160, 114, 96, 15, 26, 27, 67, 58, 57, 106, 54, 72], only four actually report any long term data retention experiments, the rest either assuming retention or citing one of the other studies.

At the low fields self-induced by ϕ_{fg}, FNT is negligible compared to charge loss by thermionic emission *over* the Si-SiO_2 boundary [74]. This charge loss can be accelerated by 'bake retention testing' (Appendix E) and using the calculated activation energy to predict low temperature charge loss

$$\frac{Q(t)}{Q(0)} = \exp\left(-t\nu \exp\left(-\frac{\phi_B}{kT}\right)\right) \tag{5.25}$$

where $Q(0)$ is the initial charge on the floating gate and $Q(t)$ is the charge at time t. ν is the electron-lattice collision frequency, k is Boltzmann's constant, T is the temperature and ϕ_B is the apparent Si-SiO_2 potential barrier height (the activation energy for charge loss). Activation energies of 1.1eV [161, 26](interpoly floating gate) and 1.6eV [21](gate oxide floating gate) have been calculated which predict less than 0.1% charge loss over ten years.

These results must be treated with caution since it is not necessarily straightforward to migrate this digital approach into precision analogue applications. In particular the

[25]Systems which implement continuous-time adaption rather than discrete program/read phases may, however, achieve higher resolution since any drift would be trimmed out.

charge loss rate is an almost exponential function of time; therefore initial degradation is relatively large. Other factors which must be considered are:

- Steady-state must be achieved to eliminate short-term leakage effects from long-term bake measurements; the causes, lifetimes and temperature dependencies of these mechanisms must be understood and accounted for.
- Thermal liberation of trapped charge during the bake must be accounted for: for example how does the quantity of charge liberated during 72 hours at high temperatures relate to the quantity liberated at room temperature over ten years?
- The influence and extent of non-catastrophic oxide defects must be known and how they will be accelerated by the bake.

These issues are seldom addressed in EEPROM testing as the programming margins are so large as to make shifts due to these effects negligible. The same cannot be said if high accuracy analogue operation is required. However the effects listed would probably tend to make retention appear *worse* rather than better than its true nature so the results published are very encouraging.

Some authors predict SLV-CMOS floating gates will have better retention properties than special process floating gates because of the thicker oxide but this claim would seem disputable: charge loss is dominated by thermionic emission which has no electric field (and hence no oxide thickness) dependency. However, special fabrication may optimise or screen oxidation processes to ensure high quality whereas in SLV-CMOS, the oxide is not optimised and so the defect density may be greater, promoting defect controlled charge loss. Therefore it is possible that statistical retention reliability of SLV-CMOS floating gates is poorer. Bake retention testing under the presence of defects may also be unreliable - the thermionic process has a clear physical temperature dependence but the temperature acceleration and cross-sectional influence of different types of defects may not be so straightforward (ie. one type of defect may dominate at low temperatures, another at high temperatures).

Analogue retention is clearly a complicated issue which perhaps partly explains why it has been little mentioned in the literature. A thorough examination of this topic is beyond the scope of this thesis although it was intended to do some basic bake retention tests to assure fundamental functionality. Unfortunately it was decided necessary to postpone intensive floating gate retention testing to the second chip rather than *TARDIS* for two principal reasons

1. ESD damage often gave rise to catastrophic damage would lead to an unprogrammable cell but less severe damage could simply impair the charge retention capabilities of the oxide. Thus with the high levels of ESD problems found with

Device	Storage Date	I_{dz}	Test Date	I_{dz}	Drift	
Chip3 (E)	28 Aug 1995	49.0μA	26 Sep 1996	50.6μA	+1.6μA	+19mV
Chip3 (G)	30 Aug 1995	48.0μA	01 Oct 1996	47.4μA	-0.6μA	-7mV
Chip4 (D)	30 Aug 1995	41.0μA	26 Sep 1996	40.8μA	-0.2μA	-2mV
Chip4 (F)	31 Aug 1995	44.0μA	17 Sep 1996	47.3μA	+3.3μA	+41mV

Table 5.3: I_{dz} measurements before and after a year in storage of four *TARDIS* devices. The estimated drift in ϕ_{fg} is derived from a typical `Hspice` model of the sense transistor.

TARDIS it seemed inappropriate to proceed with retention bake since leaky oxides due to ESD damage would be indistinguishable from an inherent retention problem.

2. All *TARDIS* samples came with standard packaging (40 pin DIL package) not capable of withstanding the high temperatures required for bake retention.

By deferring long term retention tests to the second chip these problems could be mitigated: anti-static precautions could be taken and unpackaged samples would be provided which could be bonded into specialised packaging by a third source [26].

However, the I_{dz} of four *TARDIS* devices was recorded immediately after programming and before they were stored in a sealed box for a year[27]. Re-examination values of I_{dz} are listed in table 5.3. Since these floating gates had not been allowed to reach steady state before being stored the short term and long term components of drift were unknown. Whilst these measurements are not particularly strong evidence (it is also unlikely that the measurement conditions were replicated exactly after a year and so the measured drift is probably worse than the actual drift) they do suggest that *TARDIS* floating gates do (at least in some cases) have the potential for reasonable long term analogue stability.

5.5 Hspice Modelling of the Floating Gate

Section 5.2.4 described the derivation of a capacitor network model with injected charge for Modelling of the floating gate. To complete the model, the tunnelling capa-

[26] *TARDIS* was fabricated by Eurochip and the subsequent two chips by Europractice. Europractice had a policy of providing a number of additional unpackaged samples, whereas this was not true of Eurochip.

[27] Obviously it would have been desirable to program and store a much larger sample and to target a wider range of stored charge (larger I_{dz} range). Unfortunately provision for this was not considered sufficiently early in the project.

citor current must be incorporated.

5.5.1 Fowler-Nordheim Coefficient Derivation by Capacitor Network Model

Fitting coefficients χ_1 and χ_2 to the Fowler-Nordheim expression (3.9) is usually done using a special test structure consisting of the tunnelling capacitor *without* a floating gate. Thus a voltage can be applied directly across the two terminals of the capacitor and the tunnelling current measured directly. Microscopic measurement of plate area and ellipsometric measurement of oxide thickness can be used to refine the fit. A least squares fit of the rearranged Fowler-Nordheim expression can be used to obtain the coefficients

$$\ln\left(\frac{I_{tunnel}}{V_{tunnel}^2}\right) = -\frac{\chi_2}{V_{tunnel}} - \ln\left(\frac{1}{\chi_1}\right) \tag{5.26}$$

or

$$\ln\left(\frac{I_{tunnel}}{V_{tunnel}^2}\right) = -\frac{A\chi_{2a}t_{ox}}{V_{tunnel}} - \ln\left(\frac{t_{ox}^2}{\chi_{1a}}\right) \tag{5.27}$$

Such a fit requires the validity of the Fowler-Nordheim expression over the region of operation. However, since it was impossible to attempt to measure directly the very small tunnelling currents with available equipment, it was necessary to reformulate the fit expression 5.26 in terms of voltages which could be more easily measured. The injected-charge capacitor network model was used for this purpose.

Reformulation of the Fowler-Nordheim Expression

To model the behaviour of the tunnelling capacitor it is necessary to derive a reformulation of the Fowler-Nordheim expression, equation 3.9, into an expression involving V_{tunnel} instead of I_{tunnel}.

$$V_{tunnel}(t) = V_{PP} - V_{fg}(t) \tag{5.28}$$

Differentiating both sides, and substituting in the Fowler-Nordheim expression, gives

$$\begin{aligned}\frac{dV_{tunnel}(t)}{dt} &= \frac{d}{dt}\left(V_{PP} - V_{fg}(t)\right) \\ &= -\frac{d}{dt}V_{fg}(t) \\ &= -\frac{I_{tunnel}(t)t}{C_{fg}} \\ &= -\frac{\chi_1 V_{tunnel}^2(t)\exp\left(-\chi_2/V_{tunnel}(t)\right)}{C_{fg}}\end{aligned} \tag{5.29}$$

Now, rearranging and integrating

$$\int \frac{dV_{tunnel}(t)}{V_{tunnel}^2(t)\exp\left(-\chi_2/V_{tunnel}(t)\right)} = -\int \frac{\chi_1 dt}{C_{fg}} \tag{5.30}$$

gives

$$\frac{-1}{\chi_2\exp\left(-\chi_2/V_{tunnel}(t)\right)} = -\frac{\chi_1 t}{C_{fg}} + const_1 \tag{5.31}$$

Which may be rearranged into the form

$$V_{tunnel}(t) = -\frac{\chi_2}{\ln\left[C_{fg}/\left(\chi_2\left(\chi_1 t + const_2\right)\right)\right]} \tag{5.32}$$

Solving for initial conditions, $t = 0$, gives

$$V_{tunnel}(0) = -\frac{\chi_2}{\ln\left[C_{fg}/\chi_2 const_2\right]} \tag{5.33}$$

Which may be arranged to give

$$const_2 = \frac{C_{fg}}{\chi_2\exp\left(-\chi_2/V_{tunnel}(0)\right)} \tag{5.34}$$

Equations 5.11, 5.32 and 5.34 can now be combined to give a model for $V_{SO}\left(t_{PP}\right)$:

$$V_{SO}\left(t_{PP}\right) = \frac{1}{m_c}\left(V_{tunnel}\left(t_{PP}\right) - V_{PP}\left(1 - m_t\right) + V_{fg}(ON)\right) \tag{5.35}$$

where $V_{tunnel}(0) = V_{PP}\left(1 - m_t\right) - V_{fg}(ON) + m_c V_{SO}(0)$, ie. the tunnelling potential at the instant the pulse is applied.

To test the validity of the model defined by equations 5.33, 5.34 and 5.35, its pre-

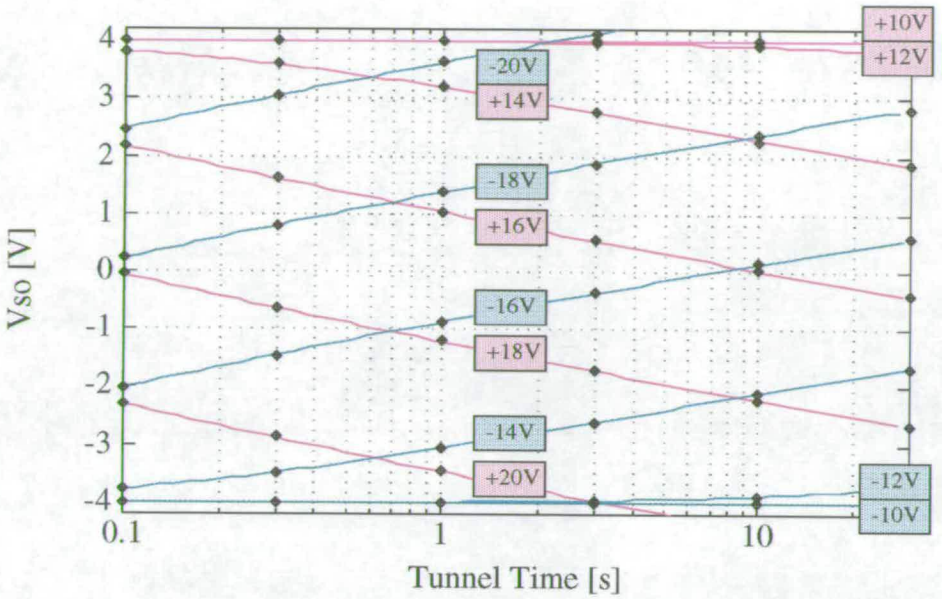


Figure 5.27: Model of the time dependency of V_{SO} in the V_{PP} and t_{PP} range of previous experiments. $V_{SO}(0) = +4\text{V}$ for each V_{PP}^{+} trace, and $V_{SO}(0) = -4\text{V}$ for each V_{PP}^{-} trace. $V_{SO}(0)$ is not shown on the graph because of the logarithmic x-axis. The points are the analytic estimations, the lines the numerical model.

dictions was compared to that of a numerical model based on

$$\delta\phi_{fg} = \frac{\chi_1 V_{tunnel}^2(t) \exp\left(-\chi_2/V_{tunnel}\right) \delta t}{C_{fg}} \tag{5.36}$$

with continual re-evaluation of V_{tunnel} between each discrete time step, δt.

This is illustrated in figure 5.27. Here χ_1 was chosen to be equal to $A_{plate}C_1\mu_1/t_{ox}$ and χ_2 equal to $\beta t_{ox}/\mu_2$ where A_{plate} is the layout area of plate overlap in the tunnelling capacitor ($92.2\mu\text{m}^2$), t_{ox} is the typical interpoly oxide thickness (60nm) and C_1 and β are the Fowler-Nordheim coefficients given in [184]. μ is included as an arbitrary field acceleration parameter such that $E_{ox} = \mu V_{tunnel}/t_{ox}$. μ_1 and μ_2 have been set to 4 to provide the realistic tunnelling voltages shown in figure 5.27. Simplistically, this represents a field enhancement of 4 times due to asperities and other effects averaged across the whole plate surface A_{plate}.

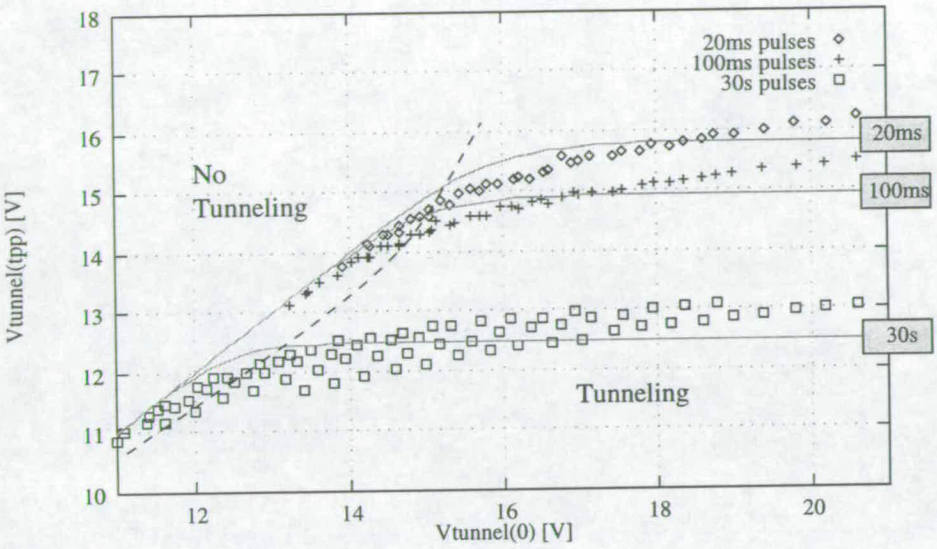


Figure 5.28: Best fit approximation of V_{tunnel} model to some experimental data for V_{PP}^{+}.

Experiment versus Model Comparison

Taking some experimental data in the form 5.12, an error function was created

$$error = \frac{1}{N_{data}} \sum^{N_{data}} \left(V_{tunnel}\left(t_{PP}\right)[model] - V_{tunnel}\left(_{PP}\right)[experiment] \right)^2 \quad (5.37)$$

An attempt was made to minimise this function in terms of χ_1 and χ_2 using the Nelder-Mead Simplex Method [138]. However, as figure 5.28 shows, a good fit was not obtained. In fact, it is not possible to fit χ_1 and χ_2 such that the slope to the right of the 'knee' in the curve is well represented. (Notice the wide spread in the 30s experiment compared to the 20ms and 100ms experiments - it is believed that this is because the random nature of trapping events becomes more diverse over a longer period). In conclusion, the Fowler-Nordheim fit proposed is not a good one for the devices fabricated on *TARDIS*. It is most likely that electric field variations due to surface asperities and possible edge effects and thinning mean that the behaviour during programming is somewhat more complicated than this simple model of universal field acceleration would predict.

It would seem plausible that a less constrained model, such as a neural network would provide an adequate fit to the experimental data. However, a model with a strong physical basis would seem impractical without further investigation of the sur-

```
13: ...
14: * Physical components
15: Cfg fg gnd! czeroed $ COUPLE=BULK=SOURCE=DRAIN=0V
16: Rfg fg gnd! 1e24 $ Dummy R - for convergence
17: Ctunnel TUNNEL fg ctunnel
18: GfnVppp TUNNEL fg cur='x1p*v(ptunn)*v(ptunn)*exp(-x2p/v(ptunn))'
19: GfnVppn fg TUNNEL cur='x1n*v(ntunn)*v(ntunn)*exp(-x2n/v(ntunn))'
20: Eptunn ptunn gnd! TUNNEL fg 1 MIN=0
21: Entunn ntunn gnd! fg TUNNEL 1 MIN=0
22: * Define intrinsic sense transistor switch on voltage
23: Vfgon fgon gnd! fgonvolts
24: * Calculation of Vso from Vfg
25: Etuncoup tunncoup gnd! TUNNEL gnd! mt
26: Esubs subs gnd! fg tunncoup 1
27: Evso vso gnd! fgon subs oneovermc
28: * Tunnel Programming Signal
29: VTUNNEL TUNNEL gnd! vpp
30: * Simplex Fitted Coefficients
31: .param x1p=2.7523e-4 x2p=334.307
32: .param x1n=31.7658 x2n=630.264
33: * Capacitor Network Model Parameters
34: .param ccontrol=464f cfb=60f cmosfet=10f ctunnel=46f
35: .param czeroed=par('ccontrol+cfb+cmosfet') ctotal=par('czeroed+ctunnel')
36: .param fgonvolts=0.97
37: .param mt=par('ctunnel/ctotal') mc=par('ccontrol/ctotal') oneovermc=('1/mc')
38: * Calculation of Initial Condition
39: .param vfg_0=par('fgonvolts-(mc*vso_0)+(mt*vpp)')
40: .ic v(fg)=vfg_0
41: .data experiment
42: vso_0 vpp
43: -4 -10
... ... ...
54: 4 20
55: .enddata
56: .print par('i(GfnVppp) - i(GfnVppn)')
57: ...
```

Table 5.4: Extract from an Hspice simulation file

Thus the calculation of V_{SO} can be implemented by lines 25-27 without deasserting V_{TUNNEL} as was done in the experiment. Notice that rearranging equation 5.39 again allows the setting of $V_{SO}(0)$ by initialising the voltage at node `fg`. This is implemented in lines 39-40 to match the initial conditions of the experiments. Line 56 calculates the tunnelling current as the difference between the two simulated sources in lines 18-19.

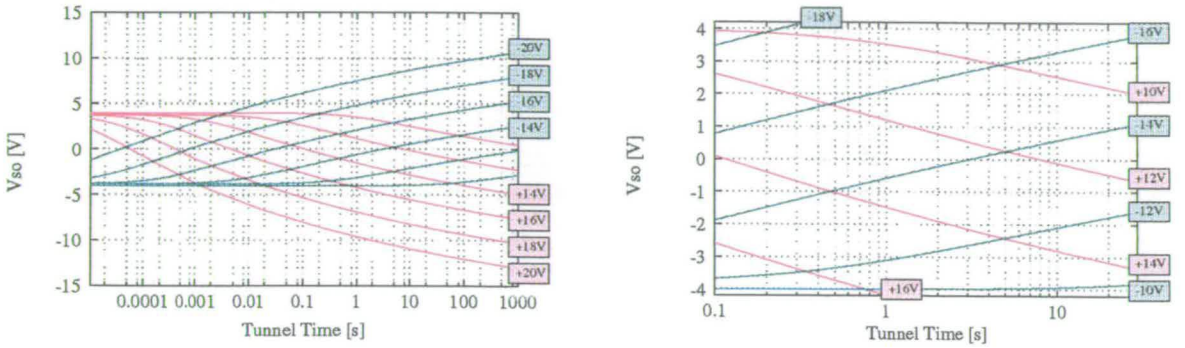


Figure 5.29: Hspice model of time-dependent programming experiment using Simplex Fitted Fowler-Nordheim coefficients (left) wide range (right) range matched to that of experiments. $V_{SO}(0) = +4\text{V}$ for V_{PP}^{+} traces and $V_{SO}(0) = -4\text{V}$ for V_{PP}^{-} traces. $V_{SO}(0)$ does not appear on the graph due to the logarithmic x-axis.

The output of this simulation is shown in figure 5.29. Notice that only four lines (18-21) define the tunnelling operation and so the model is easily transferable to new simulations.

5.6 Experiments with CHE Programming

Having investigated FNT in *TARDIS* floating gates, the second mechanism under investigation was channel hot electron (CHE) injection. Although electrons can be injected onto the floating gate the larger potential barrier means that holes generally cannot and so decrementing ϕ_{fg} still involves FNT of electrons off the floating gate, ie. CHE is a unipolar programming mechanism.

In a submicron process significant CHE currents can flow at comparatively low voltages but the large geometries of this process result in a much less extreme drain electric field. Thus there is no naturally large CHE current. It is important, then, when attempting to minimise both programming voltages and times, to maximise CHE by ensuring that $V_{gate} \sim V_{drain}$ can be achieved.

One approach to this is the system illustrated in figure 5.31 which uses a continuous time feedback loop involving an op-amp to control the voltage coupled onto the floating gate such that the difference between V_{fg} and V_{drain} is minimised. `CTRL` can

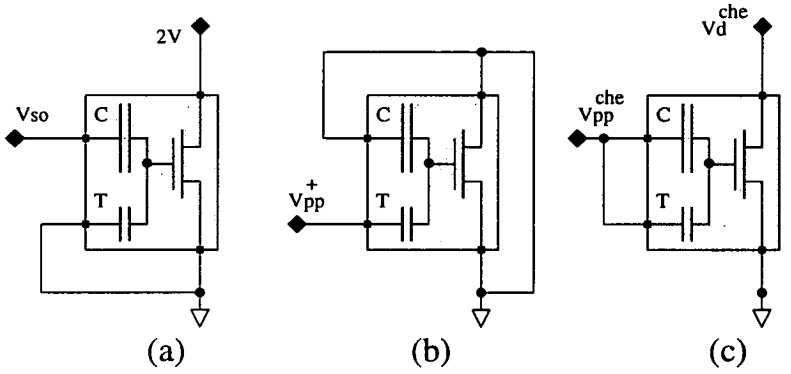


Figure 5.30: Hot Electron programming configurations for floating gate device: (a) READ mode - measurement of V_{SO}, (b) Electron REMOVAL mode through interpoly oxide by direct application of high positive V_{tunnel}, (c) Electron INJECTION mode of channel hot electrons through gate oxide by coupling high positive voltage onto floating gate and direct application of high drain voltage.

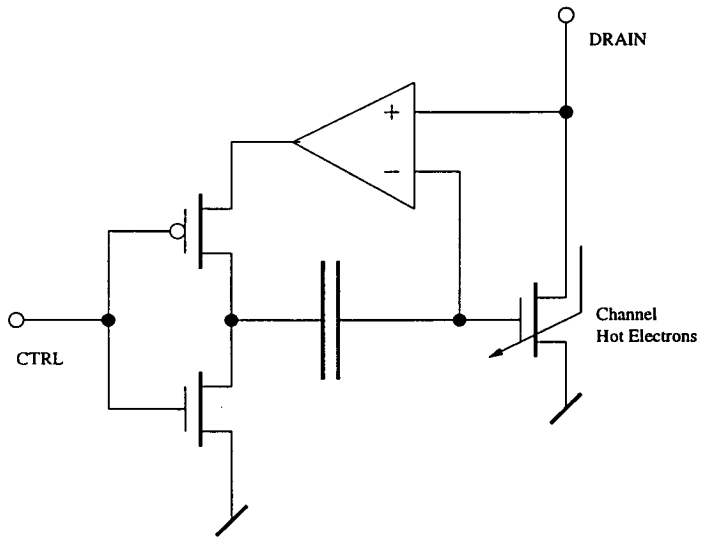


Figure 5.31: Continuous time $V_{fg} \sim V_{drain}$ system.

be used to pull V_{fg} sufficiently low to halt CHE. Although this system was not implemented on *TARDIS* it can be approximated by a discrete time (pulse/evaluate) version using the simple capacitor network model described in section 5.2.4. Using the CHE configuration illustrated in figure 5.30(c), a programming pulse, $\left(V_{PP}^{che}, t_{PP}^{che}\right)$, is applied to the (parallelised) COUPLE and TUNNEL terminals. Using equations 5.4 and 5.6, the voltage on the floating gate can be expressed as

$$V_{fg} = (m_t + m_c)\, V_{PP}^{che} + V_{fg}(ON) - m_c V_{SO} \tag{5.40}$$

Substituting $V_{fg} = V_{drain}$ into equation 5.40, and rearranging, provides a simple approximate model for determining the requisite value of V_{PP}^{che} for maximisation of I_{che}:

$$V_{PP}^{che} = \frac{1}{(m_t + m_c)} \left(V_{drain} - V_{fg}(ON) + m_c V_{SO}\right) \tag{5.41}$$

Experimental Results

A number of devices had their V_{SO} initialised to -5V by tunnelling. V_{SO} was then programmed upwards according to equation 5.41, with the drain V_{drain} ranged towards 8V, 1V above the process specified snap-back value [29]. Although programming *was* found to occur, it was about 1-2 orders of magnitude slower than for electron injection by tunnelling so t_{PP}^{che} was increased from 100ms to 3s. Notice that if V_{drain} is kept below 10V and only positive ϕ_{fg} values are valid, then V_{PP}^{che} can be constrained to be within the 12V supply voltage specification.

Some typical programming traces are shown in figure 5.32 which clearly illustrates a speed-up due to higher drain voltages due to the larger population of hot electrons (although it is still much slower than for tunnelling). If the model of equation 5.41 was correct approximately the same amount of charge should be dumped on the floating gate between each pulse leading to linear traces. This is obviously not the case and the model is suboptimal; however it is good enough to allow programming across then entire -5V $\rightarrow$ +5V range of interest.

Model Offset

In a simple attempt to measure the sensitivity of programming time to the programming model a simple offset constant was introduced. Equation 5.41 was reformulated to

[29] Some margin was expected above the process specified snapback voltage of 7V. In fact tested breakdown voltages ranged from 8.5V to 11.0V but operation above about 8V could not be assured. Breakdown was characterised by a sudden collapse of the drain current to zero and subsequent non-programmability. The drain terminal then appeared to have become disconnected possibly due to metal migration or destruction of the transistor due to thermal heating.

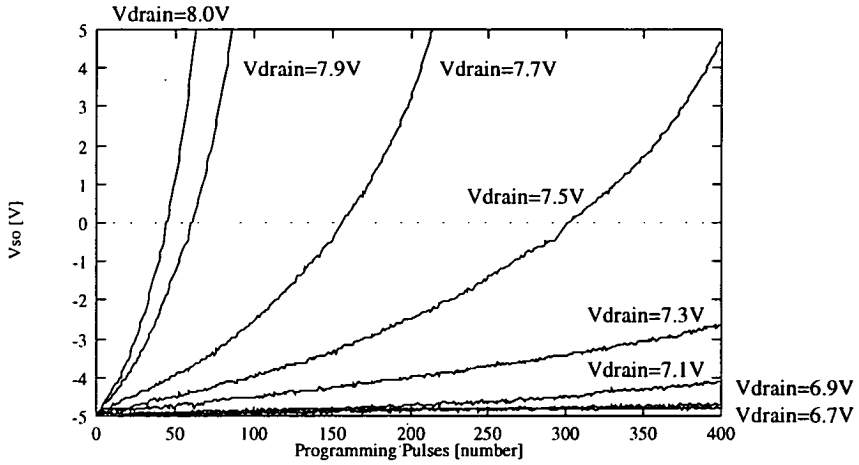


Figure 5.32: Number of model-based programming pulses required for drain voltages up to 8.0V. T_{PP} = 3 seconds.

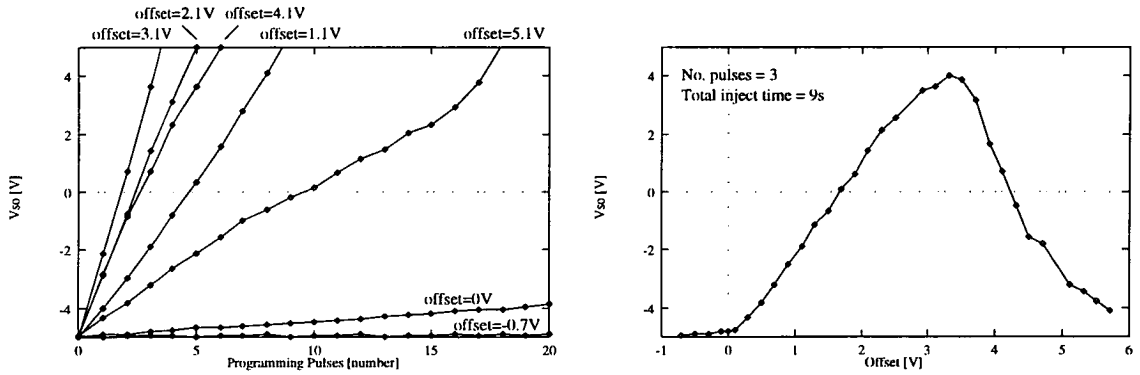


Figure 5.33: (a) Programming curves for various model offsets, (b) Final V_{SO} after a total of 9s inject time for a range of model offsets. Each V_{SO}(time = 0) = -5V.

include an offset term:

$$V_{PP}^{che} = \frac{1}{(m_t + m_c)}(V_{drain} - V_{fg}(on)) + \frac{m_c}{(m_t + m_c)}V_{SO} + offset \qquad (5.42)$$

Programming curves for various offsets are shown in figure 5.33(a), which also confirms that the model-derived V_{PP}^{che} is also not optimal. In fact, as shown in figure 5.33(b), (where V_{SO} after 3 pulses from an initial V_{SO} = -5V is plotted against offset), an offset of about 3.5V represents the peak I_{gate} for the adjusted model.

This behaviour may be explained by looking at the form of the I_{gate} against V_{gate} curve for a fixed V_{drain}, as shown in figure 5.34(a) (This curve is sketched from an experimental plot in [139]). I_{gate} falls off sharply below the V_{gate} peak but, less sharply

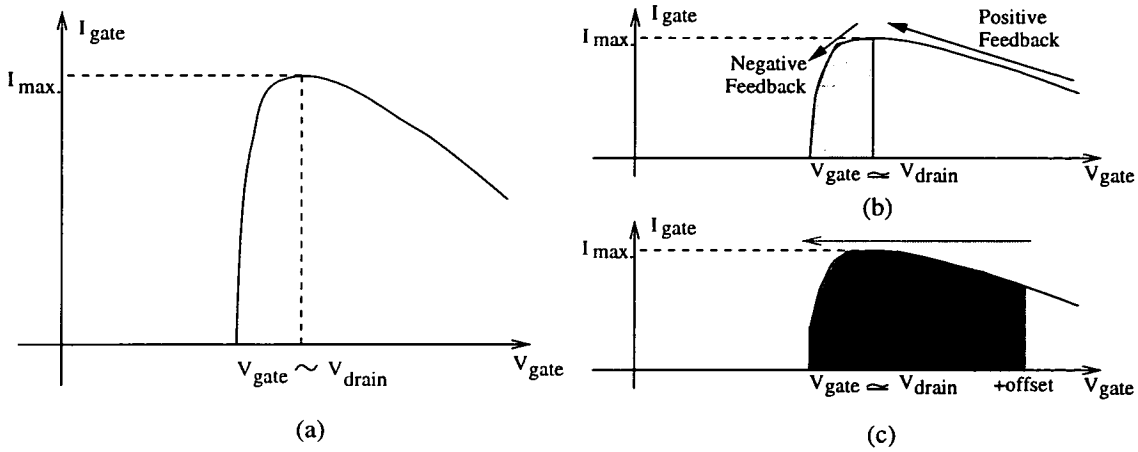


Figure 5.34: (a) Sketch of I_{gate} against V_{gate} for fixed V_{drain}, (b) Modelled V_{PP}^{che} with injected charge shaded, (c) Modelled $V_{PP}^{che} + offset$ with injected charge shaded.

above it. Hence, when the model value of V_{PP}^{che} is used, electrons are initially injected onto the floating gate, lowering V_{gate} and thus causing I_{gate} to fall away rapidly. This is shown as the shaded area in figure 5.34(b). However, with a positive offset, as current is injected onto the floating gate, I_{gate} actually increases to its peak before falling away. Hence the total charge injected can be significantly more than for the model value of V_{PP}^{che}. This is shown as the shaded area in figure 5.34(c). Of course, if the offset becomes too large, then insufficient charge is injected to allow I_{gate} to peak, and thus ΔV_{SO} begins to tail off again as found.

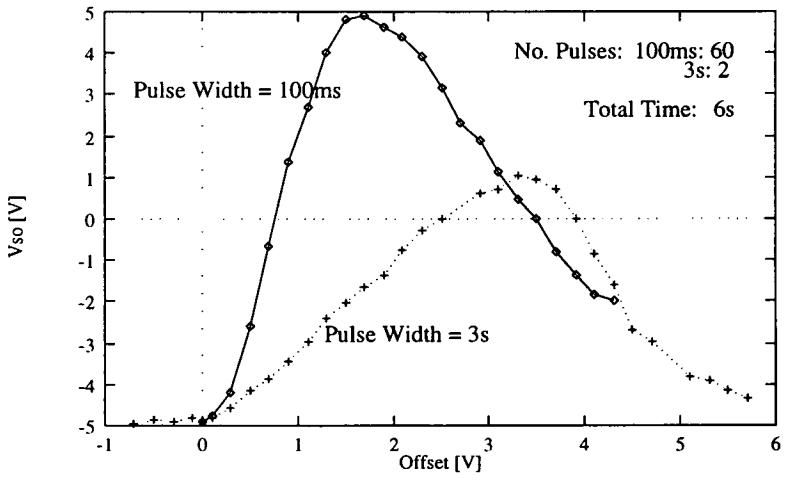


Figure 5.35: Final V_{SO} for a variety of offset voltages after a total of 6s inject time comprised of sixty 100ms pulses or two 3s pulses, where V_{SO}(time = 0) = -5V.

To confirm this idea, the experiment was re-run with pulses of 100ms instead of 3s (ie. V_{PP}^{che} recalculated thirty times more often during injection). As can be seen from figure 5.35, the largest change in V_{SO} has moved closer to the zero offset axis as expected since I_{gate} is returned to its approximate maximum thirty times more often. Also, due to the increased frequency of maximising I_{gate}, the maximum V_{SO} is higher for 100ms pulses than for 3s pulses. However this does not necessarily imply shorter programming pulses are better, since the *total* programming time does not only include the inject time but also the V_{SO} measurement time (which, with 100ms, is consequently thirty times longer).

A suitable compromise is required according to the programming algorithm; in general, a small voltage offset added to the model is preferable to a shorter programming pulse, but this can be used for fine-tuning the programmed value.

5.6.1 Current Limitation

The CHE programming so far observed is impractically slow. However it does show a strong V_{drain} dependency and so CHE might be made usable with increased V_{drain} provided avoidance of snapback breakdown can be achieved.

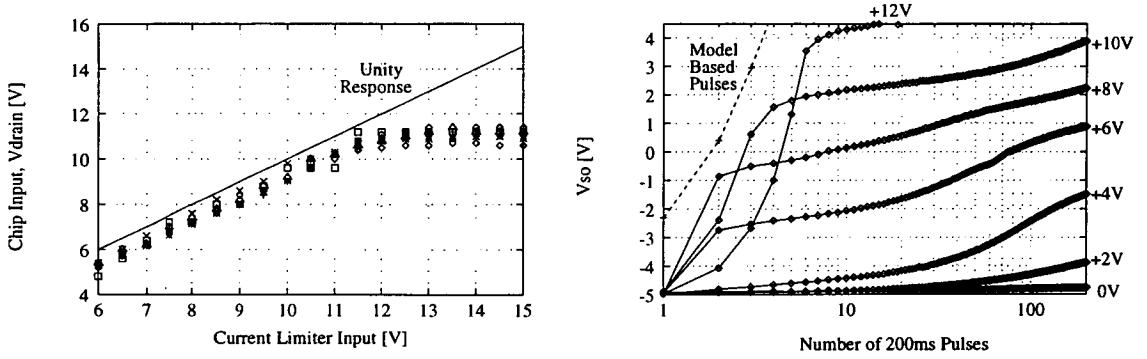


Figure 5.36: (a) Voltage response of Siliconix current limiter for a variety of test devices. Zero volts were coupled in through the control capacitor but V_{SO} was preset to < -8V for these experiments (b) V_{SO} programming characteristics with $V_{drain}^{CHE} = 10$V; the solid lines represent programming with a constant voltage pulse (of labelled magnitude) being applied to the control terminal.

Since the snapback breakdown damage would appear due to excessive currents, a

2.4mA current limiter[30] was placed between the chip V_{drain} terminal and the supply (set at V_{drain}^{CHE}). 2.4mA was found by experimentation to be sufficiently large to allow hot electron programming whilst affording adequate breakdown protection up to 15V on all floating gates tested – with the current limiter in place no more devices were destroyed by testing.

Any avalanching which occurs in the drain-channel depletion region is now limited and current cannot reach destructive levels. Since currents of only $\sim$ 1.5mA were measured during the application of V_{PP}^{che} a complete explanation of this behaviour has not been found (since 2.4mA would be expected if avalanche currents were being directly limited). Since the snapback breakdown voltage minimum is much lower than the V_{PP}^{che} applied (see section D.5), it is possible that destructive currents only occurred during rise times and it is these which are now being limited. Whatever the explanation, there was clearly an improvement in device reliability with no further failures above $V_{drain} > 8$V with the introduction of the current limiter.

The slight potential drop across the current limiter is shown in figure 5.36(a) which shows that it traces V_{CHE} up to about 12V (and that the drop is not large enough in itself to explain the reliability improvement). Faster programming was indeed found with higher V_{drain} although, as figure 5.36(b) shows, model based programming is still faster than coupling a constant voltage onto the control terminal.

5.6.2 Self-Induced Programming

CHE should still occur *without* the requirement for capacitive coupling of high voltages onto the floating gate provided ϕ_{fg} was high enough to promote hot electrons towards the gate. This would permit electron injection as well as FNT electron removal (with a sufficiently small m_t) to occur whilst simultaneously reading the floating gate since no signals would be capacitively coupled onto it. This would provide a suitable mechanism for bi-directional feedback programming or dynamic programming of the floating gate (this idea was initially proposed by Diorio *et al* - see section 7.2.3).

To test this proposal ϕ_{fg} of some *TARDIS* devices were preset high ($\phi_{fg} \sim$ 9.5V $\rightarrow$ 12.5V) by tunnelling through the gate oxide (interpoly tunnelling could not support tunnelling to such a high level of ϕ_{fg} without breaking down) using V_{PP} of 40-45V. This is on the margin of gate oxide breakdown.

Figure 5.37 then shows the V_{SO} drift over a period of seconds subsequent to the application of a drain voltage immediately after presetting. The 0V drain voltage compares how V_{SO} decays by self-induced *tunnelling* of electrons onto the floating gate.

[30] A Siliconix constant current diode was used. The built-in current limiter of the power supply was insufficient since it contains an internal large stabilising capacitor across the terminals which can produce a current spike whilst discharging that is sufficiently large to damage the device.

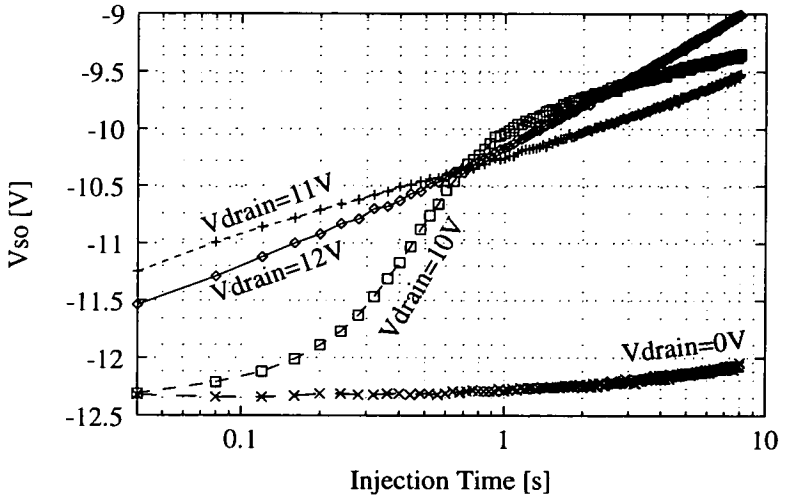


Figure 5.37: V_{SO} drift by self-induced CHE electron injection onto the floating gate subsequent to immediate application of V_{drain} (labelled).

While self-induced tunnelling then contributes slightly to the V_{SO} decay, it is clearly dominated by self-induced CHE injection in the presence of a high drain voltage.

As seen in figure 5.37, the self-programming of V_{SO} is rather slow and saturates within a small range in linear time. Enhanced programming was considered by setting $V_{drain} \sim V_{fg}$. A circuit to achieve this is shown in figure 5.38(a). However since again this circuit did not exist, it could be approximated in discrete time using a model based on equation 5.6:

$$V_{drain} = V_{fg}(ON) - m_c V_{SO} \tag{5.43}$$

Figure 5.38(b) shows that while this approach did not degrade range or speed, no improvement was evident (for any offset). This is probably because the higher hot electron population due to the higher drain-channel electric field of a high fixed V_{drain} compensates for the lower injection efficiency due to V_{drain}-V_{fg} mismatch.

Figure 5.39 shows the self-induced CHE programming of a selection of test devices which have been preset with a 45V 100ms gate oxide tunnelling pulse. Although the preset characteristics are consistent, the programming range available in a short time is small (and has a spread dependent on the preset tunnelling characteristics). Additionally the high ϕ_{fg} required makes interfacing to standard 5V neural processing circuits inconvenient. For this reason it was decided not to pursue this design further.

However Diorio *et al* have implemented innovative single transistor synapses [55] using their floating gates (and have also introduced a p-channel version which requires

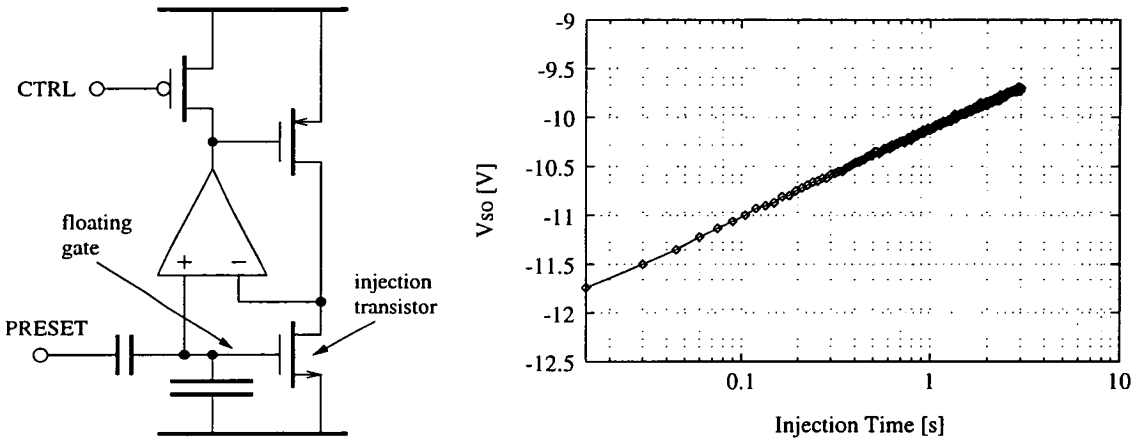


Figure 5.38: (a) Continuous time V_{drain}-V_{fg} follower circuit, (b) Programming characteristics of discrete time model.

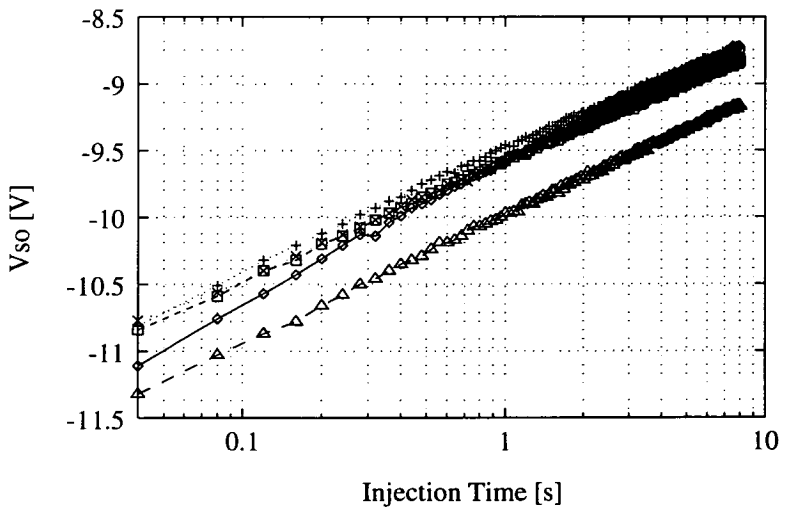


Figure 5.39: Self-induced CHE programming characteristics of a selection of test devices.

no special implant since CHE is naturally inherent to its weak inversion biasing[31]). These synapses have a continuous time learning rule based on the tunnelling and injection physical characteristics thus implementing a slowly adaptive neural network. However the slowness would make weight downloading, as is the aim of this project, prohibitively time consuming (ie. tens of minutes per weight).

5.6.3 Aging and Retention

The aging during CHE was slower than for interpoly tunnelling. Although it may be similar to that of gate oxide tunnelling due to the inherent built-in defect density of the oxide, the lower oxide electric fields required by CHE must be played off against probably a more localised injection current due to the electric field distribution in the channel. Unfortunately there was insufficient time to investigate this further.

Power-up retention tests showed the same general trends as seen earlier for tunnelling programmed floating gates indicating that the electron injection mechanism did not worsen the retention characteristics.

5.6.4 Polarised Detrapping

One further unexpected observation occurred when CHE programming was finished. A tunnelling capacitor had been used to preset the floating gate (removal of electrons) through uni-directional FNT. When Fowler-Nordheim experiments were resumed (ie. bi-directional FNT in tunnelling capacitor) there was a clear reduction in V_{fe}. A further investigation of this behaviour is shown in figure 5.40.

It is seems likely that bipolar operation of the tunnelling capacitor is causing field-activated detrapping which also limits the rate at which the net accumulation of oxide charge can age the device.

5.7 Discussion

TARDIS test devices have shown that it is possible to build programmable floating gate memories in the Alcatel-Mietec 2.4μm CMOS process. FNT occurs at a lower potential in the interpoly oxide than in the gate oxide and at potentials lower than planarised tunnelling theory would predict but still at voltages much greater than the specified maximum power supply. Electric field enhancement at asperities is suspected of causing the greater tunnelling currents although alternative explanations may

[31] Energetic electrons are created when accelerated channel holes collide with the semiconductor lattice liberating ionised electron-hole pairs.

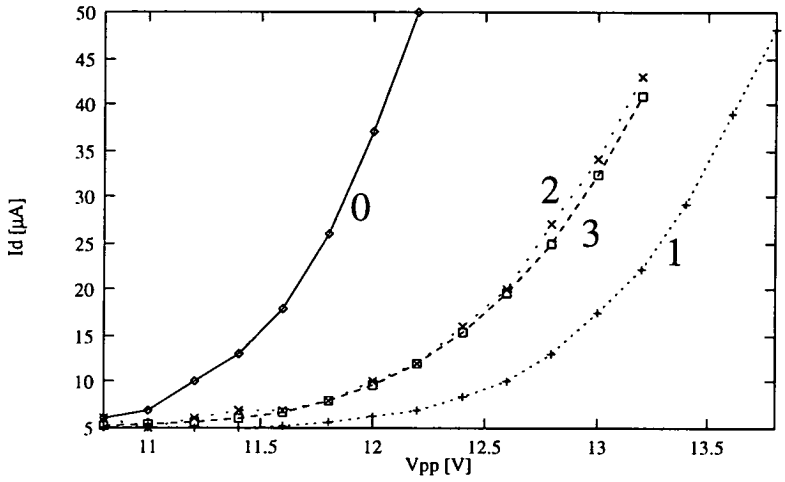


Figure 5.40: Trace (0) shows the programming of I_{dz} in an unaged device, (1) shows the same programming sequence after the device has been aged through 30 programming cycles with hot electron injection from the channel of M_1 and electron removal by FNT as normal (ie. unipolar FNT). Trace (2) shows the programming sequence after a further 10 cycles, this time with bipolar FNT for programming, and trace (3) after a further 30 cycles.

exist. Some tunnelling enhancement results from overlapping the poly2 over poly1 in violation of the layout design but the reasons for this, possibly edge-related asperities, are unclear. There is clearly no corner enhancement due to rounding of corners during etching. The tunnelling-enhancement behaviour is evidently a complicated mechanism but the simple interpretations and rules drawn from *TARDIS* experiments should be sufficient for use of this behaviour in analogue neural memory cells.

Trap up related aging is a significant problem, meaning no one programming cycle is repeatable and leading to intrusive, destructive testing. Inter-device variability is large and can be exacerbated by aging. Temperature treatment can mitigate some trapping but only for a few cycles.

Programming characteristics are naturally extremely non-linear. While this may be mitigated to an extent by linearization circuits described in section 3.5.5[32] , these cannot account for inter-device variability and aging and so would be unsuitable for the *absolute value* weight downloading scheme envisaged. Therefore iteration or feedback is required for programming analogue values. A simple empirical programming

[32]The method involving a high resistance floating gate is not viable due to the absence of sufficiently high resistance polysilicon (ie. $R \sim 1G\Omega$ [117]) in SLV-CMOS. Alcatel-Mietec 2.4μ CMOS can provide polysilicon resistors of either $\sim 20\Omega/\square$ or $\sim 2k\Omega/\square$ (but not both in the same design). However, special SRAM processes sometimes offer resistances of up to $\sim 10M\Omega/\square$ [52] (used for passive pull-up devices in inverters).

algorithm has been implemented and been demonstrated successful in programming a current reference and a Euclidean distance calculator circuit. Programmed floating gates show a relaxation behaviour which can hinder precise programming. This, standard packaging and possible ESD damage, have hampered retention testing although some degree of non-volatility is evident from less rigorous measurements.

Channel hot electrons have also been demonstrated to provide unidirectional programming and have the potential for operating within the specified power supplies. However a model-based programming algorithm and large currents (with surge protection) are required to allow programming times comparable with FNT.

Chapter 6

NEMO: Non-Volatile RBF Subcircuits and Addressing

6.1 Introduction

Having established the feasibility of SLV-CMOS floating gates as analogue non-volatile memories in the given Mietec 2.4μm fabrication process and now equipped with better knowledge of their layout requirements and programming voltage magnitudes and currents[1] it was possible to progress the aim of building a non-volatile analogue VLSI RBF by meshing floating gate cells with concurrent research into RBF functional subcircuits [110]. The vehicle for this work was the $NEMO^2$ chip.

6.2 Design Motivations

The motivations for the *NEMO* chip were:

1. Build on the success of non-volatile analogue memories examined in *TARDIS* to construct RBF component blocks with non-volatile weights with a view to implementing a complete VLSI RBF with programmable non-volatile memory. Modify existing circuits and evaluate new ideas for implementation of the RBF-functional elements.

2. Investigate the problems behind, and design, implement and evaluate schemes for array-based programming of floating gate cells within SLV-CMOS to facilitate neural networks of medium to high component density.

3. Examine data retention properties of SLV-CMOS floating gate memories to overcome the failure to properly obtain this data from *TARDIS*.

[1] With, of course, the caveat that *TARDIS* chips came from a single wafer; subsequent designs would be subject to any inter-wafer variability in undocumented characteristics, such as surface asperities believed to dictate tunnelling potentials.

[2] Non-Volatile Euclidean & Multiplier Options.

6.3 Issues in Programming Hardware ANNs

This section introduces two concepts which are important in understanding the issues and experimental results to be discussed later. The first is that of 'chip-in-the-loop' (CIL), a widely-used technique to program analogue hardware ANNs. The second concept is that of programming fidelity, which explains the terminology used in later sections.

6.3.1 Chip-in-the-Loop (CIL) Programming

ANNs require many iterations of high accuracy weight updates during training but feedforward (recall) operation can require a much lower accuracy [17]. On-chip training (see section 3.3.6) additionally requires potentially complex circuitry which would be idle subsequent to training.

Consequently, if a static weight set is sufficient, determination of the weights is best performed in a software model of the ANN chip. The weights would then be digital-to-analogue converted and downloaded to the chip.

The effectiveness of this approach is largely dependent on the problem and on the efficacy of the software model. However, even the most accurate model cannot accommodate the effectively random intra- and inter-chip device variation and the particular downloaded weight set will thus always be sub-optimal for the exact form of the feedforward net actually implemented by the hardware. This can be considered as a shift in the error surface away from the software determined minimum.

Fortunately it is possible to recover ANN performance by 'trimming' out this shift by augmenting software training with subsequent training using the chip instead of the model for the forward pass. This is the CIL technique and has been demonstrated as successful in improving a downloaded weight set in analogue VLSI ANNs (for example see [64] or [59]). Cairns [17] and Jackson [86] have demonstrated that CIL can be achieved in pulse-stream VLSI MLP implementations in only a few ($\sim$ 10s) of epochs for problems using real-world data. Recently Mayes [110] has demonstrated similar results for the RAM-refreshed pulse-stream VLSI RBF implementation, although on artificially-generated training data.

The procedure proposed here is then to apply CIL training to a floating gate pulse-stream VLSI RBF network. The inherent aging of the floating gate devices should not preclude sufficient iterations to complete the training. The aim here then is to produce RBF subcircuits which can be conveniently *parameterised*, by floating gate

programming, to desired weight sets so to allow such training to occur[3].

Having observed the extreme non-linear and non-stationary FN and CHE programming behaviours in *TARDIS* it was obvious that a simple relationship between target RBF parameters (multiplier weights, centre positions and widths) and programming signals would not exist. Therefore the goal of downloading RBF parameters from a trained software model could only be achieved in two ways:

1. Designing RBF subcircuits with integral continuous-time feedback which would allow establishment of target parameters values.

2. Using discrete-time feedback involving multiple pulse-evaluate cycles (as was used for programming *TARDIS*) to establish target parameter values (iteration).

It was observed that the iterative programming approach (which was selected by the designers of ETANN) has several advantages:

- Smaller circuits: these allow higher integration density. This is important when it is considered that feedback is only required during programming which may be very small part of lifecycle, for example in a write-once-read-many-times (WORM) network.

- The programming algorithm not hardwired into feedback circuits but is incorporated in controller state-machine (probably in software). Thus easily modified eg. to emphasise speed over resolution or vice versa.

For these reasons, the cells in *NEMO* were designed to be programmed iteratively. Both these points also have relevance in the context of the project: since these were to be the first floating gate RBF subcircuits designed, lower complexity was favoured as it would reduce scope for error. Additionally the programming algorithm is a topic open to investigation on an iteratively programmed chip.

However, two *disadvantages* were observed to exist with the iterative approach:

- Slow speed: many pulse-evaluate cycles can mount up to very long program periods; this would make continuously adapting systems impossible and many-epoch chip-in-the-loop training sessions impractical. Notice that the length of the evaluation phase is intimately connected with the cell design; measuring a single voltage is rapid, searching input-space to find a specific output is not.

[3]The aim of *NEMO* is to demonstrate weight downloading in single cells and small test arrays, not to build an entire functioning RBF as this was considered too large a step to take from the knowledge derived from *TARDIS*.

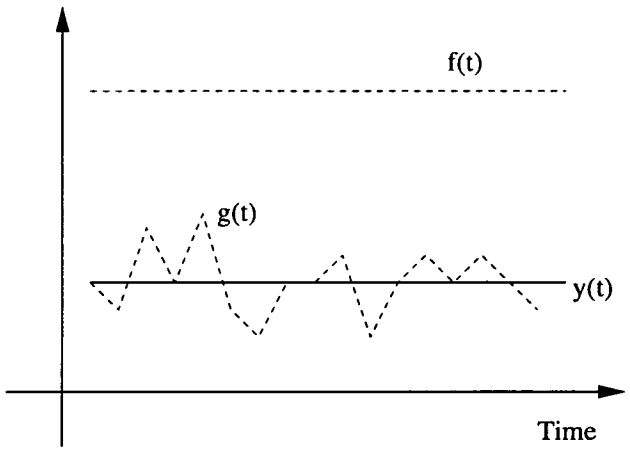


Figure 6.1: Distinction between *precision* and *accuracy* of a signal.

- Fast aging: since multiple pulses are required per programming attempt, devices age correspondingly more quickly limiting the useful lifetime of the chip.

Because these disadvantages are significant ones, it was deemed important to also investigate a continuous-time feedback approach to programming. The vehicle for this was the *PARAFIN* chip which is discussed in the next chapter.

6.3.2 Programming Fidelity

The fidelity of an ANN is manifested in two (often confused) ways using the definitions proposed by Kirk [91] and quoted in [60]:

- **precision:** "the degree of agreement of repeated measurements of a quantity"
- **accuracy:** "the degree of conformity to some recognised standard value"

This distinction is illustrated in figure 6.1. Function $y(t)$ is an ideal target value, which $f(t)$ and $g(t)$ attempt to match. $f(t)$ is a *precise* approximation (as it is entirely repeatable) but it is not accurate; $g(t)$ is not precise but it is much more *accurate* with a mean value close to $y(t)$. Digital neural networks (implicitly including software neural networks) are always extremely precise but their accuracy depends on the number of bits used for numerical representation. Analogue circuits however, may suffer imprecision (due to circuit noise) in addition to inaccuracy (due to circuits offsets and memory specific issues)[4]. It is the task of the designer to attempt to minimise both imprecision and

[4]Loosely, analogue accuracy is the conformity of the signal mean, and analogue precision is the degree of repeatability of the mean (ie. the signal distribution); exact statistical definitions have yet to be standardised.

inaccuracy. Imprecision can often be substantially reduced (although not eliminated) and good design can lead to high accuracy.

In case of a tradeoff, it is worth also noting from [60] that

- High precision is often not required if hardware attains high accuracy.
- A larger network can often overcome problems of low precision.

Additionally, it is usually possible to employ some technique which reduces the required bit accuracy for a given problem.

In digital ANNs, it has been found that in typical benchmark problems feed-forward accuracy is generally required to be much lower than the accuracy required during training where small weight updates are required [78, 7]. In analogue ANNs, the situation is more complicated, with researchers commonly describing circuit fidelity in terms of *digital bit equivalence*, where circuit noise is often treated as quantization noise (ie. precision and accuracy are not well distinguished). This can lead to misleading representations: analogue ANNs may perform better than digital ANNs of the estimated number of bits. However, again, training is found to require higher digital bit equivalence than for the feedforward pass.

The topic of analogue imprecision/inaccuracy is an involved one and beyond the scope of this thesis. Additionally, the issue is very much problem-specific. It is impossible to say just how precise and just how accurate the RBF circuits and their programming support must be. The RBF components are mostly adapted from their dynamic capacitor implementations (which have been shown to compose a successful RBF classifier). However, design of floating gate interaction and programming support is a tradeoff between circuit size, complexity and design time.

As a crude estimate of the fidelity of these circuits the common digital bit equivalence metric will be used but it must be interpreted with care for the reasons outlined here.

6.4 Chip Design Issues

The section discusses some of the important issues which had to be considered during the design process for *NEMO*.

6.4.1 Choice of Floating Gate Designs

With *TARDIS* it was observed that interpoly tunnelling capacitors offered lower programming voltages than for gate oxide tunnelling. For this reason they were selected for the implementation of floating gates on *NEMO*. Although significant trap-up was

observed, it was not severe enough to preclude the few ($\sim$ 10s) of CIL cycles envisaged.

Additionally, since a smaller tunnelling oxide area (ie. concentrated about asperities) allows a higher oxide defect density [164], this enhances the prospect of a higher yield.

Since the poly2-poly1 overlap was found to result in a drop in field-emission tunnelling threshold but no benefit have been found in the use of elaborate long perimeter or multi-cornered tunnelling structures, tunnelling capacitors were designed as simple poly2-poly1 overlapping rectangles.

6.4.2 RBF Components

As seen in chapter 2, the components of a classical RBF are Euclidean distance calculators, nonlinearities and multipliers. It was also shown how such functionality may be implemented in pulse-stream VLSI.

The aim here then was to *adapt* these circuits for compliance with floating gate storage. As shown in equation 5.4, the floating gate voltage is a combination of both ϕ_{fg} and the voltages capacitively coupled in from surrounding circuit nodes. Here during evaluation the capacitively coupled nodes are set to zero (excluding parasitics). Whilst this reduces somewhat the computational richness afforded by the floating gate structure as it sets $V_{fg} \sim \phi_{fg}$, this allows the stored charge to become simply equivalent to the charge dumped on storage capacitors in the refreshable versions of the RBF subcircuits and thus makes the transition to non-volatile versions both easier and faster. It may be possible to exploit the fuller behaviour of the floating gate to build smaller or better circuits but this would require a much more extensive redesign.

Unfortunately, due to pressure of time to the chip deadline (and also because its efficacy was at the time unknown), the nonlinearities were not investigated on $NEMO$. However, the Euclidean distance calculators and the multiplier were included.

6.4.3 Programming Arrays

Simply attaching floating gates where dynamic weight storage capacitors previously were in the RBF subcircuits and connecting the programming lines directly to pads is not difficult - and therefore these were implemented to allow direct behavioural investigation (pinned-out circuits)[5] . However, pinning out every floating gate quickly

[5]It was considered expedient to incorporate into $NEMO$ floating gate RBF subcircuits which were directly pinned out (like $TARDIS$ test devices); these would allow functionally evaluation of the combination of RBF subcircuits with floating gate memory in the absence of the further complexity (and potential fallibility) of experimental programming schemes.

becomes impractical for more than a trivial number of floating gates, n, since the number of access pads rises in O(n). This has a number of serious disadvantages:

- Wasteful use of silicon budget; designs heavily pad limited.
- Expensive packaging costs (roughly linearly dependent on the pad count) .
- Addressing requirements not eliminated, merely deferred to board-level where component costs are higher. Additionally, loss of integration which impacts upon applicability and manufacturing costs.
- ESD protection is difficult to implement and must be used on every addressing pad.
- Network size heavily limited by largest available package.

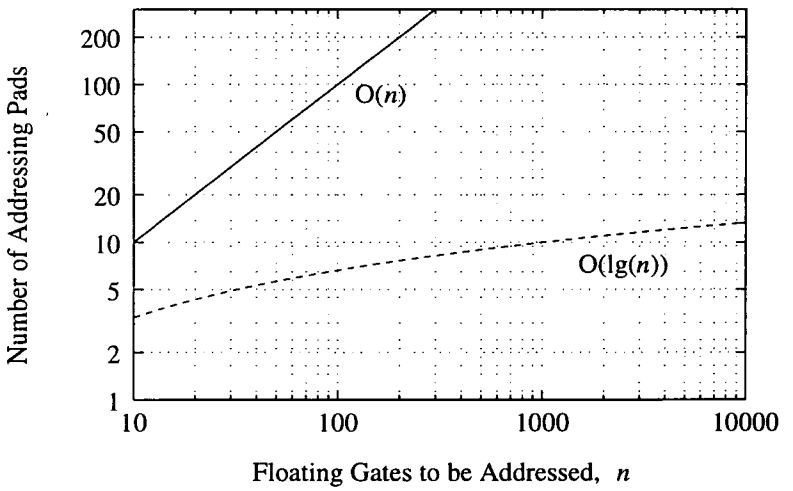


Figure 6.2: Pads dedicated to programming of floating gates for directly pinned-out, O(n), and digital addressing scheme, O(lg(n)).

Thus even for small networks it is not efficient to have the addressing and high voltage drivers off-chip. This then provides the rationale for the development of on-chip programming circuits. These allow floating gates to be accessed by means of a standard digital row/column address decoder, and as such, the pad requirement grows with only O(lg(n)). A comparison of these techniques is shown in figure 6.2. Single pads may then carry high voltages onto chips, or these may be generated on-chip using charge pumps (see appendix C) , whilst cell addressing may be by a standard digital address bus.

Three different schemes have been proposed, designed and incorporated on *NEMO* as individual test blocks. These are discussed in detail in section 6.7.

6.4.4 *NEMO* Floorplan

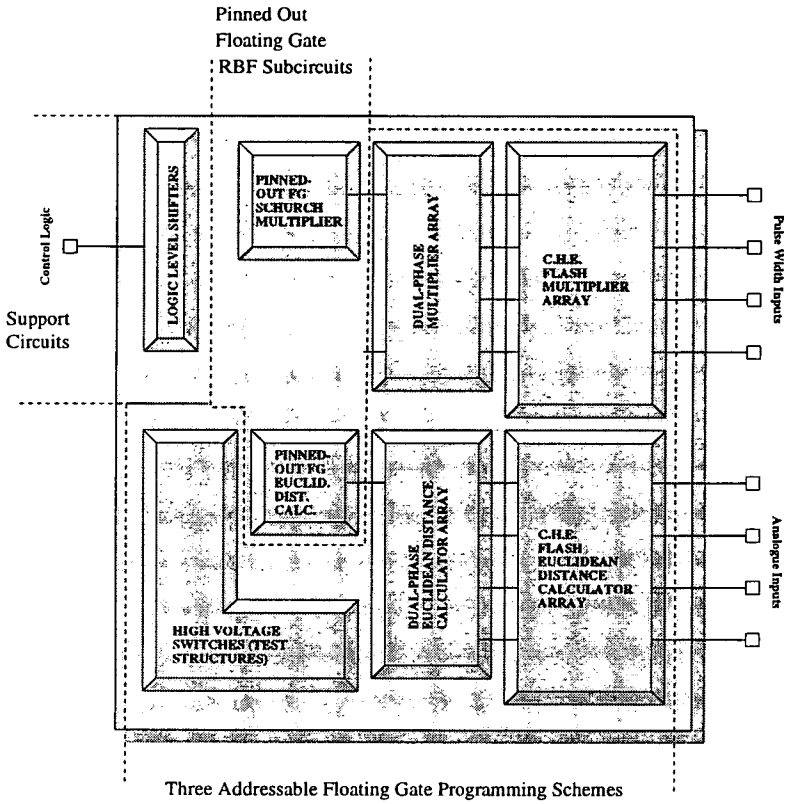


Figure 6.3: *NEMO* Block Diagram

Figure 6.3 summarises the contents of *NEMO* and shows approximately their location on chip. There is a directly pinned-out floating gate Euclidean distance calculator cell and a directly pinned-out multiplier cell, three different on-chip addressing systems and some support circuits (logic level shifters) for the addressing circuits.

6.5 Floating Gate RBF Building Blocks

This section describes and analyses the various subcircuit designs required to implement a floating gate RBF chip. It explores a number of issues concerned with implementing the circuits in VLSI and gives some experimental results of their performance.

6.5.1 Revised Floating Gate Layout

In *TARDIS* the shunt capacitance, C_{fb}, compromises the programming pulse coupling ratio (reducing the effective tunnelling potential for any given V_{PP}). It was there-

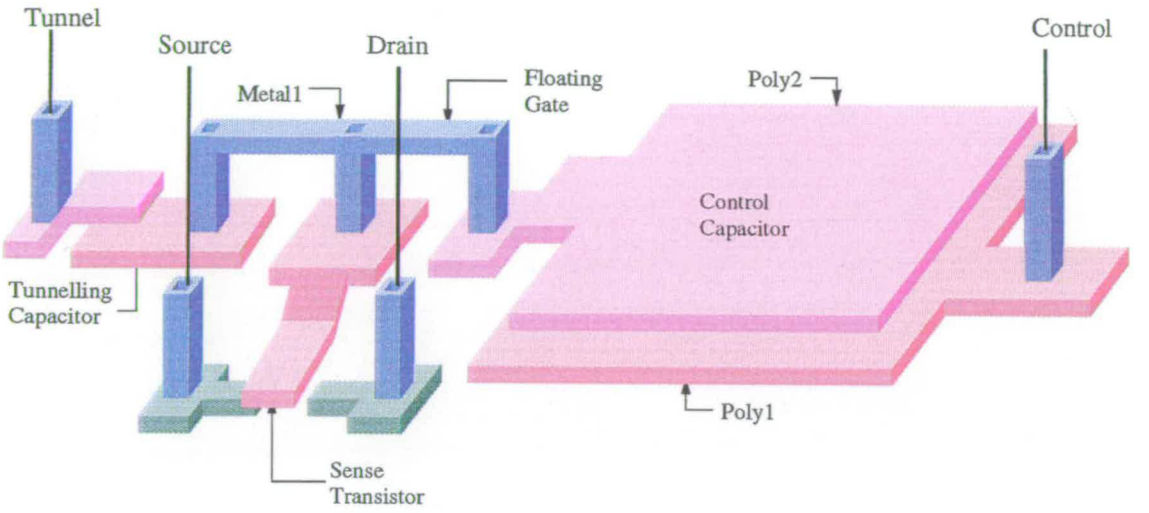


Figure 6.4: Cross sectional cut-away layout schematic for floating gate.

fore desirable to reduce this capacitance. C_{fb} is dominated by the parasitic capacitance between the lower (poly1) plate of $C_{control}$ and the substrate (which forms a thick oxide transistor). By instead using the *upper* (poly2) plate as the floating-gate-side connection as shown in figure 6.4, this greatly reduces C_{fb} since the large poly1 plate is no longer part of the floating gate. However, since it is not possible to connect poly2 directly to poly1, the connection to the sense transistor and tunnelling capacitor must therefore be through contacts to metal1 as also shown. The metal1-poly contact on the sense transistor is *not* stacked as in [96] as tunnelling is not being attempted in the gate oxide.

6.5.2 Euclidean Distance Calculator

As described in section 5.3, Collins, Marshall and Brown demonstrated that two floating gates can form a non-volatile analogue Euclidean distance calculator. Despite the elegance of this design, it was decided to use the 'home-grown' circuit described in section 2.3.1 for a number of reasons:

1. In-house experience and layout existed for the cell.

2. Only one floating gate was required per centre rather than two, with commensurate savings in addressing circuitry and iterative programming time.

3. No need to generate and propagate the complement of the input, $\overline{in}$.

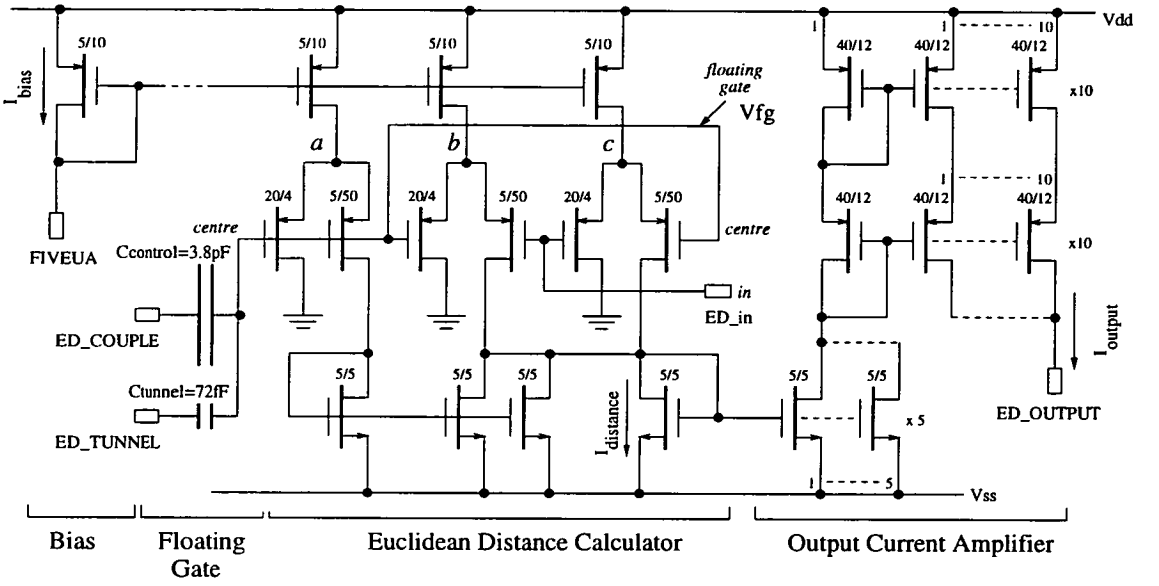


Figure 6.5: Directly pinned-out floating gate adaption of Euclidean distance calculator circuit with output current amplifier.

Circuit Design

The complete Euclidean distance calculator circuit using a directly pinned out floating gate to store the centre is shown in figure 6.5.

Hspice simulation of the distance circuit using precharged capacitors to model programmed floating gates led to an observed asymmetric distance current dependent on $C_{control}$ as shown in figure 6.6(a). This asymmetry was found to depend on a shift in the programmed floating gate voltage roughly proportional to the input signal, V_{in}. The error can be quantified by the definition

$$Error = \left(\frac{V_{fg}(V_{in} = 3\text{V}) - V_{fg}(V_{in} = 0\text{V})}{3\text{V(centre range)}} \right) \times 100\% \qquad (6.1)$$

Then figure 6.6(b) shows how the error corresponds to the magnitude of $C_{control}$.

This asymmetry can be explained in terms of parasitic capacitances associated with the floating gate. The three sets of ratioed pair transistors in the Euclidean distance calculator (marked (a),(b) and (c) in figure 6.5) are shown again in figure 6.7 along with the sense transistor parasitic capacitances connected to the floating gate. Since the 'fat' transistors dominate the voltage at the node connecting the sources of the ratioed pair, then in the compensation pair (a), this voltage is controlled by the floating gate voltage itself. Similarly, this is the case in ratioed pair (b) which defines half of the Euclidean characteristic (ie. when $V_{in} \leq V_{fg}$). Thus there are always approximately

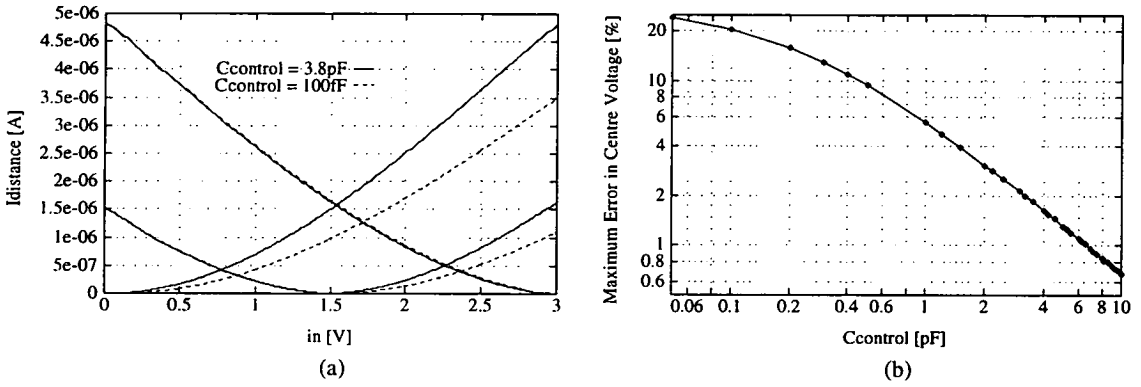


Figure 6.6: (a) Hspice simulation of the distance calculator circuit response for two different values of $C_{control}$ ($V_{fg}(V_{in} = 0V)$ = 0.0V,1.5V,3.0V),illustrating the extreme asymmetry for the smaller capacitance. (b) Error in $V_{fg}(V_{in} = 3V)$ against $C_{control}$ from simulation results.

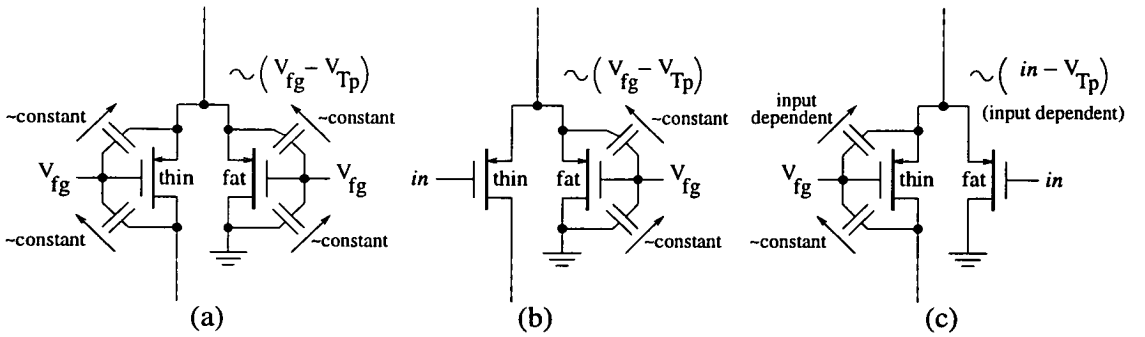


Figure 6.7: Parasitic capacitances and voltages associated with the floating gate for the three ratioed pairs (a),(b),(c) of the Euclidean distance calculator circuit shown in figure 6.5

constant voltages across these parasitic capacitors on the floating gate; these are not of concern (the drive voltage of the output current mirror covers a small range ($\sim 0.2V$) for $I_{distance}$: $0 \rightarrow 5\mu$A, and so again may be considered approximately constant). However, in ratioed pair (b), this node is controlled by the input signal, *in*. Thus an input-dependent voltage is parasitically coupled onto the floating gate distorting the other half of the Euclidean characteristic (ie. when $V_{in} \geq V_{fg}$). This consequently leads to the observed asymmetry. Notice the similarity of this discussion to that of section 5.3.

As shown in figure 6.6(b), the effect is mitigated by use of a larger $C_{control}$ since this increases the ratio of C_{fg} to $C_{parastic}$. This effect did not come to light in the original capacitor/refresh implementation since a large centre-storage capacitor was already required to limit the refresh rate. A simple solution for *NEMO* then was to use a larger capacitor (3.8pF reduced the error to $< 2\%$). This is, of course, expensive in terms of area, and a more application sensitive approach would have to consider how much of an error would be acceptable.

Functional Characteristics

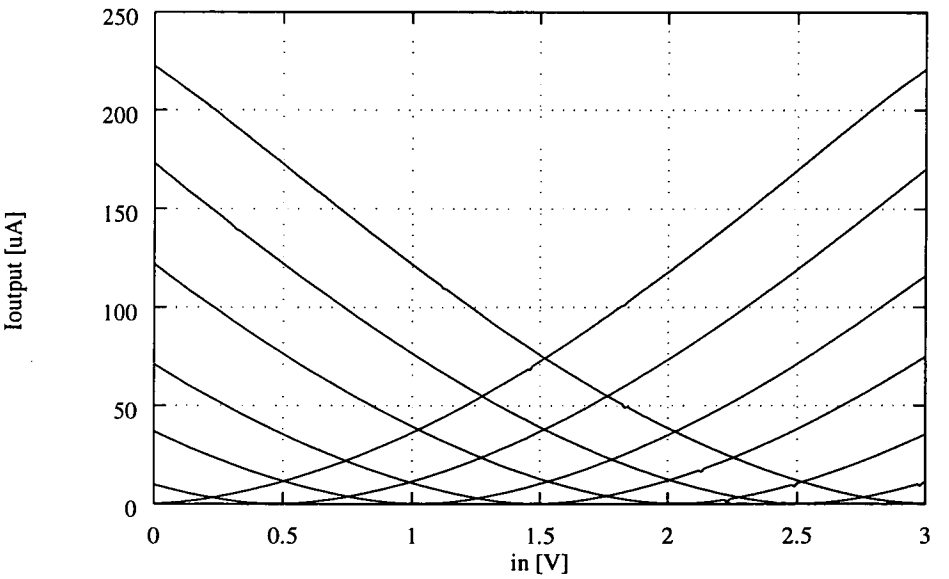


Figure 6.8: Typical measured output trace for distance calculator iteratively programmed to 0.0V,0.5V,1.0V,1.5V,2.0V,2.5V and 3.0V

The gain of the current amplifier across all chips tested was found to be in the range 43.1 – 45.1, slightly less than the designed gain of 50.0. This is probably due to transistor mis-match in the current mirrors.

Centre Measurement

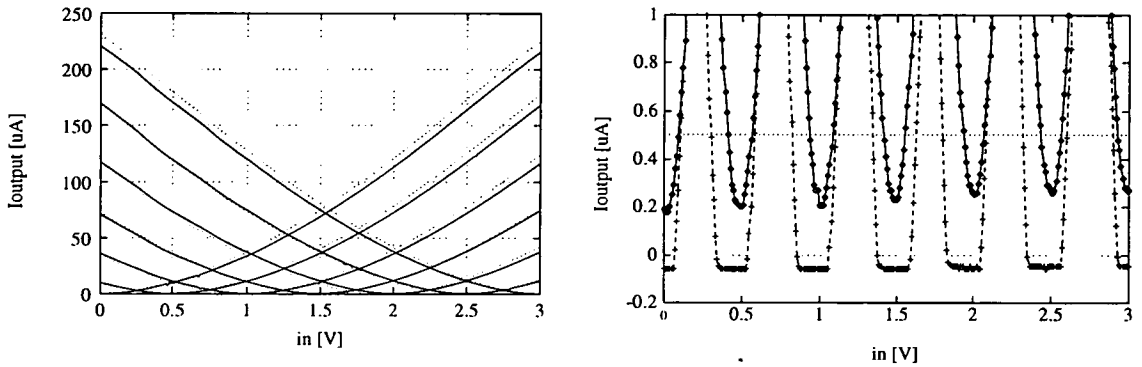


Figure 6.9: (a) Full-range functional characteristics of Euclidean distance cell, (b) 'Zoomed' view of I_{output} at the centres, $V_{fg} = V_{in}$.

Unlike the case with the capacitor/refresh circuits, the centre voltage is not known (it is 'hidden' on the floating gate). The value of the centre must therefore be measured (measurement is clearly required for programming as well as illustrative purposes) by inference from the observed overall circuit behaviour; whilst it would be possible to buffer the floating gate and directly measure its voltage, ie. V_{centre}, this would add to the cell size[6]. One advantage of this is that the actual characteristics, rather than the expected characteristics of the circuit are used (thus compensating for offsets). However, the measurement process is more complex and the evaluation cycle is longer (input-space search).

The simplest approach would be to locate the value of V_{in} which minimises I_{output}. However, for many devices this was found to be unsuitable. Figure 6.9 illustrates this problem using traces from two extreme chips. Although the full scale programmed Euclidean characteristics (a) look fairly similar allowing for inter-chip variation, zooming into the detail near the centres (b) highlights the problem. Here, I_{output} for one chip never reaches zero, whereas the other saturates for a significant range of V_{in} (The slight zero offset is inherent in the test board transresistance amplifier, not the distance circuit).

The effect can be explained quite simply due to cross-chip variation causing the requirement specified in equation 2.12 not to hold true. If this is the case, V_{in} $(I = 0)$ is slightly different for the two ratioed pairs, a and b, since the compensation pair is not effective in completely removing the offset from zero at $V_{fg} = V_{in}$. Figure 6.10 illustrates this problem schematically.

[6]Recall that direct measurement would only be of use during programming and the strategy of iterative programming was to avoid additional circuits for only this short phase so to minimise area.

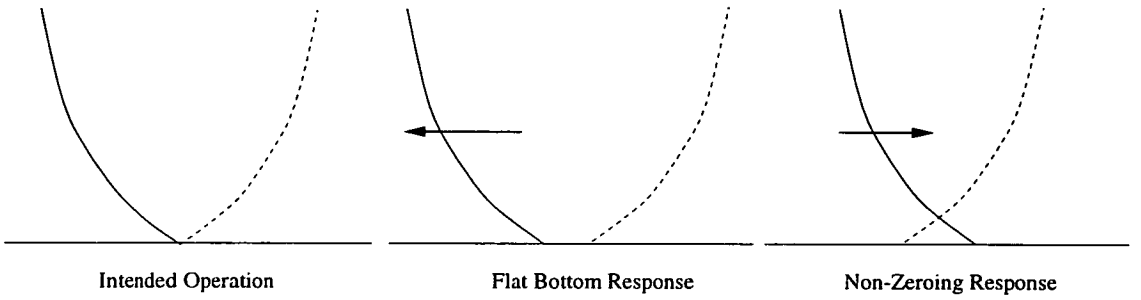


Figure 6.10: Distortion of Euclidean response due to mismatch between ratioed pairs.

The directly pinned out Euclidean distance calculator was tested on five chips. One behaved ideally, two had non-zero behaviour (offset 20nA and 200nA at $V_{fg} = 1.50$V) and the other two had flat bottoms (of widths 150mV and 220mV at $V_{fg} = 1.50$V).

This non-ideality had been obscured in the RAM-refreshed implementation of the circuit [1 10] since there had been no similar motivation to closely observe the operation of the distance current about the centre point. In terms of magnitude, the effect is a minor one and so is unlikely to be an impediment to overall RBF functionality, it demonstrates the redundancy of the compensation transistors in the Euclidean distance calculator since this type of mis-match behaviour cannot be removed.

It is therefore proposed that for a more compact design, that the compensation transistors are removed and the non-linearity circuit is replaced (as described in section 2.3.2).

Returning to the issue of centre measurement, A practical approach to measurement of the centre location had to be taken: experimental evidence showed that centres offset from zero were not offset by more than 0.5μA in any cell tested. Since a large floating gate capacitor was used to circumvent the source-coupling problem, the Euclidean characteristic can be expected to be symmetric, especially close to the centre. The centre was therefore defined as the midpoint between the two 0.5μA crossings for a V_{in} sweep of $0 \rightarrow 3$V. This approach gives consistent results for all ideal, flat-bottomed and zero-offset characteristics. In practice two sweeps were used: a coarse granularity sweep to define the centre locus and then a fine granularity sweep in the locus to resolve the exact centre; this reduced the number of measurements required whilst maintaining maximum resolution. Recorded measurements of the 'width' between the two 0.5μA crossings, allowed centre estimation at the extrema (ie. $\sim$0V and $\sim$3V) where only one 0.5μA crossing would occur.

6.5.3 Pulse Stream Multiplier

Circuit Design and Layout

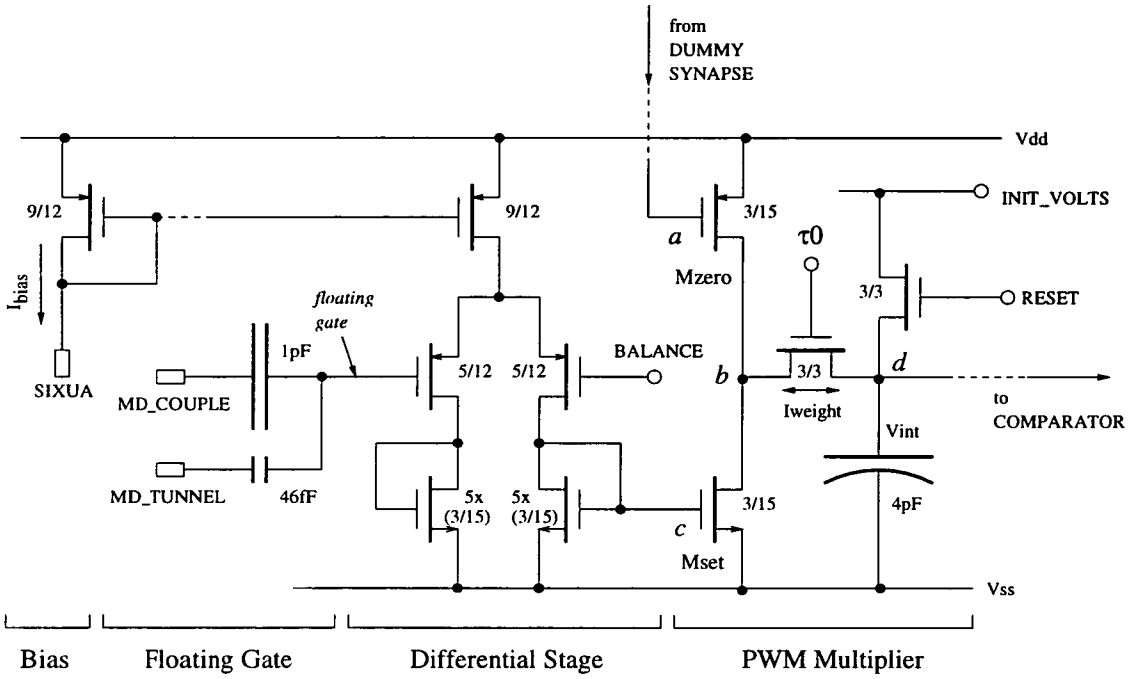


Figure 6.11: Directly pinned-out floating gate adaption of Pulse Stream Multiplier circuit.

The directly pinned-out floating gate implementation of the multiplier is shown in figure 6.11. The differential stage, although not necessary as the floating gate could directly drive M_{set}, was retained to:

- Provide buffering between the floating gate and the *sum* node which switches rail-to-rail and may be significant because of the high g_{ds}. (Typical `Hspice` simulations suggested a disturbance of up to 10mV on an M_{set} floating gate compared to 1mV with the differential stage in place).
- Increase the useful range of ϕ_{fg} (a lower transconductance can be achieved in a smaller area than with a weak transistor). The illustrated scheme has a useful ϕ_{fg} range of $0 \rightarrow 3$V, the same as for the Euclidean distance calculator.
- Introduce a much more linear $\phi_{fg} \rightarrow I_{weight}$ mapping to improve accuracy and simplify the programming algorithm. This mapping is simulated in figure 6.12 which shows an approximately linear mapping between 0.5V and 2.5V. However, since programming is iterative the full 0-3V range is available for use.

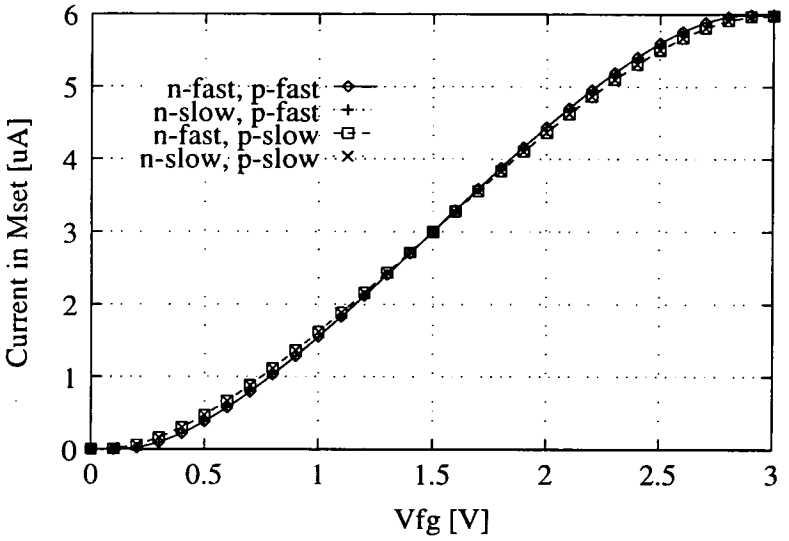


Figure 6.12: Simulated differential stage V_{fg} to I(Mset) mapping

Functional Characteristics: Multiplier Cell

Typical measured multiplication characteristics for a directly pinned-out multiplier are shown in figure 6.13(a) for a various iteratively programmed weights. Notice the charge-sharing 'kick' above $\tau_{in} = 0$. This is due to injection of charge from the parasitic capacitance at node b onto the explicit capacitance at node d when the switch transistor is turned on. Here, for $\tau_{in} = 0$, $\tau_{out} \approx 4.68\mu$s. This zero offset, along with zero symmetry and output range is one of this multiplier characteristics easily adjustable by off-chip biases as shown in figure B.3.

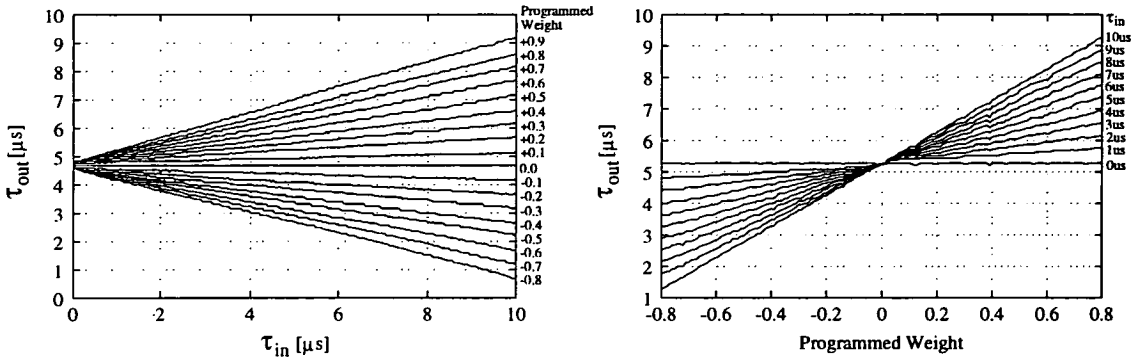
Weight Measurement

As for the distance cell, the weight, ω, of the multiplier must be inferred from observation. ω is the gradient of plotted line, τ_{out} versus τ_{in}. This gives rise to the definition

$$\omega = \frac{2}{N} \sum_{i=2}^{N} \frac{\tau_{out}[i] - \tau_{out}[i-1]}{\tau_{in}[i] - \tau_{in}[i-1]} \tag{6.2}$$

where there are N incremental inputs of $\tau_{in}[1] \rightarrow \tau_{in}[N]$. Thus this metric scales to any number of test observations. The 2 simply normalises the weight range to $-1 \rightarrow +1$.

The column running down the right side of figure 6.13(a) demonstrates the target programming weights for the experiment. Programming bounds were set at an accuracy of ± 0.01. In a second experiment, the multiplier was programmed for each weight in the range -0.8 $\rightarrow$ +0.8 in steps of 0.01. For each programming, the characteristics



(a) (b)

Figure 6.13: Measured multiplication characteristics (a) τ_{out} versus τ_{in} for various programmed weights (τ_{in} ramped from 0 to 10μs in 40ns increments), and (b) τ_{out} versus the programmed weight for various τ_{in} (ω programmed from -0.8 to +0.8 in 0.01 increments).

were recorded and this is plotted in figure 6.13(b) which demonstrates the 2-quadrant multiplication characteristics in the more conventional way. Once again, the charge-sharing kick above $\tau_{in} = 0$ is visible.

6.5.4 Fowler-Nordheim Programming Characteristics

Typical programming characteristics are shown in figure 6.14(a)-(b). Initialisation was by the algorithm described later in section 6.6. Although the $V_{fe}^{\pm}$ spread across chips was not studied explicitly this time, tunnelling characteristics did appear similar to those found on $TARDIS$ which pointed towards a run-to-run repeatability of tunnelling capacitor fabrication.

Typical programming characteristics for the pinned-out multiplier cell is shown in figures 6.14(c)-(d). Initialisation was also by the algorithm described in section 6.6. Although the characteristics are similar to those of the Euclidean distance calculator, they are clearly bounded between about -1 and +1, and ω updates are in the opposite direction to V_{centre} updates. Both effects are due to the action of the differential stage.

It has therefore been demonstrated that it is possible to augment the pulse-stream VLSI RBF subcircuits designs with floating gates and that is possible to program the parameters of these circuits across the full weight (or centre) range.

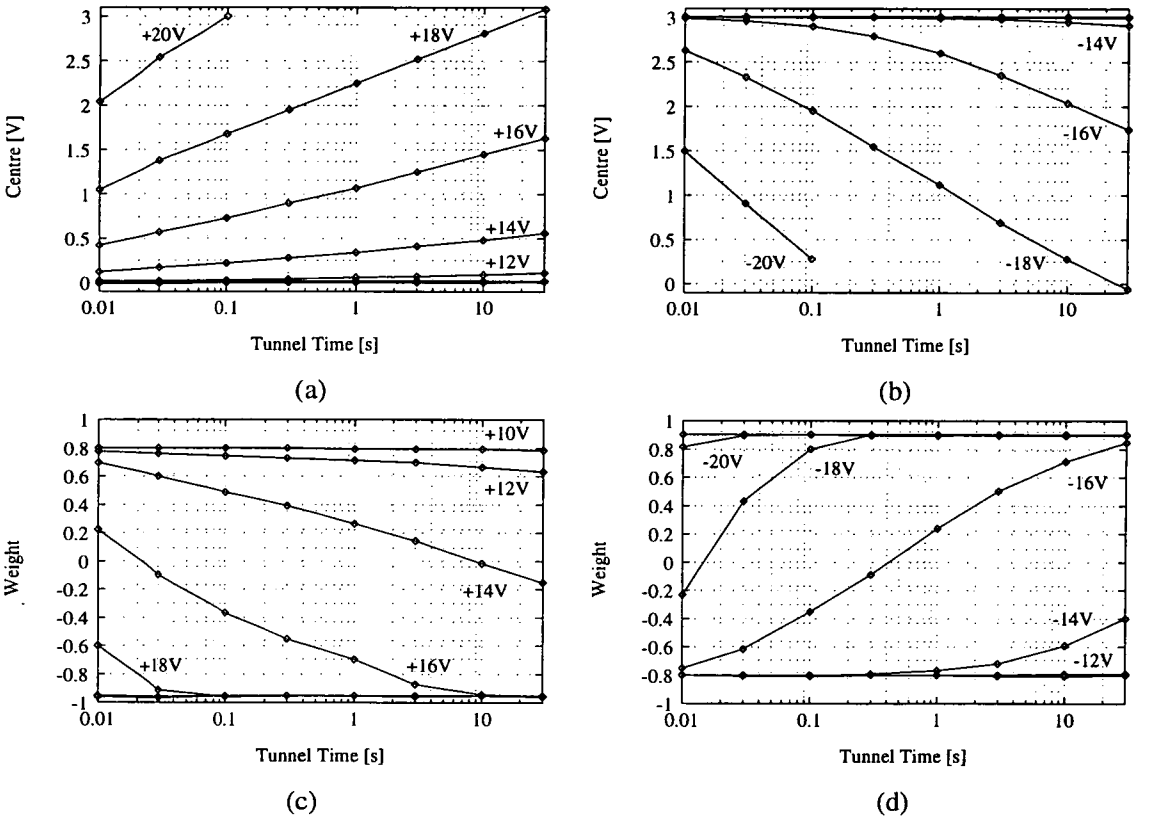


Figure 6.14: Typical programming characteristics: (a)-(b) V_{centre} in pinned-out floating gate Euclidean distance calculator, with $V_{centre}(0) = 0$V for positive pulses and $V_{centre}(0) = 3$V for negative pulses, (c)-(d) ω in pinned-out floating gate multiplier. The experiments were run in increasing $|V_{PP}|$ order with a reset to $V_{centre}(0)$ between each experiment. (Note that the reset voltage at $V_{centre}(0)$ is not shown on the graphs because of the logarithmic x-axis).

6.6 Programming Algorithm and Accuracy

The graphs from the previous section have already demonstrated that controlled iterative programming towards a target is possible. This section outlines the method by which this was achieved.

Programming proceeds then in a similar fashion to that described in section 5.3, with V^T_{centre} or ω^T replacing I^T_{dz} as the target value.

Programming *accuracy* in bits was defined in terms of acceptable voltage bounds on the value of V_{centre} or ω.

Defining $V_{fg}(range)$ as the range of floating gate voltages which form legitimate circuit parameters (ie. 0-3V in both the Euclidean distance calculator and the multiplier) then for a digital equivalence of N bits separation between targets,

$$\frac{V_{fg}(range)}{2^N - 1} \tag{6.3}$$

separation is required between the distinct target voltage levels (eg. for 2-bit equivalence, 1V separation is required with voltage targets at 0V,1V,2V and 3V). Thus a successfully programmed floating gate may deviate from its target value by no more than half this separation before an incorrect level is sensed. This sets tolerances (ξ) on the target value of

$$\xi = \frac{1}{2} \times \frac{V_{fg}(range)}{2^N - 1} \tag{6.4}$$

Programming is an issue of *short-term accuracy*. Theoretically it is possible to have floating gate accuracy limited only by the packet of charge carried by a single electron, achieving very high bit equivalence by iterating interminably. This is not a practical goal. Real, *long-term*, accuracy must allow for degeneration of the stored value due to slow detrapping, oxide depolarisation and any leakage currents once programming has finished.

6.6.1 Circuit Imprecision

In the short-term, it is hard to distinguish imprecision from inaccuracy. To measure system imprecision (floating gate, distance circuit *and* measurement hardware), four Euclidean distance calculators were each reprogrammed over two hundred times to a randomly selected target centre. Two measurements of V_{centre} were then made in rapid succession (attempting to eliminate any influence from long-term degeneration of the stored value). The spread of differences between the two measured values is shown in figure 6.15 distributed into 10mV separated bins.

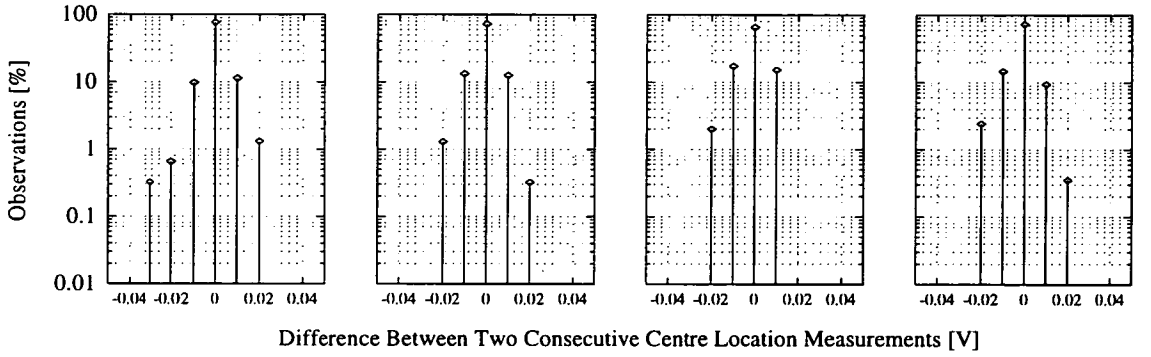


Figure 6.15: Centre measurement precision for four different Euclidean distance calculators.

From this experiment it is expected that in $\geq 95\%$ of cases, V_{centre} can be measured to a precision of ± 10mV. By rearranging equation 6.4 thus

$$N = \lg \left[\frac{V_{fg}(range)}{2\xi} + 1 \right] \tag{6.5}$$

it is possible to give a bitwise estimate of system precision, which is typically about 7 bits. So, for programming accuracy above about 7 bits, the immediate measurement may be expected to be sufficiently imprecise as to trigger a spurious target *hit/miss* response. This should not, however, detract from the programming results: the target must instead be considered within bounds of accuracy *and* limited precision, ie. that more precise centre measurement (eg. time averaged) is required for higher programming accuracies.

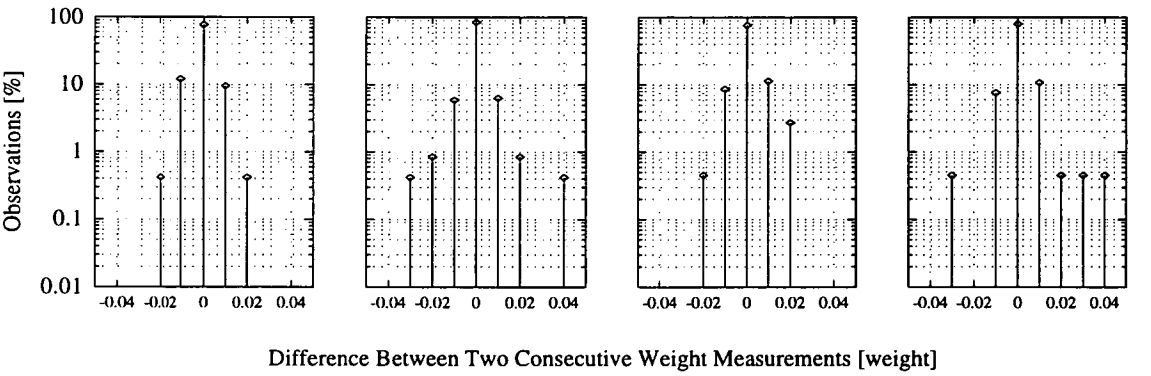


Figure 6.16: Weight measurement precision for four different multipliers ($\Delta\tau = 0.04\mu$s).

This experiment was repeated for four multipliers, as shown in figure 6.16. This

gives rise to the expectation that in $\geq 95\%$ of cases, ω can be measured to an absolute precision of ± 0.01. Using the multiplier equivalent of equation 6.5

$$N = \lg \left[\frac{\omega(range)}{2\xi} + 1 \right] \tag{6.6}$$

the bitwise system precision is estimated as typically $6\frac{1}{2}$ bits. This is slightly less precise than for the Euclidean distance calculator but it need not necessarily mean that the circuit is inherently less precise: the V_{centre} measurement involves two passes through V_{in} of a large and small voltage step which may extract more information than the few points read to obtain ω (due to the larger time overhead of having to download and upload test pulse widths). Clearly some of the imprecision must be apportioned to the measurement system.

6.6.2 Algorithmic Improvements

The *TARDIS* algorithm was tested for convergence at various resolutions by progressively selecting a random target to program towards according to

$$V^T_{centre} = \left(\mathcal{R} \mod 2^N \right) \times \frac{V_{fg}(range)}{2^N - 1} \tag{6.7}$$

where $\mathcal{R}$ is a pseudo-random integer. Thus programming was defined complete when

$$V^T_{centre} - \xi \leq V_{centre} \leq V^T_{centre} + \xi \tag{6.8}$$

Fifty targets were programmed to for each value of N in the range $1 \rightarrow 8$ bits of resolution. A typical programming sequence for a fairly aged device is shown in figure 6.17(a), with the mean and standard deviation for each of the fifty sequences shown in figure 6.17(b).

Although programming was found always to converge correctly to the target it was seen that the number of programming pulses required did not exhibit a strong dependency on the programming resolution. This is undesirable since it should be possible to choose between a fast, approximate programming and a slow, exacting programming dependent on the resolution requirements of the neural training or evaluation procedure. The reason for this lack of dependency can be seen in figure 6.17(b). Using the scheme described in section 5.3, the pulse magnitude can change only in increments of 200mV. Since the magnitude always starts at a default value below that required to cause programming, most of the programming pulses are wasted as the magnitude ramps gradually up to one which can support tunnelling. It is this ramping, rather than fine tuning V_{centre} which dominates the number of pulses.

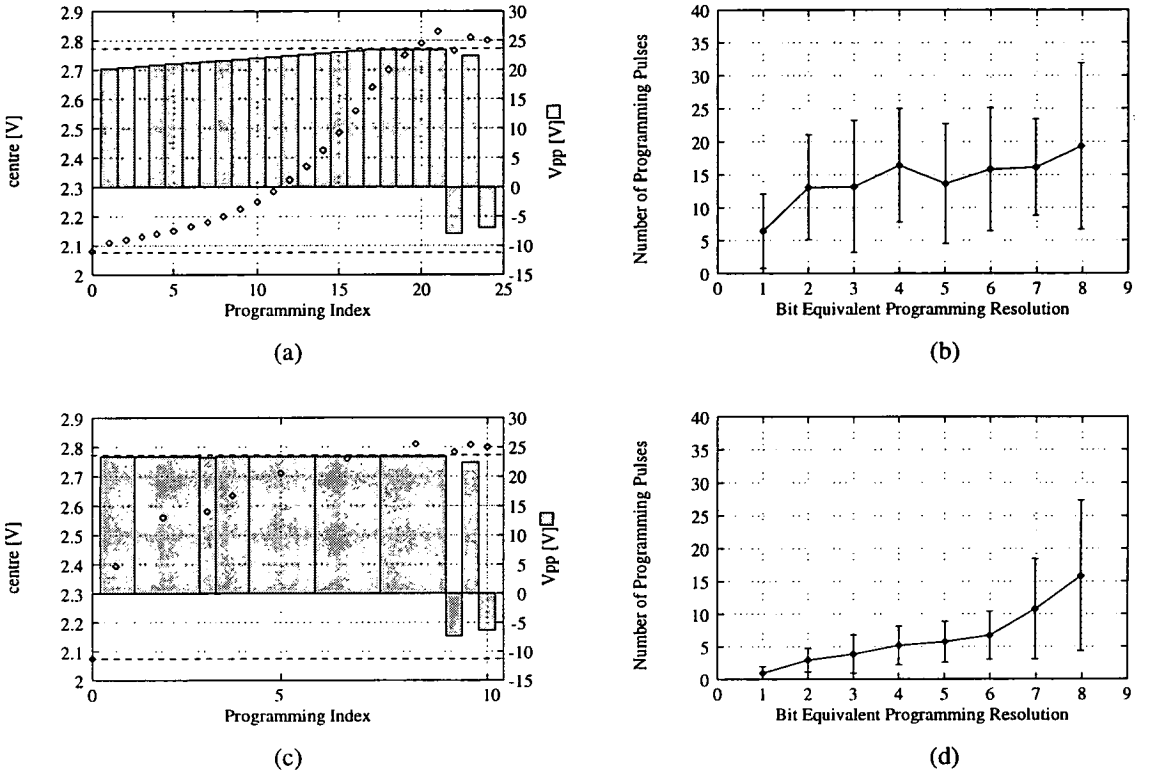


Figure 6.17: (a) Typical programming trace using the $TARDIS$ algorithm (programming centre to 2.80V at 8-bit resolution). Each bar represents a 100ms programming pulse. (b) Number of programming pulses (mean and single standard deviation) required for various resolutions of programming to random targets using the $TARDIS$ algorithm. (c) Typical programming trace using the modified $TARDIS$ algorithm (again programming to 2.80V at 8-bit resolution). Minimum width bar represents a 100ms programming pulse, and represented widths are scaled proportionately with applied pulse widths. (d) Number of programming pulses (mean and single standard deviation) required for various resolutions of programming to random targets using the modified $TARDIS$ algorithm. In both (a) and (c), the dashed lines represent the bounds on V_{PP} to prevent oxide breakdown. Notice that here during V_{PP}^{-} pulses, $V_{CONTROL}$ = 10V (this is similar to the dual-phase scheme described later). This allows smaller magnitude negative pulses. $V_{CONTROL}$ = 0V during read-out.

A number of modifications were therefore made to the *TARDIS* algorithm:

1. Calculated $V_{fe}^{\pm}$ values for each cell (eg. between chips or in an array) were retained to allow appropriate selection of default initial V_{PP} magnitudes close to that expected to induce tunnelling. $V_{fe}^{\pm}$ values were able to track observed behaviour and thus keep up to date with aging-induced deterioration.

2. ΔV_{PP} became a function of $V_{centre}^{T} - V_{centre}$ rather than arbitrarily 200mV, thus allowing rapid magnitude increments if a significant charge transfer was required. 200mV was retained as a lower bound to allow small movements in V_{centre} to be achieved within a reasonable time.

3. If the algorithm required a V_{PP} magnitude increment but V_{PP} was at the limit imposed to prevent oxide breakdown, then t_{PP} would be doubled instead. t_{PP} would be reset to 100ms if the magnitude required fell back below the limit. This allowed aged devices to be programmed as if with a long sequence of maximum magnitude V_{PP} pulses but without the time overhead of unnecessary intermediate V_{centre} evaluations.

Figure 6.17(c) shows a typical programming cycle with the modified algorithm, showing how the number of time-consuming evaluations cycles is reduced by (i) starting with a high V_{PP} due to prior observation of V_{fe} and (ii) rapid increments in t_{PP}, since the device is aged and V_{PP} required is maximal. The convergence experiment was repeated using the new algorithm, with typical results illustrated in figure 6.17(d). Clearly, not only had a resolution dependency been introduced as required but the number the pulses has been reduced for all resolutions, thus speeding programming throughout. Notice however, that this improvement is one of convenience only - since the 'removed' pulses were generally the ineffectual ramping ones which did not induce significant charge into the oxide, the aging characteristics (ie. V_{fe} vs. target number) are unlikely to have been improved.

This algorithm was also adopted for use in the perceptron arrays. Since positive global programming pulses increase ϕ_{fg} by removal of trapped charge, this leads to an increase in V_{centre}. However, this increase in ϕ_{fg} in the perceptrons leads to a *decrease* in the ω as defined earlier. Therefore the pulse polarity of the algorithm had to be reversed. Notice that although in both cases $\phi_{fg}(range)$ is 0V $\rightarrow$ 3V, there is a direct mapping between ϕ_{fg} and V_{centre}, the less direct mapping between ϕ_{fg} and the multiplication ω had to be accounted for.

6.7 Programming Arrays

The pinned-out designs have demonstrated that SLV-CMOS floating gates can be used to successfully parameterise various RBF subcircuits. However, as discussed previously, a practical implementation of a VLSI RBF will require addressing of these floating gates on-chip within RBF subcircuit arrays. Since programming involves going beyond the normal remit of a SLV-CMOS process (high voltages, tunnelling, hot electrons) then, even with the experience of *TARDIS* as a basis, it could not be taken for granted that any one approach would be free from subtle modes of failure. As a precaution, then, it was decided to investigate *three* entirely separate designs of programming arrays in the hope that this would increase the prospects of obtaining successful results. The methods have been identified as:

- High Voltage Switches
- Dual Phase Programming Scheme
- CHE Flash

These will be described in turn in the course of the next three sections.

6.8 High Voltage Switches

This section describes how addressing the floating gates can be achieved using high voltage switches and how these may be implemented in SLV-CMOS.

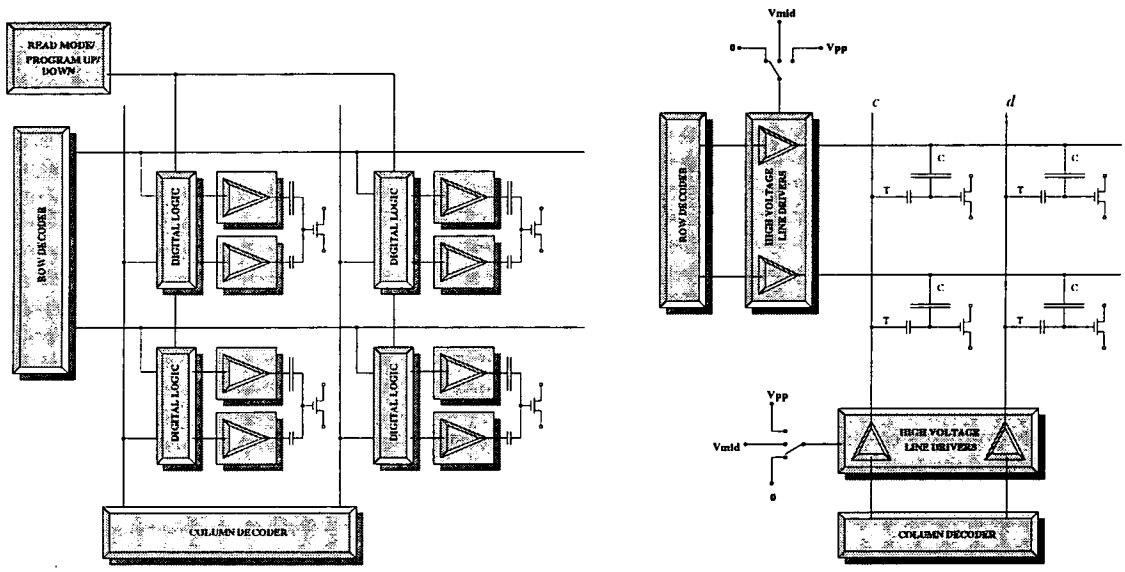
An obvious approach to building an addressable array of floating gates is to provide local high voltage[7] drivers at each cell which level shift from standard logic levels up to V_{PP}. This is illustrated in figure 6.18(a). To remove electrons, V_{PP} is switched onto the terminal of the tunnelling capacitor, with 0V switched onto the terminal of the control capacitor, and vice versa for electron injection (thus capacitively pulling the floating gate high).

A further reduction in cell size is possible by eschewing such binary behaviour in favour of a tertiary level shifter arranged as shown in figure 6.18(b). Here the output of the level shifter may be $0, V_{PP}$ or $V_{mid} \sim \frac{1}{2}V_{PP}$.

For example, suppose that it is desired to remove electrons from the bottom right floating gate in this diagram. 0V would be applied to column c since this is deselected, and V_{PP} would be applied to column d since this is connected to the tunnelling

[7]Modern power semiconductor devices can handle blocking voltages of several kilovolts and source several kiloamps of current. However this is way beyond the tens of volts (and tiny currents) required to program floating gates; this is what is meant by high (ie. significantly above standard logic voltages) in the context of this work.

capacitor terminal **T**. 0V would be applied to row b to ensure the maximum potential difference between the selected device **T** and **C** terminals; the bottom left cell is deselected since it has 0V at both **T** and **C** terminals. V_{mid} would be applied to row a. This would mean a potential of V_{mid} between terminals **C** and **T** of the top left cell and $-V_{mid}$ between terminals **C** and **T** of the top right cell. Due to the exponential Fowler-Nordheim characteristics, the tunnelling current thus incurred would be of several orders of magnitude lower than in the selected cell resulting in negligible disturbance of the deselected cells. It is easy to see how this approach could be expanded to an array of arbitrary size, requiring only $2\sqrt{n}$ high voltage level shifters compared to $2n$ for the localised arrangement of figure 6.18(a), where n is the number of floating gates addressed (this arrangement was used by Lazzaro *et al* [98] for programming SLV-CMOS floating gates for parameter storage in a silicon auditory model).



(a) (b)

Figure 6.18: (a) Localised binary high voltage Drivers, (b) Banked tertiary high voltage drivers. High voltage level shifters are represented as triangles.

However, a problem arises in the implementation of the high voltage level shifters. The Mietec process was designed to cope with analogue circuits running up to a maximum supply voltage of 12V. However, the results from *TARDIS* (and the directly pinned out *NEMO* RBF subcircuits) suggest that voltages of up to 25V are required for use of interpoly tunnelling capacitors. At such voltages various CMOS breakdown events can occur (see appendix D).

Fortunately, there has recently been increasing interest in *smart power circuits*

which integrate high-voltage output stages with low-voltage analogue or digital signal processing. Applications include LCD drivers, thermal printer heads, incandescent lamps, telecommunications, automotive and biomedical electronics [9]. The high voltage stages usually require modified CMOS processes (typically requiring 2 supplementary masks and implantation stages [109, 123]) which result in less readily available and more expensive processes. Fortunately, with motivations analogous to those behind SLV-CMOS floating gates, some researchers have been investigating SLV-CMOS high voltage stages.

Petersen and Barlow [137] describe a method of implementing high voltage switches based on a pull-up circuit and pull-down device. Their design was implemented in a p-well process but it is straightforward to modify it to a n-well process (which is described here).

Pull-Up Circuit

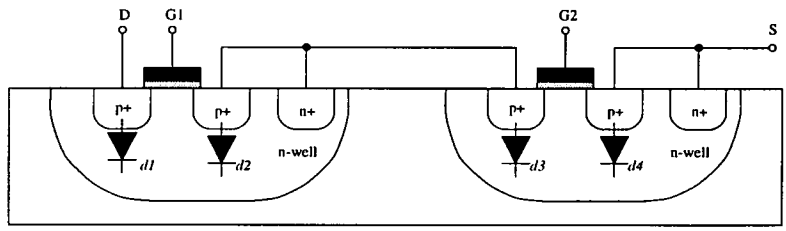


Figure 6.19: Schematic cross section of high voltage pull-up cascode.

The cross-section of a pull-up cascode is shown in figure 6.19. It is required to keep the reverse bias of the diodes $d1 - d4$ below the reverse breakdown voltage. This is achieved by having a number of p-channel transistors (two in figure 6.19) each in *its own n-well.* Only a partial V_{BB} drop is permitted across each transistor, such that the drain-well potential does not exceed the reverse breakdown voltage. Each well therefore is ohmically connected to the drain of the next transistor up in the cascode (with the top well connected to V_{BB}) until sufficient transistors exist to accommodate the entire V_{BB} range. This method is fundamentally bounded by the well-substrate diode reverse breakdown voltage ($>$ 100V in Mietec 2.4μm).

Figure 6.20 shows how this biasing is achieved in practice (a pull-down circuit attached to V_{out} is assumed for the moment, and, for convenience, V_{T_p} is assumed to be -1V), permitting less than 12V reverse bias at any p+/n- diode; for this four transistors/wells are required for the pull-up circuit. The upper circuit shows the switch in the `high` position, with the control line at 0V. Thus $V_{BB} = 40$V can ripple down the cascode through all the transistors, since each bias voltage is below V_{BB}, thus

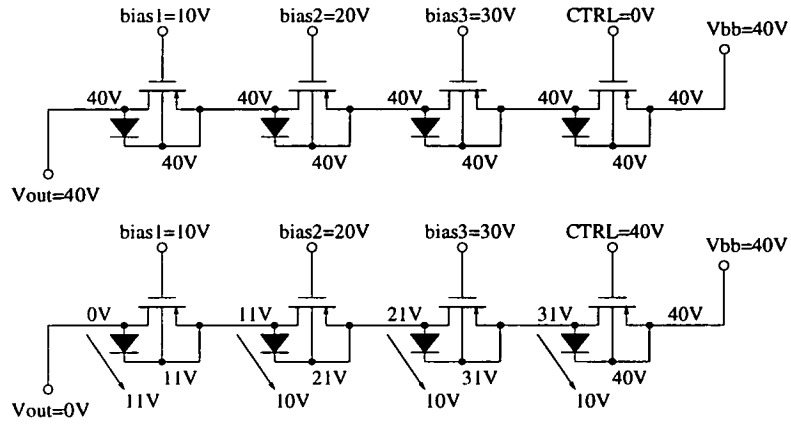


Figure 6.20: Biasing of high voltage pull-up cascode

establishing $V_{out} = 40$V. No p+/n- diodes are reverse biased as all wells and drains are at V_{BB}. The lower circuit in figure 6.20 shows what happens if the control line then changes to V_{BB} (and the pull-down circuit turns on). The transistor drains all discharge to about 1V above their respective bias voltages resulting in a reverse bias of 10V or 11V on all p+/n- diodes while V_{out} goes to 0V. This is the `low` position of the high voltage switch.

Notice that there is potential for gate-drain oxide breakdown in this arrangement if V_{BB} is driven too high.

Pull-Down Devices

Since there are no p-wells to cascode in an n-well process, an alternative approach is needed. For this, a *device* approach, rather than a circuit approach is required. The principal limiting factor is the low reverse bias breakdown voltage of the drain which is due to two factors [4]:

1. Junction Curvature - high curvature provokes high electric fields.

2. Junction doping gradient and gate oxide thickness - lower doping increases the depletion region width [155], thus reducing the electric field at the pn-junction[8], and also the electric field between the gate electrode-drain overlap. Additionally, curvature effects are much less pronounced for graded junctions [157].

[8]This is similar in principle to 'graded transistors' which are formed by the use of spacers either end of the gate during diffusion and allowing lateral diffusion to create a lightly doped gradient between source/drain and channel. Graded n-channel devices *are* available in the given process but only increase the reverse breakdown voltage over conventional n-channel transistors from 18V to 21V as shown in figure 5.1.

A technique, sometimes known as 'soft-drain' or 'lightly doped drain (LDDMOS)', was developed based on the design of early power MOS devices in which the well implant was used to provide a 'buffer region' between the n+ drain diffusion and the channel. The deep, lightly doped well has the low curvature and improved doping profile required to reduce electric field around the drain. Figure 6.21 shows the typical cross section of such a device.

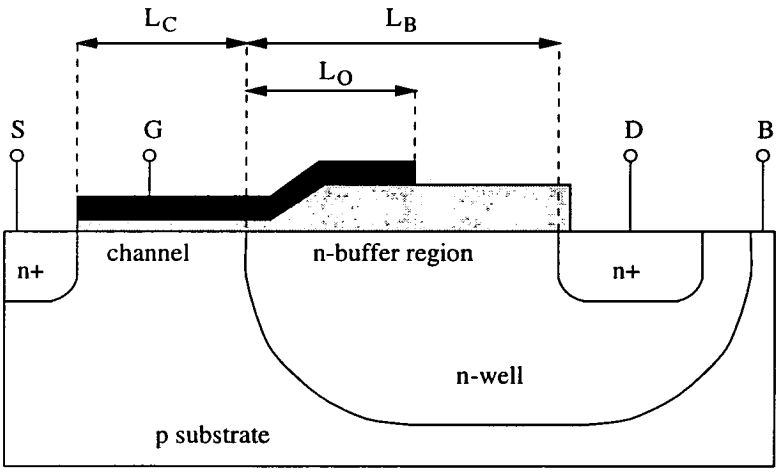


Figure 6.21: Schematic cross section of n-channel LDDMOS transistor.

Many researchers have tried to extend the LDDMOS concept to both pull-down and pull-up devices since the cascode has a number of problems:

1. A large area overhead of multiple wells in a cascode.
2. Limited by the gate-drain breakdown voltage.
3. Poor for large currents because of the number of channels.

This requires a lightly doped region in the well of the opposite polarity. Such a layer is available in certain processes, typically BiCMOS, although it is certainly not standard. A field implant (channel stop) does however exist in some processes and has been used as shown in figure 6.22(a). An alternative is the creation of a 'tub' by designing two wells closely spaced such that they underdiffuse into each other enclosing an isolated area of substrate [4]. However, knowledge of lateral underdiffusion characteristics are required to use such a tub, which is also heavily space limited. The breakdown of this device was inconveniently low ($\sim$ 21V) due to its short length and laterally non-homogeneous doping profile. Another fundamental limitation is due to vertical punchthrough of the tub from drain to substrate where the wells meet (and is hence not very deep). A summary of SLV-CMOS high voltage work is given in table 6.1.

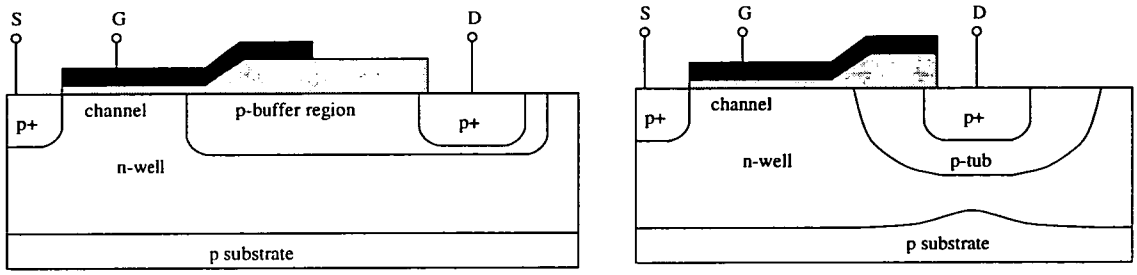


Figure 6.22: Schematic cross section of p-channel LDDMOS transistor using (a) field implant buffer region, and (b) p-tub.

Authors	Process	Pull-Up Device	Pull-Down Device	V_{BD}†
Petersen & Barlow [137] (1982)	p-well	LDD-PMOS using p-well (native device)	1. Cascode Configuration, 2. Lateral Bipolar	$\geq$ 60V
Parpia, Salama & Hadaway [135] (1987)	p-well	LDD-PMOS using p-well (native device)	LDD-NMOS using n field implant	50V
Apel, Habekotté and Höfflinger [4] (1989)	p-well	LDD-PMOS using p-well (native device)	1. LDD-NMOS using n-tub, 2. Lateral bipolar	$\leq$ 21.5V
Haas, Au, Martin, Portlock & Sakurai [70] (1989)	n-well	LDD-PMOS using p field implant	LDD-NMOS using n-well (native device)	$\leq$ 40V
Mann [106]	n-well with bipolar implant	LDD-PMOS using p-base implant in cascode configuration	LDD-NMOS using n-well	$\geq$ 30V
Declercq, Clement, Shubert, Harb & Dutoit [47] (1993)	n-well	LDD-PMOS using p field implant	LDD-NMOS using n-well (native device)	$\leq$ 75V
Behrens, Finco & Simas [12] (1993)	n-well	none	LDD-NMOS using n-well (native device)	$\leq$ 160V‡

Table 6.1: Selected publications on design of SLV-CMOS high voltage devices. †Quoted breakdown voltage of the weakest overall device in the circuit (‡This remarkably high voltage is achieved since there is no non-native devices – ie. pull-up in an n-well process – which tends to be the limiting factor. However their absence seriously limits the applications). Due to their low gain and substantial power dissipation, the use of lateral bipolar devices [167] is more appropriate to applications which demand high current (LDDMOS techniques would be too area hungry for such applications).

A team from EPFL Electronics Laboratories in Lausanne, Switzerland have been heavily involved with SLV-CMOS devices and have designed a number of analogue and digital cells [47, 49, 48, 149, 10]. The process used for these circuits has a thin gate oxide which can only support a 5V V_{GS} swing. This adds constraints and complexities beyond that experienced by most other researchers in this field.

Negative High Voltage Switches

The previous discussion describes work based around the construction of *positive* high voltage switches with reference to the substrate bias. It is not possible to switch negative-referenced voltages on a standard CMOS chip as this will inevitably result in the forward-biasing of parasitic diodes. The solution here was to develop circuit approaches which circumvented the need to switch negative voltages on-chip. However, if switching of negative voltages was required, one approach would be to migrate to a *twin-tub process* [170] which provides both n and p-type wells (tubs). Normally this process is used to balance n and p transistor performance by separately optimising the threshold voltage, gain and body effect. However, in this process negative voltages with respect to the substrate can be accommodated by appropriate biasing of the p-tub.

6.8.1 Mietec High Voltage Devices

No high voltage work has previously been reported with the given Alcatel-Mietec 2.4μm process. From the evidence of interpoly tunnelling characteristics and oxide breakdown data in $TARDIS$, it is required to support voltages of up to only 30V. This is sufficiently lower than the gate oxide breakdown voltage ($\sim$ 40V) that techniques of the EPFL group are not required although the oxide breakdown voltages were obtained for different biasing conditions. However, the p+ field implant mask is generated automatically (by oversizing of the n-well mask) and so it is not possible to generate the p-layer within a well as would be required for a p-channel LDDMOS device[9]. Since V_{BB} requirements are moderate and no large currents are needed, a LDDMOS pull-down and cascode pull-up circuit were chosen as the most promising combination.

[9]Strictly, it *is* possible to do this since it does not involve any further masking steps, rather an extra 'NONGEN' layer must applied which disables automatic generation (using logical formulae) and the extra layers must be designed by hand. However access to the 'NONGEN' layer is not available through Europractice and must involve direct intervention at the mask shop. This detracts from the desire for a layout conveniently accessible to the VLSI designer. In any case, it would introduce a further non-standard device (DRC-violation), introduce more scope for error and since no information (depth, dopant density) about the channel stop field implant is freely available, is likely to further compromise the chances of successful fabrication. For this reason, the cascode circuit approach is still preferred (EPFL devices by comparison *do* depend on mask shop intervention techniques).

L_C	Channel Length	20μm
L_B	Buffer Region Length	12μm
L_O	Gate Overlap Length	4μm
	Channel Width	20μm

Table 6.2: Layout geometry of LDDMOS transistor

The geometry of this device is given in table 6.2. The sizes were selected by comparison to those in the literature and by study of the provided layout rules. Specifically the well had to be sufficiently far from the source to preclude underdiffusion or lateral punchthrough, and the gate had to extend sufficiently far into the well to ensure channel continuity – since self-alignment is obviously lost – but not too far as this would promote gate-drain oxide breakdown [47]. Notice that since the LDDMOS device was intended as a pull-down element in a digital switch, the actual sizing was not considered to be critical provided that some leeway was allowed for slippage of the masks. Since the source and channel regions are the same as for a conventional n-channel MOS transistor, V_{T_n} is expected to be approximately the same although this is not a requirement.

These LDDMOS devices generate design rule check (DRC) violations:

- N+ diffusion to N-Well spacing $< 10\mu$m (Mietec layout rule 2G)[10].
- N+ diffusion inside N-Well not fully overlapped by N-well (Mietec layout rule 2H).

These rules provide some protection against accidental layout errors but do not impact upon mask generation; therefore fabrication can proceed directly from layout without intervention of the mask shop (there would only be a problem if DRC violations were banned, but Europractice allows these although liability falls on the designer).

6.8.2 Circuit Designs for High Voltage Switches

Having now discussed pull-up and pull-down high voltage circuits, the requirements of the floating gates and the specific constraints of the Mietec 2.4μm process, a high voltage *switch* can now be presented which makes use of these designs.

[10]This rule is based on the distance between active area in the substrate (the channel) to the well edge; in reality the N+ diffusion is more than 10μm from the well

Cross-Coupled Digital Switch

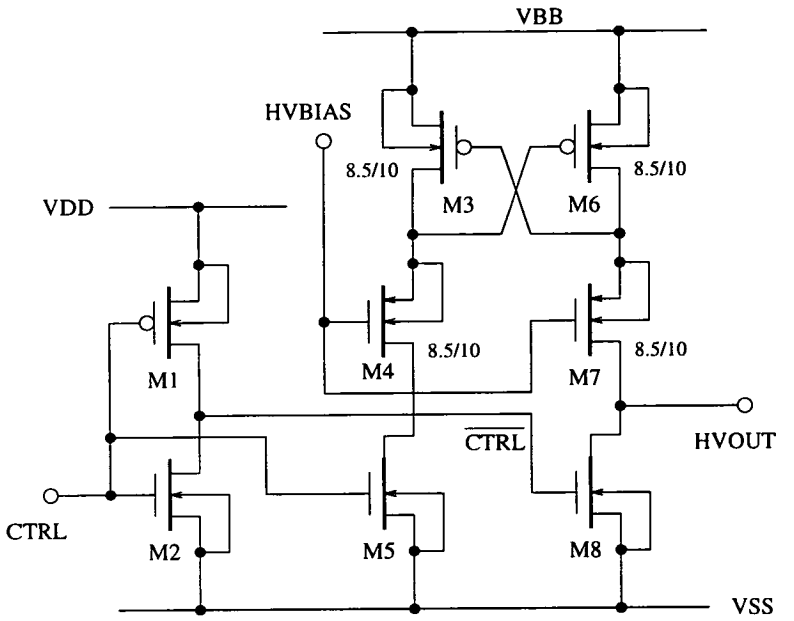


Figure 6.23: High voltage digital switch schematic. LDDNMOS transistors are represented with drawn extended drains.

Such a switch circuit is shown in figure 6.23. The circuit acts digitally under the control of the standard 5V logic control $CTRL$ and its 5V complement $\overline{CTRL}$ generated with a standard digital inverter. Either $M5$ pulls the left side of the switch low, and because of the cross-coupled arrangement, pulls the right side high or $M8$ pulls the right side low which pulls the left side high. In this way $HVOUT$ can be switched between V_{BB} and V_{SS}. $M4$ and $M7$ are biased with $HVBIAS$ which is such that the high voltage is distributed across the two transistors ($HVBIAS \sim \frac{1}{2}V_{BB}$ in practice). This circuit is very similar to the design by Mann [106] (which was subsequently used by Lazzaro *et al* [98]). However, Mann used only a single well for all the p-channel transistors since avalanching of drain diodes was not of concern in his process (primary breakdown was in the channel due to high V_{DS}).

Differential Stage Analogue Switch

An alternative analogue approach was suggested by Lande [11] but not previously implemented in silicon. It is based on a differential stage as shown in figure 6.24(a). As before $CTRL$ is the digital switch control. $BIASV$ is about mid way between

[11] Private correspondence with Tor Sverre Lande, February 1996.

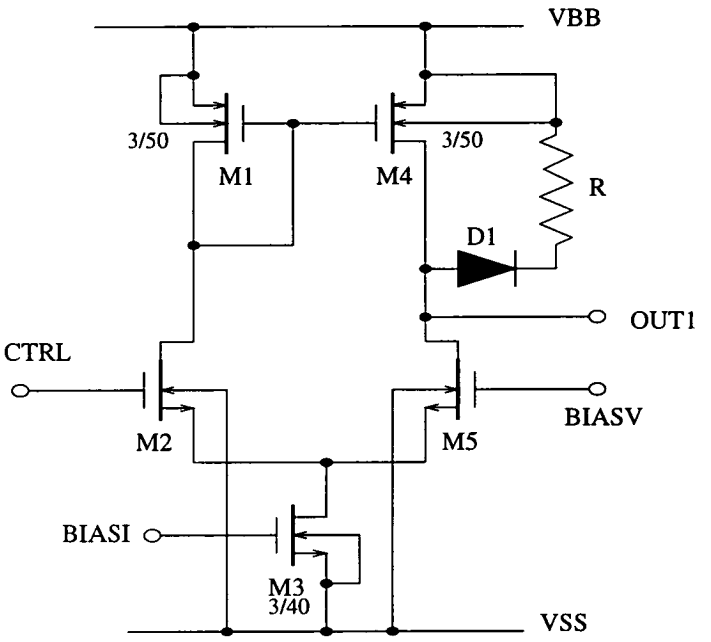


Figure 6.24: High voltage differential stage switch schematic.

V_{DD} and V_{SS}. Thus the switch can be thought of as a normal differential stage where $OUT1$ can switch between V_{BB} and V_{SS}. The matched pair are LDDNMOS and so support the high voltages, $M1$ is diode connected, so its drain is only about V_{T_p} below the well potential; thus the only problem is the drain diode of $M4$: when $CTRL$ is low, $M5$ tries to pull $OUT1$ low and this reverse biases this diode such that it breaks down. However, if the tail current set by $BIASI$ on $M3$ is sufficiently low[12] that the breakdown current in the diode is insufficient to burn out the junction; it is the thermal acceleration in reverse-biased junctions which is destructive, not simply that they are passing some current. Thus, in comparison to the digital switch, breakdown is *controlled* rather than avoided altogether. Notice now, that in the switch's `off` state the conducting reverse-biased diode forms part of a potential divider between V_{BB} and V_{SS}. Therefore, while $OUT1$ may switch fully to V_{BB} it cannot switch fully to V_{SS}. By laying the well out such that this diode is physically distant from the well-tap, the well resistance can be used to further limit the reverse-bias current and so decrease the low-state voltage of $OUT1$ (This is shown as well resistance, R and diffusion diode $D1$ in figure 6.24). There is a compromise to be made between well resistance and

[12]This bias was set up by a 1μA current source and a divide-by-100 current mirror such that about 10nA was expected as the tail current. This is within the weak-inversion (sub-threshold) regime of MOS operation where current exhibits an *exponential* V_{GS}-dependency. Thus it is difficult to mirror currents accurately. However, this is not important, all that is necessary is that the tail current be low – an order of magnitude either way about 10nA is not of significance.

area and latch-up immunity. Notice that this switch can only source small currents and changes state comparatively slowly. These characteristics are respectively sufficient and desirable for programming floating gates.

Layout of High Voltage Switches

Layout for the high voltage switches described is shown in figure 6.25. The actual designs are quite large and look somewhat unusual due to a number of layout precautions which were taken:

- Clearance - Substantial clearance was allowed for around the wells (drains and cascode stages) to allow for substantial depletion region growth on the appliance of high voltages.

- Shielding - Where possible, close to active devices, lines carrying high voltages were run in metal2 and shielded from the substrate by metal1. Where metal1 was used, many substrate taps were placed to prevent surface inversion.

- Guard Rings - High voltage switches were heavily ringed by *double* guard rings: a n-diffusion ring in the wells and a p-diffusion ring about the entire structure. Due to the high voltage there was a higher likelihood of high energy carrier injection into the substrate. The guard rings were intended to absorb this current and so prevent latch-up. (The Mietec 2.4μm process uses an epitaxial start material which also helps provide good latch-up immunity [170]).

6.8.3 Experimental Results

LDDNMOS Transistors

The I_{DS}-V_{DS} characteristics of a LDDNMOS transistor from seven different sample chips are plotted together in figure 6.26(a) with V_{DS} up to 30V, in excess of the floating gate programming requirements. Despite the DRC violations, no faulty LDDNMOS transistors were found in over forty tested (either individually or as part of a switch). Moreover, as the figure shows, the characteristics are very consistent between chips providing good evidence of reliable device fabrication. The threshold, $V_{T_{lddnmos}}$ was found to be about 0.9V, the same as V_{T_n}. By way of comparison, the `Hspice` model of a conventional Mietec n-channel MOSFET is plotted in figure 6.26(b). The higher transconductance of the LDDNMOS transistor may be attributable to some channel shortening due to lateral underdiffusion of the well.

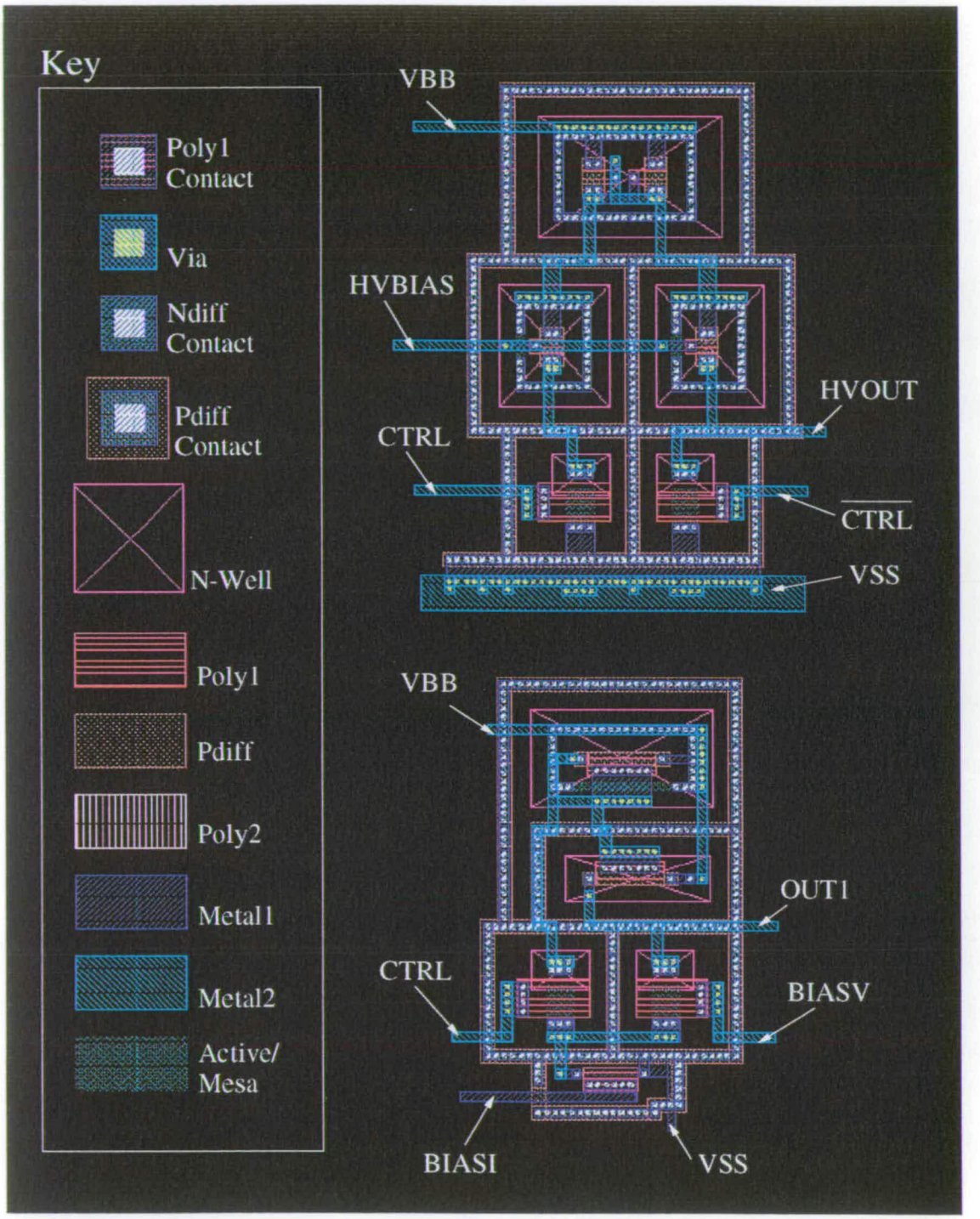


Figure 6.25: Layout of high voltage switches: (a) Cross-coupled digital switch, (b) Differential switch. Approximate pre-shrink sizes are (a) 380μm $\times$ 245μm, and (b) 330μm $\times$ 200μm

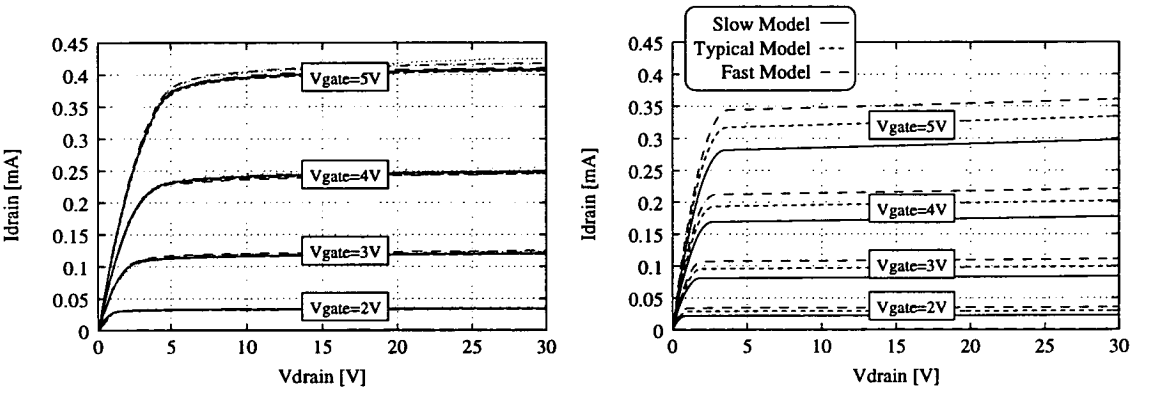


Figure 6.26: (a) LDDMOS I_{drain},V_{drain} characteristics for transistors on seven different chips, (b) `Hspice` simulation of 20μm/20μm NMOS transistor

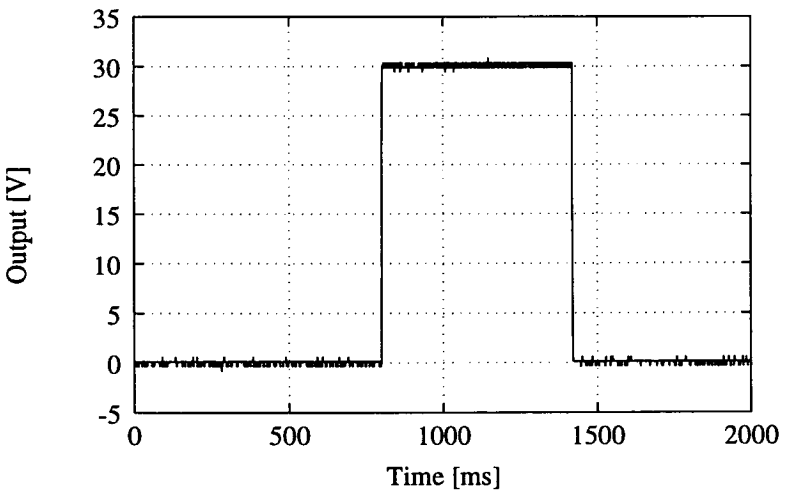


Figure 6.27: CRO trace of $HVOUT$ with $V_{BB} = 30$V.

Cross-Coupled Digital Switch

Figure 6.27 shows a CRO trace of $HVOUT$ with V_{BB} = 30V illustrating that the cross-coupled digital switch can generate high voltage pulses (under the control of $CTRL$) to the extreme of the specification. One switch was used to generate 30V pulses with a 50% duty cycle and period of 600ms; it was allowed to run for a week which represents over a million pulses and no degradation of the device was observed.

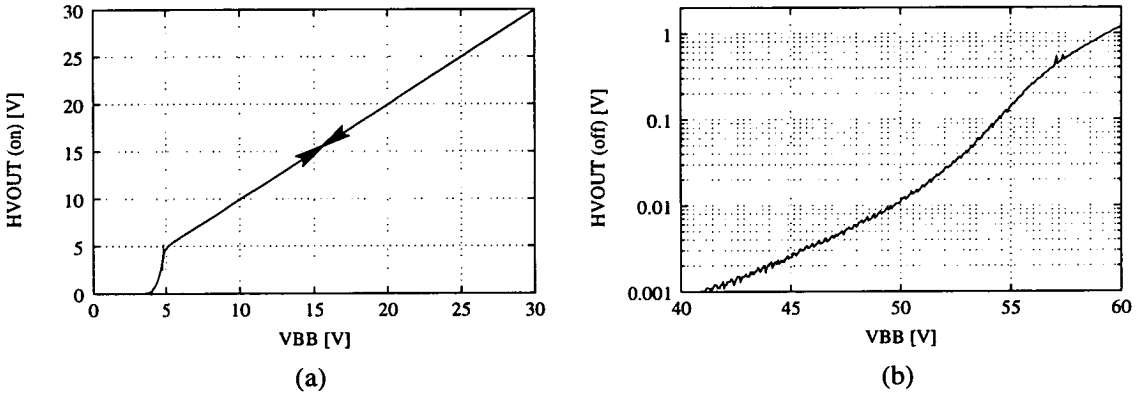


Figure 6.28: (a) $HVOUT$-V_{BB} for V_{BB} symmetrical ramp and switch in on-state, (b) $HVOUT$-V_{BB} for high V_{BB} and switch in off-state.

With $CTRL$ high (switch on), V_{BB} was ramped linearly from 0V to 30V and then back to 0V while $HVOUT$ was monitored. This is illustrated in figure 6.28(a). With $CTRL$ low (switch off) the ramp was repeated and 0V was obtained at $HVOUT$. This shows correct operation of the switch in the range $\sim$ 5V – 30V without significant hysteresis.

Although there is here no practical purpose to operating above about 30V (since this would result in floating gate oxide breakdown) it was interesting to observe the behaviour of the switch at higher V_{BB} values. V_{BB} was increased to 60V, and $HVOUT$ followed it when the switch was in the on state. However, with the switch in the off state $HVOUT$ began to rise above V_{SS} as shown in figure 6.28(b). As V_{BB} rises, the reverse biased diodes in the pull-up chain begin to break down; as seen at about 60V the reverse bias current is sufficient that the off-state output is above 1V. Although the switch continued to operate successfully after this experiment it probably impinges upon reliability to operate at such voltages.

Notice that due to the high V_{BB} voltage, transient power consumption can be high. If required, significant power reduction can be achieved by simple overlap-cancellation logic which avoids simultaneous conduction of pull-up and pull-down stages during

switching[13].

Differential Stage Switch

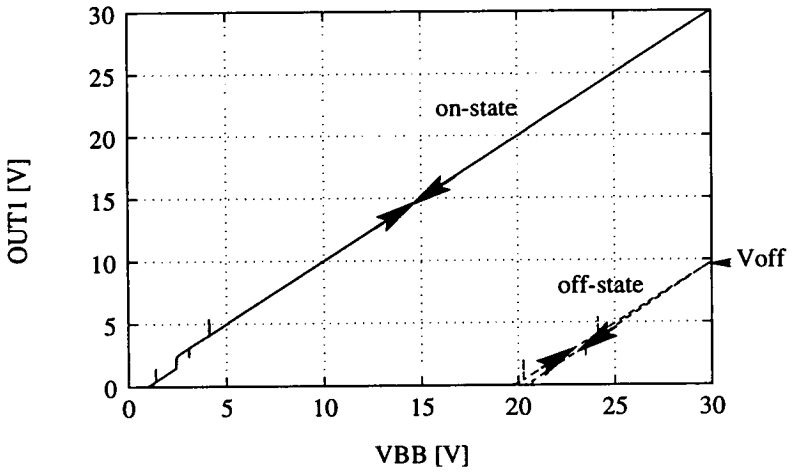


Figure 6.29: $OUT1$-V_{BB} for V_{BB} symmetric ramp in both on and off states.

The differential stage switch was also found to work and behaved as shown in figure 6.29 when a ramped V_{BB} was applied. Notice that with V_{BB} above about 20V, $OUT1$ cannot be pulled to V_{SS} in the off-state. This is as expected since this is about the reverse bias breakdown voltage of the p-channel drain diode. Whilst this is clearly not as ideal as the cross-coupled digital switch, it should be sufficient for programming floating gates since it is below the tunnelling threshold, V_{fe}. A peculiar observation was made during the ramping experiment in the off-state: that the off-state $OUT1$-VBB response line at 30V (V_{off} in figure 6.29) fell continuously during the experiment. Further study showed that it was not the number of pulses but the amount of time held in the off-state with $V_{BB} = 30$V which resulted in this behaviour.

Figure 6.30(a) shows how V_{off} decays while left in a constant off-state for about an hour. It is not possible to wholly attribute this behaviour to junction heating in the breakdown diode since figures 6.30(b) and (c) show that, while some recovery occurs, leaving the device unpowered either overnight or for 2 days does not reset to the behaviour from fresh. Clearly some permanent or semi-permanent change has occurred at the breakdown junction although with a 10nA current limit it is difficult to attribute this to a specific cause. Fortunately the decay does not seem to harm the switch (it still runs up to V_{BB}) and indeed actually improves it since the switch grows closer to V_{SS} with repeated use.

[13]This is a common feature of class-D power amplifiers.

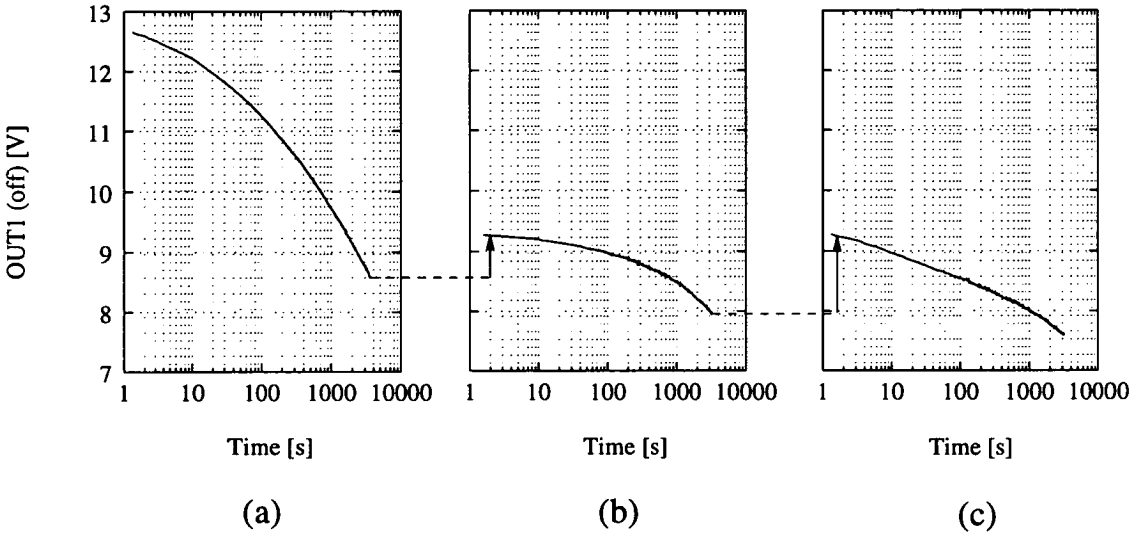


Figure 6.30: (a) Initial decay of V_{off} in fresh device, (b) Repeat of experiment (a) after leaving overnight, (c) Repeat of experiment (a) after leaving for two days.

Yield

Seven cross-coupled digital switches and seven differential stage switches were tested and all were found to function as expected. However since the sample population was limited[14] it is impossible to draw any real conclusions on yield in respect of the DRC violations. However the evidence of these results is promising. Notice too that the high voltages tests were found to have no noticeable effect on the other test arrays on *NEMO*.

6.9 Dual Phase Programming Arrays

At design-time there was no way of knowing if the high voltage switches would operate successfully since they depend critically on non-standard device elements which in turn are based on non-documented (and possibly poorly controlled) features of fabrication. For this reason it was desirable to look at alternative approaches to addressing a floating gate array in case of failure.

Additionally, even with the evidence that such switches *can* be designed in the Mietec 2.4μm process, such designs are susceptible to process flow changes and technology migrations, and are also area hungry due to their very cautious design (although

[14]There was only one of each type of switch on each chip. 10 packaged chips were supplied from fabrication: 7 were tested, 2 held in reserve and one was damaged prior to test.

reducing size is now worth considering since their operation in principle has been proven). Therefore, the investigation of alternative approaches is still worthwhile.

Thomsen and Brooke [160] describe a method which does not require high voltage switches and allows loop-controlled programming; unfortunately it requires a high-gain feedback stage associated with each memory cell which can be costly in area.

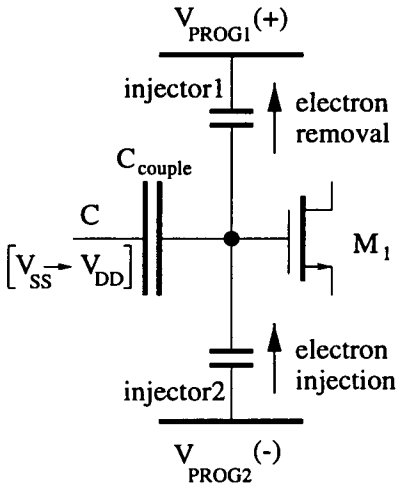


Figure 6.31: Schematic of Thomsen and Brooke's dual-injector floating gate memory

Figure 6.31 shows the dual-injector[15] structure described in [160]. Terminal **C** is normally biased at the midpoint of V_{SS} and V_{DD}. Two static global signals, V_{PROG1} and V_{PROG2}, are set-up such that if **C** was driven to V_{SS} pulling down the floating node voltage through capacitive coupling, there would be sufficient electric field across injector1 that electrons would be removed from the floating node. Similarly, by driving **C** to V_{DD}, electrons can be injected through injector2. Thus programming can be controlled *without* having to switch high voltages. Unfortunately, since only half of the potential rail-to-rail voltage swing is utilised to differentiate selected and deselected devices, coping with the wide distribution of $|V_{fe}|$ can be a problem.

Firstly analysing the situation when electrons are being *removed* from a selected device (ie. through injector 1), then using the definition in equation 5.1, $V_{tunnel} = V_{PP} - V_{fg}$, the lowest possible tunnelling potential in a *selected* device, since V_{PP} is fixed at V_{PROG1}, is when ϕ_{fg} is maximal:

$$V_{fg}^{selected}(max.) = m_c V_{SS} + m_{t_1} V_{PROG1} + m_{t_2} V_{PROG2} + \phi_{fg}^{max} \tag{6.9}$$

Conversely the highest possible tunnelling potential in a *deselected* device is when ϕ_{fg}

[15] In keeping with Thomsen & Brooke's terminology *injector* will be used instead of *tunnelling capacitor* although the terms are synonymous.

is minimal:

$$V_{fg}^{deselected}(min.) = m_c\left(\frac{V_{DD}-V_{SS}}{2}+V_{SS}\right)+m_{t_1}V_{PROG1}+m_{t_2}V_{PROG2}+\phi_{fg}^{min} \tag{6.10}$$

Therefore, for the dual-injector structure to function correctly, it is required for programming of selected devices that

$$V_{PROG1}-m_cV_{SS}-m_{t_1}V_{PROG1}-m_{t_2}V_{PROG2}-\phi_{fg}^{max} \geq V_{fe}^{+}(max) \tag{6.11}$$

And for no re-programming of deselected devices that

$$V_{PROG1}-m_c\left(\frac{V_{DD}-V_{SS}}{2}+V_{SS}\right)-m_{t_1}V_{PROG1}-m_{t_2}V_{PROG2} \\ -\phi_{fg}^{min} \leq V_{fe}^{+}(min) \tag{6.12}$$

Defining $V_{fe}(range) = V_{fe}^{+} - V_{fe}^{-}$, then subtraction of 6.12 from 6.11 gives

$$\begin{aligned} -m_cV_{SS}-\phi_{fg}^{max}+m_c\frac{V_{DD}}{2}-m_c\frac{V_{SS}}{2}-m_c+\phi_{fg}^{min} &\geq V_{fe}^{+}(range) \\ m_c\left(\frac{V_{DD}-V_{SS}}{2}\right)-\phi_{fg}^{max}+\phi_{fg}^{min} &\geq V_{fe}^{+}(range) \\ m_c\left(\frac{V_{DD}-V_{SS}}{2}\right)-\phi_{fg}(range) &\geq V_{fe}^{+}(range) \end{aligned} \tag{6.13}$$

Similarly, for electron injection (ie. through injector 2), it is required that

$$m_c\left(\frac{V_{DD}-V_{SS}}{2}\right)-\phi_{fg}(range) \geq V_{fe}^{-}(range) \tag{6.14}$$

So, if we define $V_{fe}(range) = max.\left\{V_{fe}^{+}(range), V_{fe}^{-}(range)\right\}$, then for correct operation

$$\left| m_c\left(\frac{V_{DD}-V_{SS}}{2}\right)-\phi_{fg}(range)\right| + V_{margin} > V_{fe}(range) \tag{6.15}$$

Where ϕ_{fg}^{range} is the total range of stored charge values which a programmed floating gate may take. V_{margin} is included to allow biasing selected devices *above* V_{fe} for faster programming and also to prevent parasitic reprogramming due to sub field emission threshold currents during the entire down loading period; its value may be defined by the application.

It is difficult to match the constraint of equation 6.15 with the limited V_{DD} of SLV-

CMOS. *TARDIS* data shown in figure 5.8(c)-(d), and also more recent *NEMO* RBF subcircuit programming data in figure 6.14 indicate that even eight volts differentiation would lead to some disturbance of deselected cells (ie. figures 6.14(a)-(d)show fast programming with $V_{PP} = 20$V and slow programming with $V_{PP} = 12$V[16]. Even Thomsen & Brooke might expect similar disturbance (eg. for differentiation of 6V in figure 4 of cite [161]). Whilst it has been useful in preceding work to talk of a field emission threshold, V_{fe}, to allow comparison between different tunnelling capacitor designs, this distinction is based on an arbitrary value of $\Delta\phi_{fg}$ for an arbitrary t_{PP}. However dissecting tunnelling potentials into binary 'tunnelling exists' and 'no tunnelling' categories is not realistic: cumulative effects of small parasitic tunnelling currents over long programming periods (ie. the sum of t_{PP} over all programmable elements) can lead to severe disturbance. FNT in SLV-CMOS floating gates *cannot* be treated as an ideal diode.

Added to this is the inevitable impact of interdevice variability – even Thomsen & Brooke admit that "Strong non-uniformity of tunnelling thresholds leads to a not fully programmable array of devices." [160] – and aging: deselected devices with small parasitic tunnelling currents will not age as fast selected devices, hence the V_{PP} for the selected device must be increased to maintain programmability whilst additionally incrementing the parasitic tunnelling current in the deselected device. *TARDIS* results, summarised in table 5.2, show both the V_{fe} population and aging characteristics to be widely distributed.

If this scheme is to be made workable then the *maximum possible* differentiation in tunnelling potentials between selected and deselected floating gates afforded by the process voltage limitations is necessary: the partial control voltage swing of [160] is insufficient. Instead, full *rail-to-rail* switching of control is required. This demands a clocked, high-voltage programming waveform which is more conveniently generated off-chip and distributed globally (fig. 6.32).

In this scheme each floating gate has a low-voltage digital control signal associated with it. As V_{PP} swings between its positive and negative peaks, the digital control swings between V_{DD} and V_{SS}. Deselected devices swing in phase with V_{PP}, so to *minimise* the potential across the tunnelling injector while selected devices swing out of phase with V_{PP} and *maximise* the potential across the tunnelling capacitors.

Now, the equivalent expressions for those of equations 6.9 and 6.10 for the dual-injector scheme are

$$V_{fg}^{selected}(max.) = m_c V_{SS} + m_t V_{PP}^{+} + \phi_{fg}^{max} \tag{6.16}$$

[16]Simply decreasing V_{PP} clearly increases the tunnelling time for both selected and deselected devices and so does not reduce disturbance. Indeed simulation with the *TARDIS* programming model suggests that it would in fact aggravate the disturbance.

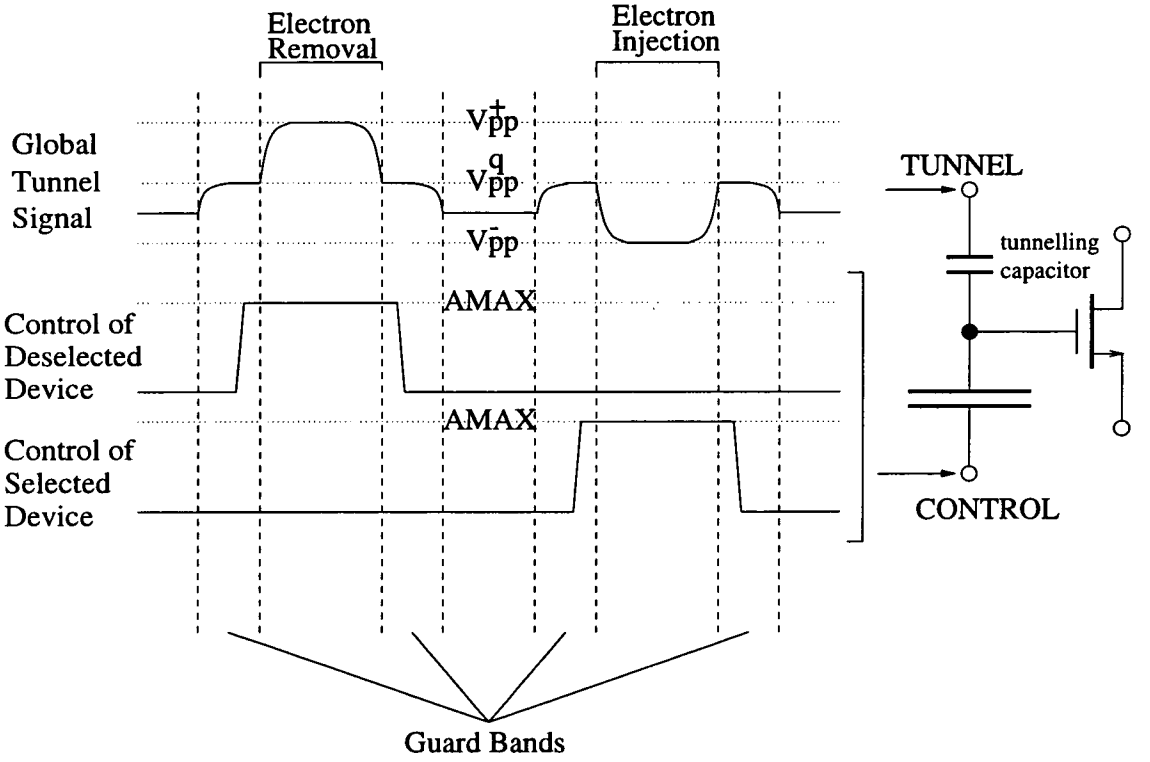


Figure 6.32: Dual-Phase FNT programming scheme

and

$$V_{fg}^{deselected}(min.) = m_c V_{DD} + m_t V_{PP}^{+} + \phi_{fg}^{min} \tag{6.17}$$

which, by a similar analysis to before, leads to the expression for correct operation

$$\left| m_c \left(V_{DD} - V_{SS}\right) - \phi_{fg}(range) \right| + V_{margin} > V_{fe}(range) \tag{6.18}$$

which is an improvement in separability of

$$\frac{m_c}{2}\left(V_{DD} - V_{SS}\right) \tag{6.19}$$

The previous discussion used V_{DD} and V_{SS} to allow easy comparison with the dual injector scheme. However, from now, to avoid confusion with the conventional +0V,+5V supplies which drive the parameterised neural circuits (and for reasons of power conservation should not be driven at up to +12V), and to emphasise the availability for tuning, $V_{DD} - V_{SS}$ is replaced by $AMAX$. $AMAX$ is thus the zero-referenced control voltage used exclusively in the dual-phase floating gate programming elements.

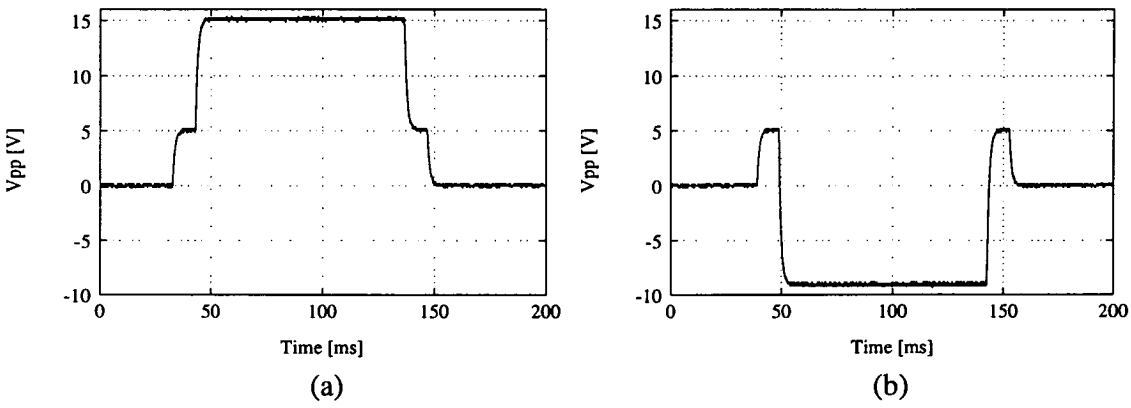


Figure 6.33: Oscilloscope trace of the actual Global Tunnel signal used in the iterative Dual-Phase FNT programming scheme: (a) V_{PP}^{+}, (b) V_{PP}^{-}

Fig. 6.32 shows the details of the signals for a pulse/measure iterative programming procedure. The global tunnelling signal is shown rounded by a single low pass filter (in practice an RC-filter of $f_{-3dB} \sim$ 65Hz was used). (Figure 6.33 shows the actual measured global tunnelling signal as generated on the testboard described in section B.3.2).

The indicated guard-band periods are in place to ensure that no parasitic tunnelling occurs due to the coupling of an $AMAX$ control signal onto the floating gate whilst the global tunnelling signal is still low. In these guard bands, the global tunnelling signal

is raised to V_{PP}^{q} which is still low enough not to induce removal from devices with 0V control lines, but large enough to reduce the tunnelling potential in devices with $AMAX$ control lines (ie. $m_c AMAX - V_{PP}^{q}$). In practice $V_{PP}^{q} = 5\text{V}$ was used. To further simplify the programming scheme the guard bands may be removed. These are only really necessary for *high* $AMAX$ and *very low* V_{fe} devices (typically very unaged devices); in practice extremely few devices were sufficiently prone to tunnelling as to be re-programmed by the coupled-in $AMAX$ signal even with $V_{PP}^{q} = 0\text{V}$.

Benefits of the New Scheme

With the Mietec process' analogue $V_{DD(max.)} = AMAX = 12\text{V}$, this is the maximum used control voltage. Thus, rewriting equation 6.18, gives

$$\left| m_c\,(AMAX) - \phi_{fg}(range) \right| + V_{margin} > V_{fe}(range) \qquad (6.20)$$

This improvement in separability is in excess of that required to compensate for the size of $V_{fe}(range)$, so other gains can be achieved:

- $\phi_{fg}(range)$ can be increased.
- V_{margin} can be increased, reducing parasitic injection/removal during downloading.
- m_c can be reduced, leading to a smaller coupling capacitor.
- Programming can be faster since selected devices can be biased further into the FNT regime.

Other benefits obtained are:

- Only one injector per cell and only one global signal to be routed saves area.
- The onset of tunnelling is controlled by the shaped global signal, not the sudden switch in bias point; use of a sinusoidal waveform can reduce oxide stress and limit aging.
- Bipolar rather than unipolar tunnelling in the injector may also limit aging - see figure 5.40.

The last two aspects increase window-closure endurance which is particularly important here for asperity-dominated tunnelling where trap-up is exacerbated by a thicker oxide and trap influence is also enhanced by field acceleration.

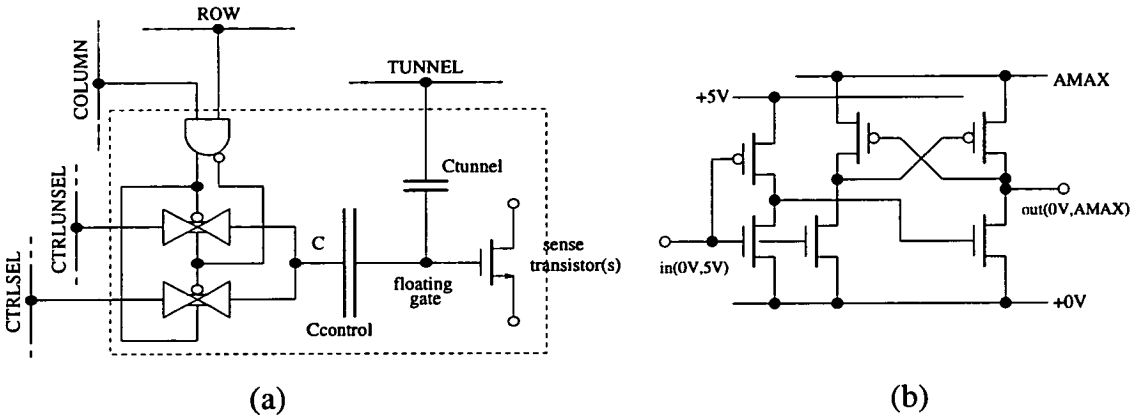


Figure 6.34: Circuit implementation schematic for dual-phase FNT programming scheme

6.9.1 Circuit Implementation

The implementation details of the Dual-Phase scheme are shown in figure 6.34(a) where the dashed box indicates the memory element for a single cell. The digital control signals, CTRLSEL and CTRLUNSEL and the address lines all run at $AMAX$, to allow passing of the correct logic HIGH levels through the two pass transistors. The cell selected for programming is addressed individually. This passes CTRLSEL onto node **C** of the selected cell and CTRLUNSEL onto node **C** of all the other (unselected) cells. CTRLSEL and CTRLUNSEL must operate in accordance with the scheme described in the previous section during programming. During readout, the global tunnelling signal and both CTRLSEL and CTRLUNSEL are LOW, zeroing the large coupling terms and effectively allowing ϕ_{fg} to parameterise the circuit. These signals are shown in table 6.3.

Control Signal	Program		Read
	V_{PP}^{+}	V_{PP}^{-}	
CTRLSEL	0	1	0
CTRLUNSEL	1	0	0

Table 6.3: Control signals for iterative dual phase programming scheme

Control and addressing signals are generated in conventional 5V digital logic and mapped into $AMAX$ logic levels using the logic level shifter circuit shown in figure 6.34(b). Minimum length 'digital' transistors cannot be used for $AMAX > 7$V due to the danger of snapback.

6.9.2 Programming Characteristics

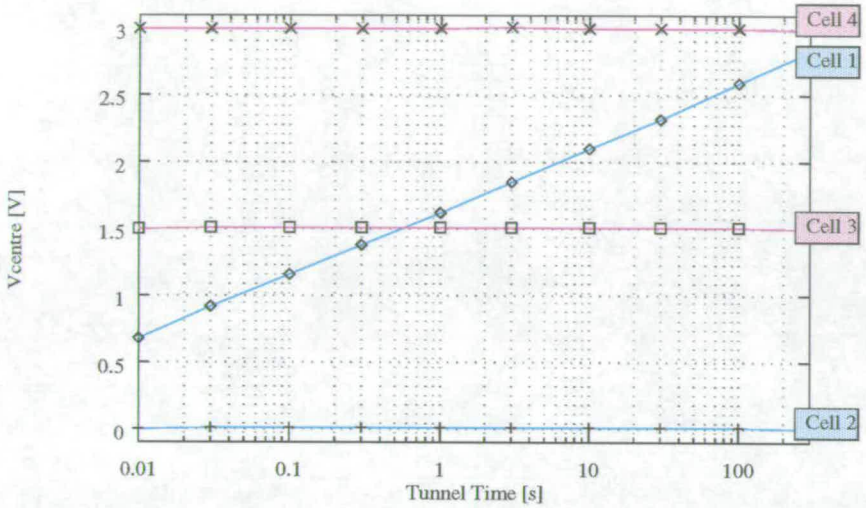


Figure 6.35: Typical programming characteristics: cell centre versus time for $V_{PP} = +16$V. Cell 1 is selected and initialised to 0.0V. Cells 2,3 and 4 are deselected with centres initialised to 0.0V, 1.5V and 3.0V respectively. $AMAX$ is 10V.

Figures 6.35 and 6.36 shows typical programming characteristics for the dual phase scheme, illustrating how a selected cell may be programmed through the range 0V$\leq V_{centre} \leq$ 3V or $-1 \leq \omega + 1$ respectively without disturbing the programming of deselected cells. As always, the speed of programming is determined by the magnitude of V_{PP}.

To test the resolution of this scheme for the Euclidean distance calculator array, cell 1 was programmed using V_{PP} values of $6V \rightarrow 20V$, and $(-6V + AMAX) \rightarrow (-20V + AMAX)$, and the maximum disturbances in the preset values of cells 2, 3 and 4 were measured. Programming ceased when $V_{centre}^{cell(1)}$ traversed the entire range 0V – 3V or when 300s of tunnelling time had elapsed, whichever was the sooner.

$V_{centre}^{cell(1)}$ was initialised to 0.0V for V_{PP}^{+} and 3.0V for V_{PP}^{-}. The other cells were initialised irrespective of V_{PP}: $V_{centre}^{cell(2)} = 0.0$V, $V_{centre}^{cell(3)} = 1.5$V and $V_{centre}^{cell(4)} = 3.0$V.

The maximum disturb errors are plotted in figure 6.37. Figure 6.37(a) uses the definition of the error as the difference between the measured value of $V_{centre}^{cell(n)}$ and the measured initial value:

$$\epsilon_{d_a}^{n} = sgn\left(V_{centre}^{cell(n)}(t) - V_{centre}^{cell}(0)\right) \times max.\left\{|V_{centre}^{cell(n)}(t) - V_{centre}^{cell(n)}(0)|\right\} \quad (6.21)$$

Figure 6.37(b) alternatively uses the *target* initial value $V_{centre}^{cell}(0)^{T}$ thus also incorpor-

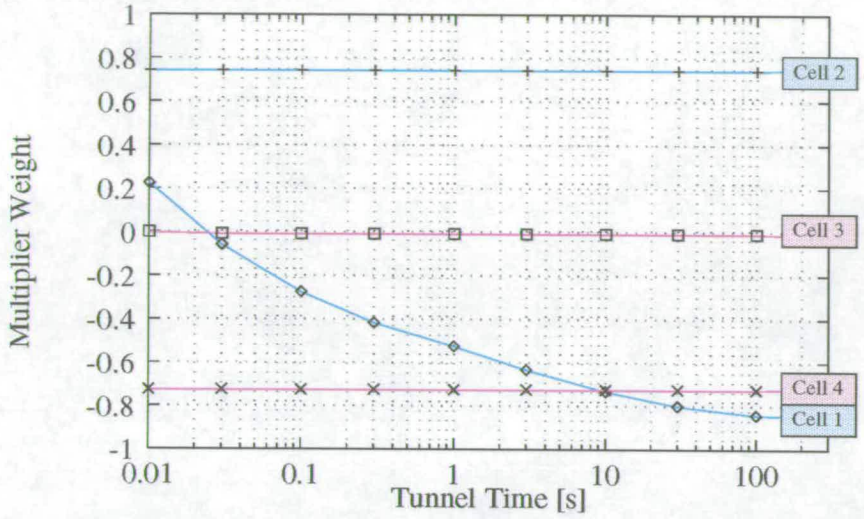


Figure 6.36: Typical programming characteristics: multiplier weight versus time for $V_{PP} = +16$V. Cell 1 is selected and initialised to +0.75. Cells 2,3 and 4 are deselected with centres initialised to +0.75, 0 and -0.75 respectively. $AMAX$ is 10V.

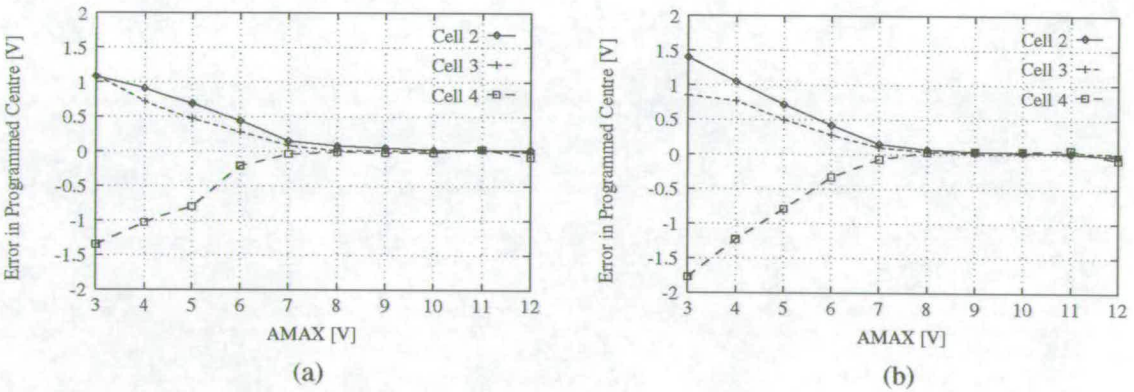


Figure 6.37: Maximum disturb error for deselected cells in dual phase Euclidean distance calculator array. (a) ϵ_{d_a}, (b) ϵ_{d_b}.

ating initialisation error

$$\epsilon_{d_b}^n = sgn\left(V_{centre}^{cell(n)}(t) - V_{centre}^{cell}(0)^T\right) \times max.\left\{|V_{centre}^{cell(n)}(t) - V_{centre}^{cell(n)}(0)^T|\right\} \quad (6.22)$$

Comparison of the two graphs in figure 6.37 shows the significance of initialisation error for low values of $AMAX$. This is because cells are initialised sequentially, thus earlier preset cells may be disturbed because of insufficient clearance. Cell 1 was always initialised last, thus all the other cells exhibit some degree of initialisation disturbance.

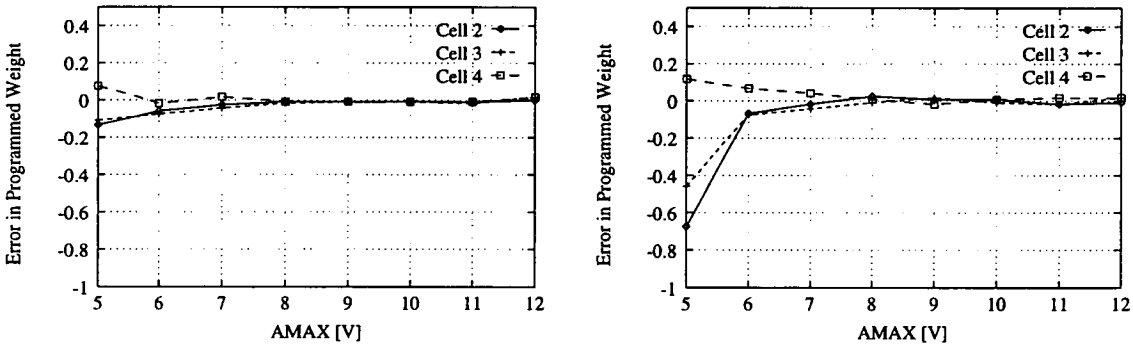


Figure 6.38: Maximum disturb error for deselected cells in dual phase multiplier array. (a) ϵ_{d_a}, (b) ϵ_{d_b}.

Equivalent graphs for the multiplier array are shown in figure 6.38. Notice that here the minimum acceptable $AMAX$ is 5V. This is because $AMAX$ logic levels control the select pass transistors, and thus need 5V to pass the maximum voltage on the integration capacitor onto the shared comparator input line. By comparison, the output of the Euclidean distance calculator cell is the voltage on the gate of a current mirror and this allows $AMAX$ to run down to 3V (although in practice it would be ridiculous to use such a low value).

To provide an estimate of the bit equivalent accuracy provided of both of the dual phase arrays, equation 6.5 was used, setting the tolerance equal to the maximum disturbance error,

$$\xi = \max\{|\epsilon_d^n|, n \in [2,3,4]\} \quad (6.23)$$

This is plotted in figure 6.39 for both dual-phase test arrays.

The system precision found previously is sufficiently high so as not to obscure the programming accuracy estimation. Notice that this accuracy is generally worse than would achieved in practice as cells are unlikely to be required to be programmed

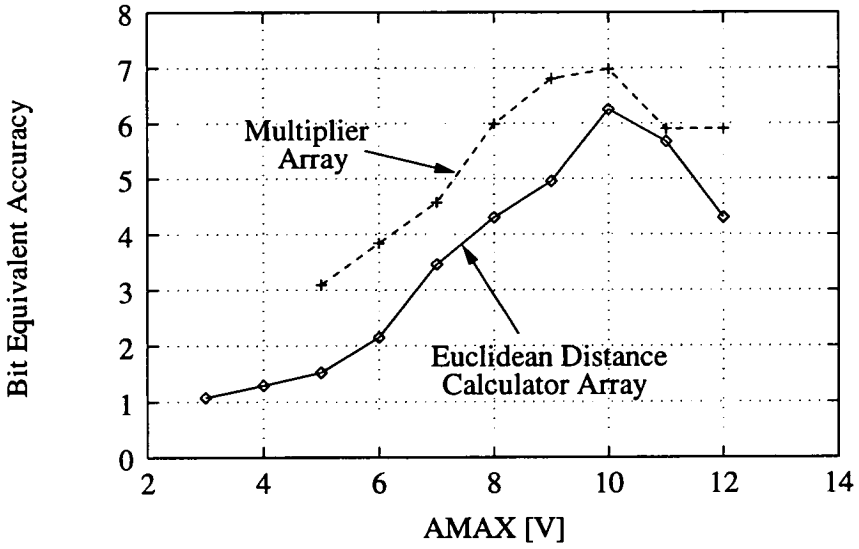


Figure 6.39: Bit-equivalent accuracy of dual phase programming scheme for various $AMAX$ values.

across the entire $V_{fg}(range)$. Notice too that all cells in the experiment were aged roughly equally. If an array contained a mix of very aged cells and very fresh cells, the accuracy is likely to be somewhat worse since the V_{fe} differential will be larger. In practice, however, it is expected that cells are likely to be reprogrammed approximately an equal number of times.

The slightly higher accuracy associated with the multiplier array is probably due to the fact that the initialised values of cells 2 and 4 are not at extrema, but at +0.75 and -0.75 respectively. This was done to provide a margin over non-idealities which would prevent the full -1 to +1 weight range being available. This was not necessary for the Euclidean distance array and extrema 0V and 3V were used. Thus the bit equivalent precision for the multiplier is probably estimated slightly higher than it should be. The use of extrema *is* a realistic metric since in practice it is desirable to use the entire available weight range.

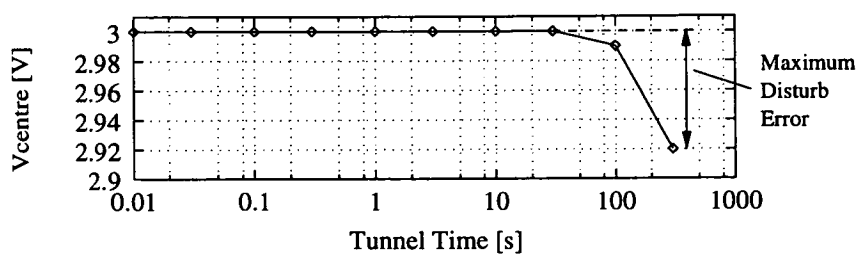


Figure 6.40: Typical centre decay trace

Unexpectedly, for $AMAX$ above 10V, the accuracy began to drop rather than continue to rise. This can be explained by a closer examination of the programming characteristics. In the Euclidean distance array the maximum error in preset values above 10V is due to a drop in the value of $V_{centre}^{cell(4)}$ from its initialised value of 3V during V_{PP}^{+} traces which proceed for the full 300s duration; a typical such trace is shown in figure 6.40. When a large $AMAX$ is coupled onto the floating gate, especially in conjunction with a high ϕ_{fg} (eg. 3V here), the gate voltage of the sense transistor(s) can become quite large. Referring back to table 5.2, the gate oxide V_{fe}^{+} for 100ms pulses is only 17.7V for a fresh device. Here, t_{PP} is three orders of magnitude longer, and additionally the total gate oxide area/perimeter is larger than in $TARDIS$. Therefore it is likely that this is sufficient to induce non-negligible tunnelling currents in the gate oxide of the sense transistor which cause ϕ_{fg} to fall as observed. Notice that there is no converse effect during V_{PP}^{-} traces as the floating gate is never driven negative with respect to the substrate.

This puts a further constraint on $AMAX$ which requires it to be carefully balanced; not merely pushed right to the limit of the technology: too low and poor separability is achieved, too high and secondary tunnelling can be problematic over very long programming periods[17]. Here the best value for $AMAX$ is $\sim$ 10V which provides a programming accuracy of around 6 bits for the Euclidean distance calculators and 7 bits for the multipliers. Notice that this does not make any assumption about the *retention* accuracy.

6.9.3 Retention Characteristics

Having implemented very basic ESD protection on $NEMO$ and taken greater care during handling and maintenance of a static-safe workbench (and avoided high voltage/current experiments), no catastrophic cell failures were observed[18]. This is a very significant improvement over the $TARDIS$ failure rate. 190 floating gates were fabricated on $NEMO$ (19 floating gates on 10 chips). While only about half of these were actually tested all of these were found to operate correctly. Additionally there were no obviously leaky cells with hold times of less that a few minutes. Whilst it was not proven, it added evidence to the claim that the high failure rate of $TARDIS$ floating gates was due to ESD problems rather than inherent in their fabrication.

[17]This constraint is also relevant to the dual injector scheme. In fact the problem is even worse as the poorer separability would normally promote a drive towards a higher V_{DD}.

[18]Several cells were damaged: the whole of chip 1 was damaged before testing and three other individual cells were destroyed as a result of a software glitch which lead to overvoltage damage.

Power-Up Hold

During a power-up hold retention test, an array was first programmed and then V_{centre} or ω continually read. It was expected that this would be a more stringent test than the power-down hold since it simulated the conditions of constant operation under which hot carrier injection effects could occur. However it did require continual use of certain laboratory equipment which was required for other experiments and therefore could only be run for a few days.

Figure 6.41 shows a hold experiment on a Euclidean distance calculator array and figure 6.42 shows an equivalent experiment on a multiplier array. Several similar experiments were run on other chips for shorter durations and were found to produce similar traces (this included very aged chips; aging appeared to have no measurable impact on the drift rate).

These experiments showed a very definite tendency for the programmed value to drift away from its original target, although the multiplier drift was always much less than the Euclideans drift over an equivalent period of time. Additionally, while the V_{centre} drift was always positive, the ω drift was not always consistent although in the majority of cases it was negative (ϕ_{fg} positive drift). Since the drift was very much smaller than for V_{centre} it was difficult to be categoric. Specifically, the drift was entirely unrelated to whether positive, negative or mixed programming pulses had been used to reach the initial target.

A significant feature of these traces - most apparent in the Euclidean array but also visible in the multiplier array - is the fact that the drift is very much more rapid during the first day or so subsequent to programming; indeed the characteristics can be split clearly into two phases or rapid and slow drift. This is unlikely to be attributable to thermal effects since chips and equipment were routinely left powered up from several days in advance of a retention tests.

Power-Down Hold

The retention characteristics of the floating gates whilst left unpowered is a distinct issue; here signal lines are not powered and this may impact on any drift characteristics; additionally hot carrier injection effects should not occur. Since such experiments do not continuously tie up lab equipment, it was possible to investigate longer time scales than for power-up hold experiments. Twenty Euclidean cells (distributed across five chips) were programmed to random targets in the 0-3V range, removed from the test board and left untouched in a sealed box for exactly one week.

Figure 6.43(a) is a histogram of the change in the measured values of V_{centre} when the chips were re-tested. After measurement, the chips were returned (unprogrammed) to the box and left sealed for a further two calendar months. Figure 6.43(b) is a histo-

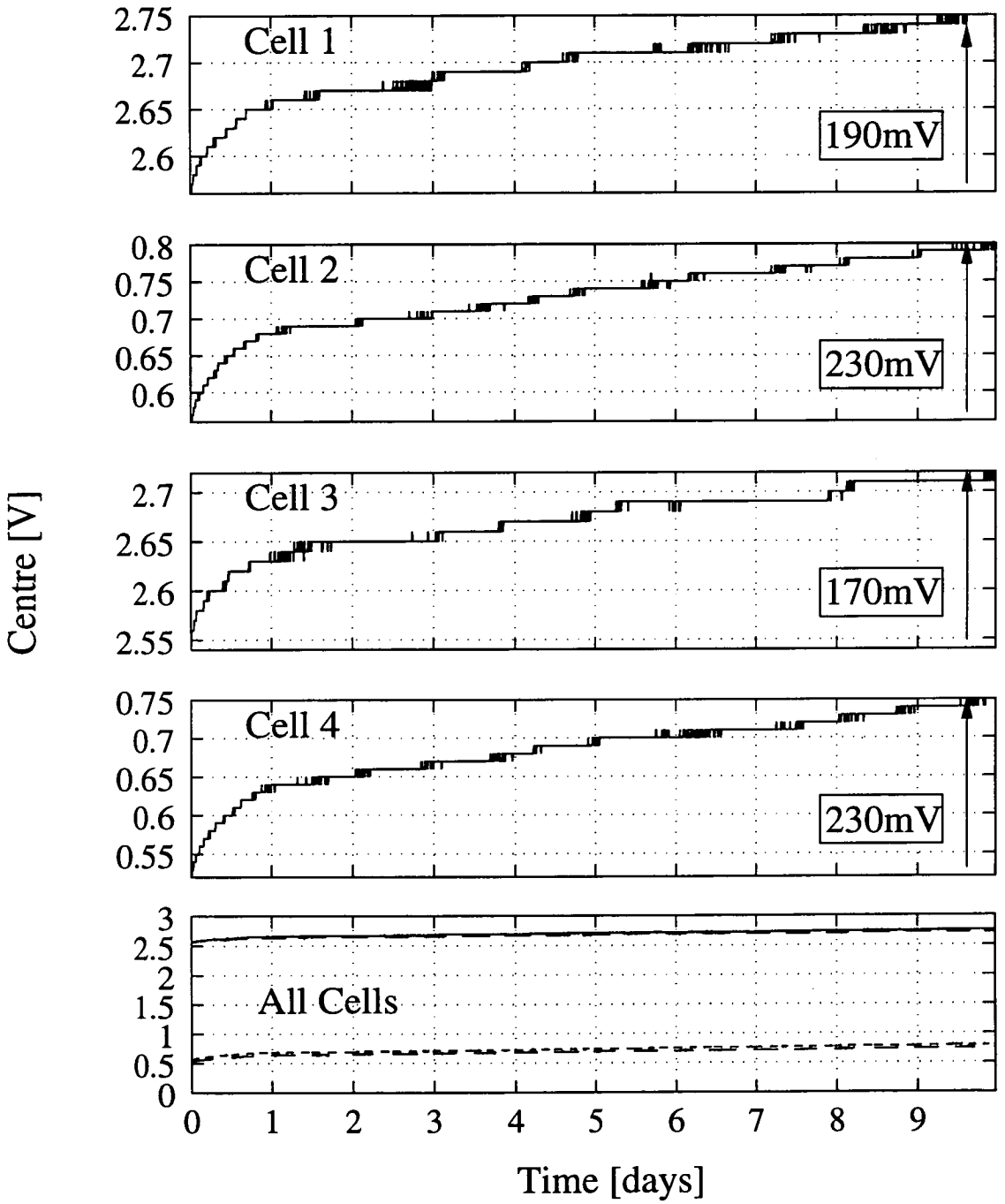


Figure 6.41: Continuous READ operation on programmed Euclidean array: the top four graphs show V_{centre} measurements on the four individual cells of the array, the bottom graph puts the effect of drift on the whole array into context of the overall range.

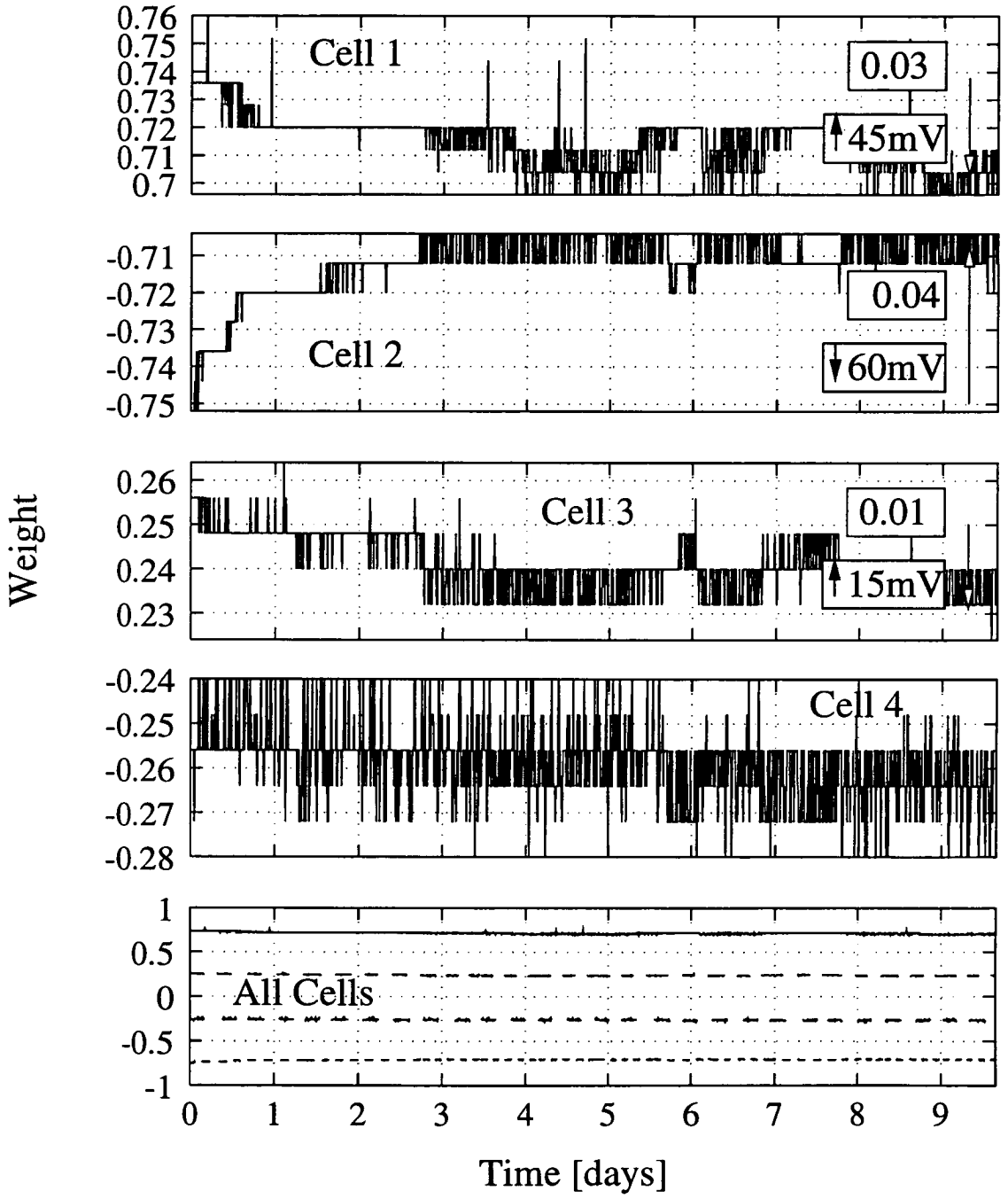


Figure 6.42: Continuous READ operation on programmed Multiplier array: the top four graphs show ω measurements on the four individual cells of the array, the bottom graph puts the effect of drift on the whole array into context of the overall range. The drifts marked in the dark boxes represent approximate floating gate drifts using the approximate differential stage mapping from the measured weight drifts: $\Delta\phi_{fg} = -\frac{3}{2}\Delta\omega$.

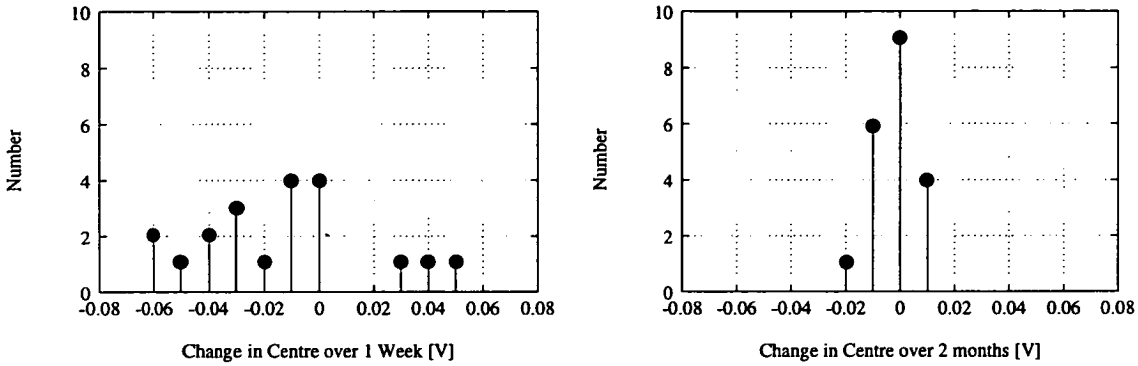


Figure 6.43: Change in programmed V_{centre} for 20 cells (5 chips) over the period of two months unpowered. (a) During the first week, (b) Change from V_{centre} recorded at end of first week.

gram of the change in the value from the one week measurement upon re-testing.

These results mimic the power-up results: a rapid initial change in V_{centre} followed by a slower change. In fact the change illustrated in figure 6.43(b) is so slight as to be within the previously identified precision bounds. Therefore it cannot be stated categorically whether or not further leakage occurs after one week.

The most likely explanation would seem to be that leakage is occurring by the same mechanisms as in the power-up case but much more slowly as there are no driven nodes within the circuit and leakage is therefore due to the gradual redistribution of stored charge. The impact of hot carrier injection cannot be discriminated from this result.

Figure 6.44 shows the power-down hold behaviour of three chips over several weeks. The marked points are measurement events and show that the stored values are held over the period of several months[19].

Retention Failure Investigation

The failure of the floating gate arrays to retain their programming well over prolonged periods of time (particularly when powered up) frustrates their application to VLSI neural parameter storage. While some neural algorithms exist in which a medium term ("forgetting characteristics") memory might be useful, these are not the principal topic of this investigation.

Since powered-down floating gate also show a sharp initial charge loss outwith the bounds expected for relaxation from *TARDIS* experiments, it was believed that an effect other than hot carrier injection was dominating this process. Fluorescent lights,

[19]The noticeable slight drift in cell 1 of chip 2 is *probably* an artifact of measurement due to board recalibration rather than an actual anomaly.

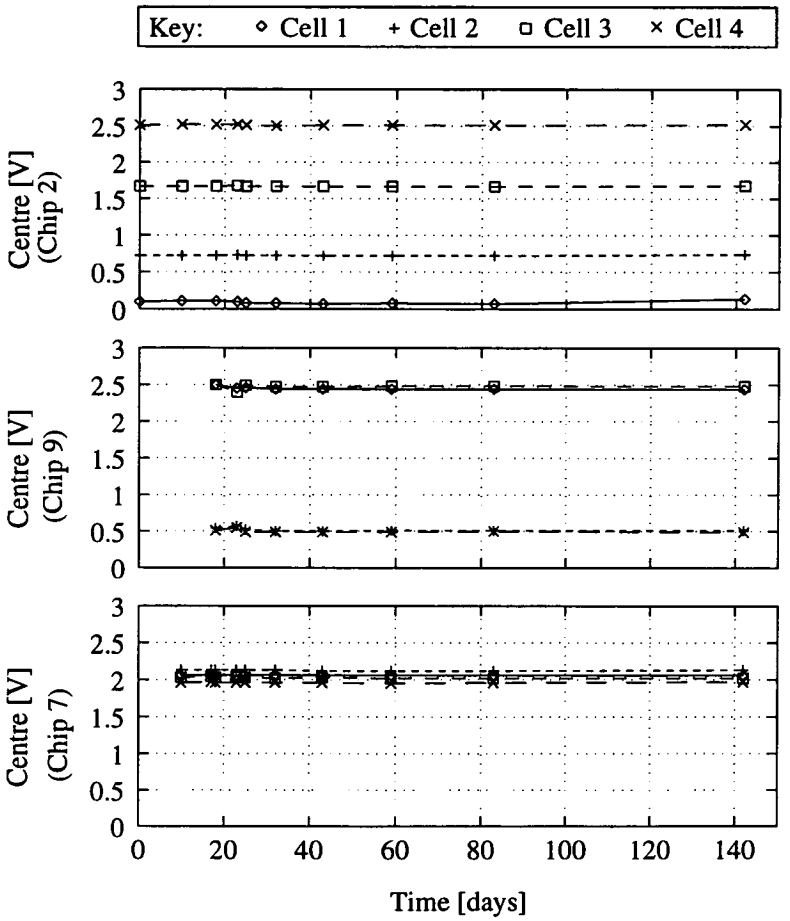


Figure 6.44: Measurement of programmed V_{centre} for 12 cells (3 chips) over a period of several weeks.

as used in the laboratory, have a significant UV content. In case this was leaking though the chip lids, the entire equipment was enclosed in a solid box and retention experiments re-run but this did not reduce drift. Additionally, programmed chips were placed in an EPROM UV-eraser for up to 2 hours but this did not increase drift. UV influence could therefore be eliminated as a cause of leakage.

Since the crude evidence from *TARDIS* was of successfully non-volatile floating gates, it seemed evident that some change in the layout of the floating gate was responsible for the leakage effect. The only change is that described in section 6.5.2. In *TARDIS*, the floating gate had occupied a single layer of poly1. However *NEMO* floating gates run through poly1 and poly2 and are interconnected with contacts to metal1. Obviously the entire structure was still notionally electrically isolated, but the use of such a multi-layer structure now fell under suspicion. This had not caused any initial concern since Lande *et al* had previously implemented multi-layer SLV-CMOS floating gates [96].

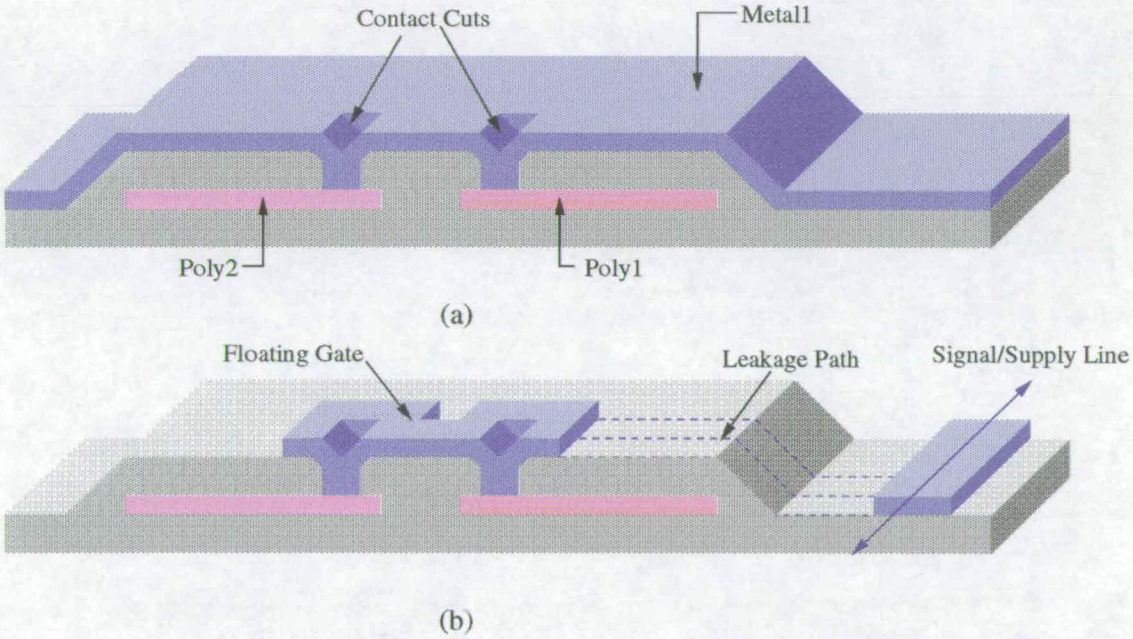


Figure 6.45: (a) Layout cross-section in vicinity of poly1-metal1-poly2 connection subsequent to metal1 deposition, (b) cross-section subsequent to metal1 etching illustrating leakage path between two notionally isolated metal1 tracks.

However, subsequent study has revealed a potential cause of a leakage path due to running the floating gate through contacts. During the fabrication process SiO_2 is deposited over the polysilicon layers and contact cuts made to allow metal1 connection. Physical vapour deposition (PVD) is then used to produce a metal layer across the entire wafer surface which is subsequently patterned by etching to the layout

design [105]. Further oxide layers, contact cuts and metal2 are subsequently added.

However, although the interconnect metal1 is specified as aluminium, it is more likely to be a dilute alloy containing impurities. These impurities - silicon[20] and copper in particular - are difficult to completely remove in a standard aluminium wet etch solution of acetic, phosphoric, and nitric acids [19]. Therefore, when further oxide layers are added, these may not form a perfect insulating surround to the floating gate but rather sandwich a narrow surface of slightly conducting material. This is illustrated schematically in figure 6.45[21] showing how this conductive surface may lead to a leakage path from the floating gate. It is likely that the quality of the oxide surface due to impurities is a poorly controlled process feature and results are unlikely to be consistently repeatable.

In the presence of such a conductive surface, ϕ_{fg} will leak towards the nearest driven voltage line in the power-up tests. Examination of the layout showed Euclidean cell floating gates were very close to V_{DD} lines which is consistent with their leakage behaviour. Multiplier cells were close to analogue signal lines but further than the Euclidean cells were from V_{DD}; again this would explain their variability and lower drift rate (higher surface resistance due to length). It should be possible to modulate the signal/supply voltage close to the floating gate and observe if the ϕ_{fg} drift is towards the value set. Whilst this would provide good evidence for or against this hypothesis, it was unfortunately not possible practically to run the experiment for a sufficiently long time (and it is also probable that the conductive surface is not uniform in all directions).

Unfortunately this fault obviously precludes any further research into the long term hold and reprogrammability capabilities of a properly functioning floating gate array.

6.10 CHE Flash

Whilst the dual-phase scheme removes the requirement for design-rule violating layout, it requires a global signal which sweeps negative with respect to the substrate[22] in order to inject electrons onto the floating gate. Experiments with *TARDIS*, however, show that it is also possible to use CHE to inject electrons, and this requires only positive potentials which are below the maximum power supply rating. This gives the advantage of possible further future integration as it is possible to switch these voltages

[20] Actually, silicon is often deliberately added to the aluminium to counter silicon consumption at contacts.

[21] Some fabrication processes use different etch methods, eg. electron beam lithography [170], and therefore this problem may not exist for other implementations

[22] More precisely, the global signal must sweep negative with respect to V_{fg}, but due to the specifications of the technology this voltage is always sufficiently low that the global signal will also sweep below the substrate potential.

using normal transistor arrangements.

The third prototype addressing scheme then uses CHE inject, FN remove. The high voltage FN remove wire is distributed globally and therefore does not have to be switched. This leads to a 'flash' scheme, where the array is block 'erased' by electron removal (analogous to CHE-type digital Flash memory) and then each cell is programmed individually by CHE inject.

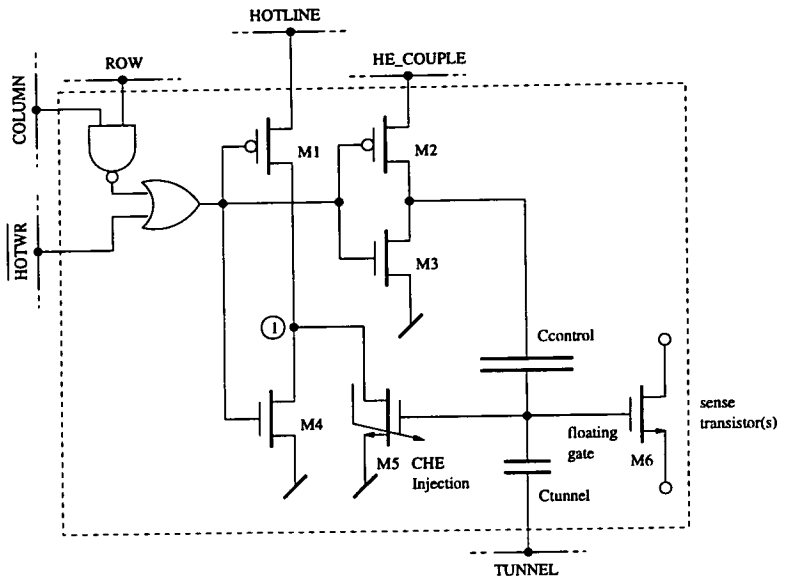


Figure 6.46: CHE/Floating Gate Cell Component

A CHE/floating gate cell component is illustrated in figure 6.46. $M5$ is the (minimum length) CHE injection transistor. Unlike in *TARDIS*, $M5$ does not also act as this sense transistor, rather the floating gate also drives the gate of $M6$ which is part of the parameterised circuit and may therefore be sized more appropriately. Injection occurs when $\overline{HOTWR}$ goes low and a high voltage is switched onto node ① through $M1$ and V_{fg} is pulled high by coupling of V_{PP}^{che} on the HE_COUPLE line through $C_{control}$. During CHE inject, $M1$ and $M5$ are both on and thus form a potential divider circuit. Since $M5$ is intended to have a high drain voltage and a high gate voltage $(\phi_{fg} + m_c V_{HE\_COUPLE})$ it will thus be very conductive. $M1$ must be ratioed such that it is sufficiently conductive to pull node ① high enough for CHE. A p-type transistor is preferred for $M1$ because

- It will pass the full value of HOTLINE, up to the maximum rated supply.
- The higher doping (due to the n-well) reduces the liability to punchthrough since the width of the depletion region is inversely proportional to the doping concentration [155].

Unfortunately the lower p-transconductance and the operating conditions make this transistor very wide, dominating the cell. Results from *TARDIS* might suggest that this cumbersome device is unnecessary since unselected devices would not have high coupled-in gate voltages (thus $V_{fg}(unselected) \sim \phi_{fg}$) and would not induce any significant hot electron injection. Unfortunately, the $M5$ of unselected cells would still pull a very large current due to their high drain voltage if connected directly to HOTLINE. This would result in two problems:

- Metal migration: HOTLINE would have to be very wide to handle the current requirements of a large array.
- Breakdown protection: since HOTLINE is shared between all cells in the array it would be impossible to select an appropriate current limit to prevent breakdown for a random array state. For example, if all deselected cells have $\phi_{fg} = 0$V, *all* the current would flow into the selected cell and the limit would have to be set low. However, if the selected cells have $\phi_{fg} = 3$V, these would each draw significant current in addition to the selected cell and the current limit would have to be set much higher to allow node ① reach a CHE threshold. Of course it is possible to limit HOTLINE to less than 7-8V and omit current limitation - however *TARDIS* showed that programming with such a low drain-channel electric field is prohibitively slow.

One solution might be to have $M3$ connect to a normally 3V signal line instead of V_{ss}, and only to pull this signal line to V_{ss} during a CHE inject cycle, ensuring that the $M5$ of all unselected cells was off. Or more involved re-design might involve parameterising cells using $\phi_{fg} < V_{T_n}$. Unfortunately, *TARDIS* showed that the floating gates returned from fabrication with generally high initial ϕ_{fg} and so this would not be sufficient to switch the $M5$s off. TUNNEL could be initially pulsed negative to 'initialise' all the cells to a moderate value but this procedure is now becoming quite complicated and violates the proposal of using only positive substrate-referenced voltages. For these reasons, the large $M1$ switch was retained. As used latterly with *TARDIS*, a 2.5mA current limiter was placed between the high voltage source, V_{CHE} and the HOTLINE terminal[23].

When $\overline{HOTWR}$ is high, the cell is in read mode and V_{ss} is selected to couple onto the cell through $C_{control}$. Additionally transistor $M4$ deliberately discharges node ① which otherwise might gradually discharge by sub-threshold currents in $M5$ depending on ϕ_{fg}; this might lead to inaccurate circuit measurements if taken immediately after a CHE pulse with node ① pre-charged high.

[23]Observed maximum currents on HOTLINE during a selection of experiments ranged from 1.3mA to 1.6mA.

6.10.1 Programming

In common with *TARDIS* programming attempted to match V_{fg} to V_{drain} by a model-based approach. This gives (allowing a constant offset term):

$$V_{pp}^{che} = \frac{1}{m_c}(V_{drain} - V_{centre}) + offset \tag{6.24}$$

Notice that here, there is an unknown potential drop across both the current limiter and $M1$. Thus V_{drain}, the voltage at node ①, is certainly not the voltage applied at the input to the current limiter, V_{CHE}. Thus, initially, V_{drain} is assumed to be equal to V_{CHE}, and the voltage drop is intended to be accounted for in the $offset$ term.

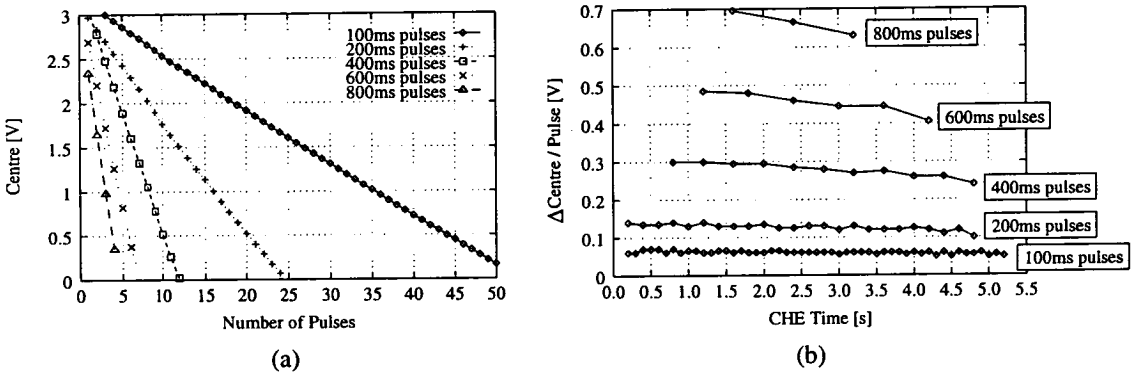


Figure 6.47: (a) Operating-range traversal programming characteristics for Euclidean distance cell for 100-800ms V_{PP}^{che} pulses. (b) Change in V_{centre} caused by each pulse in the sequence shown in figure (a) plotted against total CHE injection time.

To test the validity of the model across the entire Euclidean cell floating-gate operating range, several cells were preset to $V_{centre} \sim 3V$ by V_{PP}^{+} Fowler-Nordheim pulses. They were then allowed to traverse the entire range down to 0V or less by application of V_{PP}^{che} pulses of 100ms to 800ms duration calculated from equation 6.24. $offset$ was held at zero for this experiment. Typical characteristics are shown in figure 6.47(a). In figure 6.47(b), the characteristics of figure 6.47(a) have been replotted in the form of change in centre, δV_{centre} against the total CHE inject time. Two observations can be made from this graph:

1. For short CHE pulses (100ms/200ms), $\delta V_{centre}/Pulse$ is almost constant, showing that the model of equation 6.24 is valid across the entire range. This is only valid for the shorter pulses as these approximate continuous time feedback, whereas for longer pulses, non-linear effects have sufficient time to become significant between recalculations of V_{PP}^{che} to accommodate these effects.

2. The total injection time required for short CHE pulses (100ms/200ms) is longer than for long pulses. This is directly contrary to behaviour in *TARDIS* and suggests that a zero offset is actually *too high*.

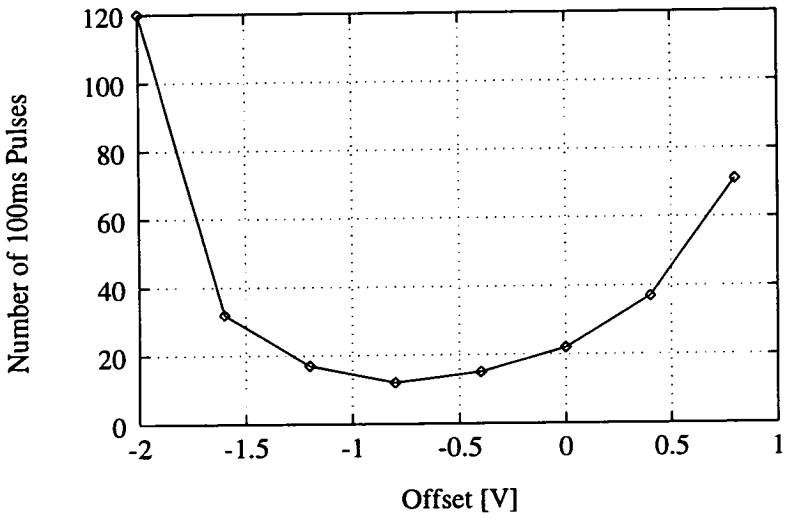


Figure 6.48: Number of 100ms CHE model-based pulses required for complete traversal of the Euclidean cell operating range ($3V \rightarrow < 0V$) for a selection of different model offsets.

To confirm the last point, the experiment was re-run for a range of offsets (limited by the requirement of keeping $V_{PP}^{che} \leq$ 12V) for 100ms pulses. The number of pulses for complete 3V $\rightarrow$ 0V traversal is shown for a typical run in figure 6.48. All cells showed an optimal offset (minimum number of traversal pulses) in the range of about -1V to -0.5V. This result is not surprising compared to *TARDIS*, when it is considered that V_{drain} is lower than V_{CHE}, the required target V_{fg} must be consistently lower. Notice that this again could be adapted for real continuous time feedback by connection of node ① to the non-inverting input of an amplifying stage as described for *TARDIS*.

6.10.2 Programming Accuracy

To test the accuracy which could be achieved with the CHE flash array, the algorithm pictured in figure 6.49 was developed. As for the Fowler-Nordheim schemes, this algorithm was developed empirically from observation of programming behaviour and is not directly derived from any injection model. Once again the motivation was simplicity for easy eventual integration. Once again vague terms such as 'likely undershoot' and 'little change in centre' could be quantified by observation. Best results were obtained by setting the 'likely overshoot' flag if the last pulse moved the centre more than

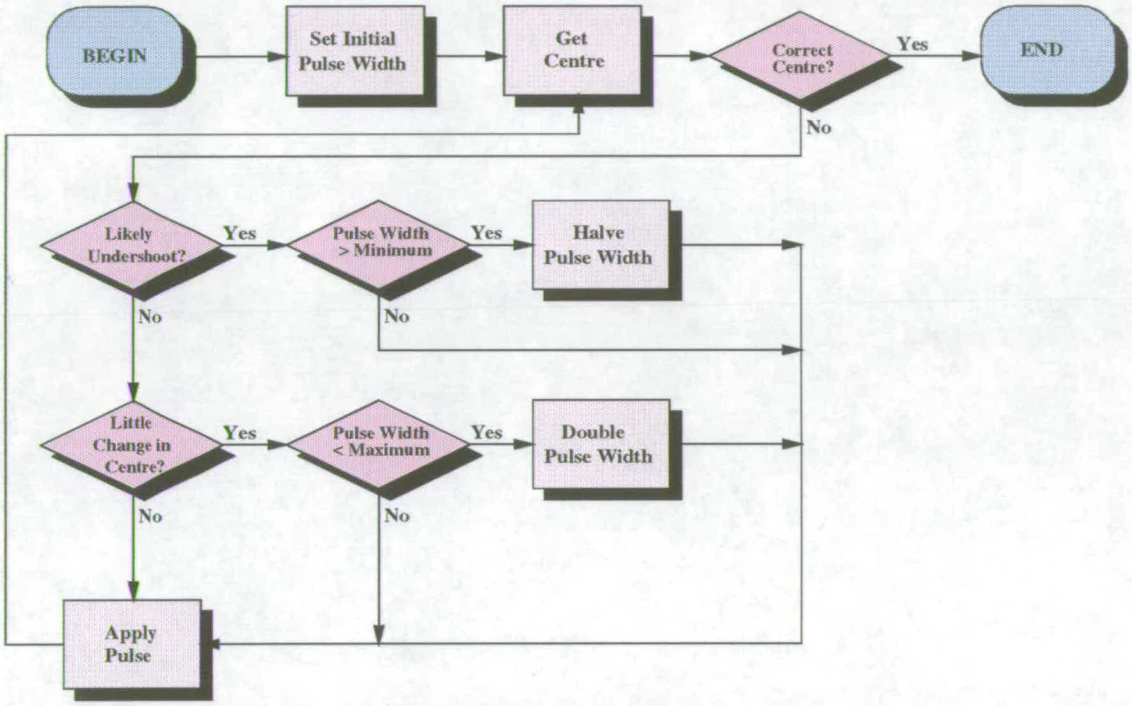


Figure 6.49: Flowchart of programming algorithm for *NEMO* floating gate Euclidean Array (omitting Fowler-Nordheim preset).

70% of the way to the target value from the preceding centre measurement, and 'little change in centre' was flagged if the last pulse moved the centre less than 20% of the way. The minimum pulse width was set at 25ms (this was required or else programming became intolerably slow). Arrays were flash preset by Fowler-Nordheim pulses to the global tunnel line.

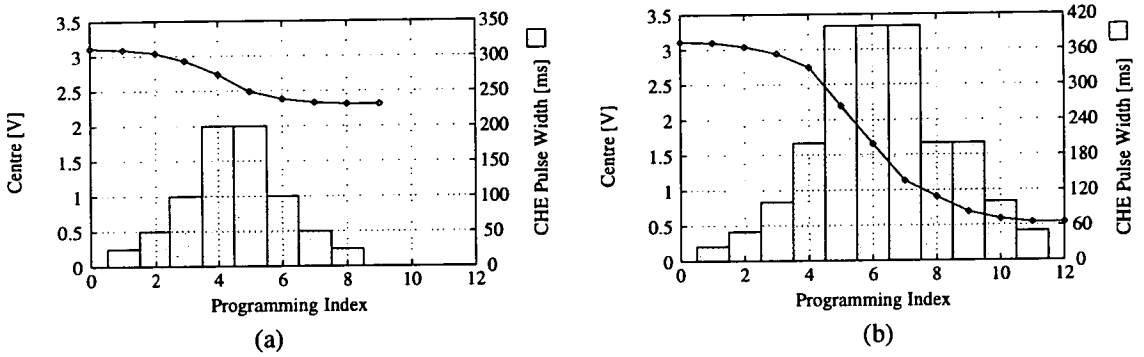


Figure 6.50: (a),(b) CHE programming traces.

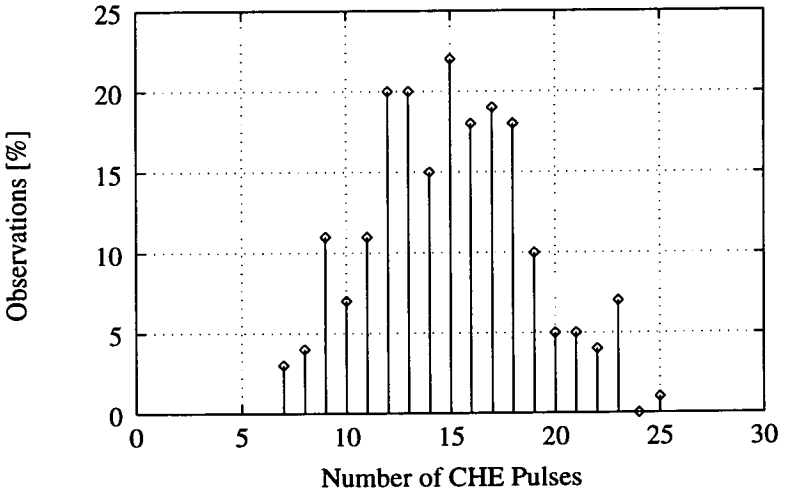


Figure 6.51: Number of CHE pulses required to program centre to target from flash erase initial state. 200 trial experiment.

Selected cells were programmed to random centre targets; two typical programming traces are illustrated in figure 6.50. The number of CHE pulses required in a 200 trial experiment is shown in figure 6.51 (excluding the Fowler-Nordheim preset pulse). Figure 6.52 shows the measured deviation of the programmed centre about the target for the same experiment. This shows a skew of deviation where $\geq$ 95% of trials resulted in a centre $\leq$ target (the other cases probably being mainly due to limited

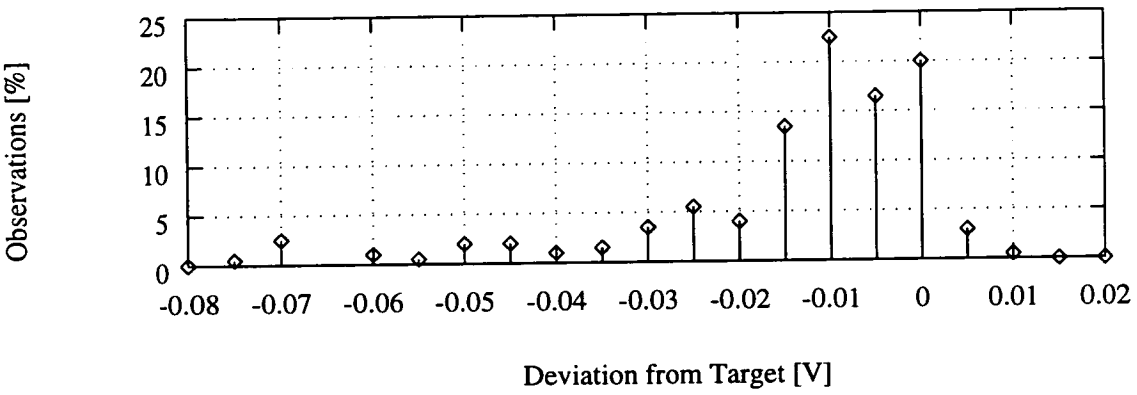


Figure 6.52: Deviation of programmed Euclidean distance cell from its target centre for 200 runs.

precision measurement). This is to be expected since it is possible to undershoot the target but not overshoot (since the algorithm would simply then apply another CHE pulse). However, by preincrementing the target before programming by about 15mV (ie. programming to a pseudo-target 15mV higher than the actual target) then it is possible to effectively shift the deviation axis 15mV to the left and thus introduce a better distribution about the actual target. This increases the bit-wise accuracy above 6.5 bits for $> 80\%$ of trials.

Notice however that this is significantly poorer than the performance of the Fowler-Nordheim algorithm (see figure 6.17) for a similar number of programming pulses (centre evaluations). The main reason for this is the lack of ability to deal with undershoot. In both the directly-pinned out and dual-phase arrays, undershooting and overshooting the target can be counteracted simply by reversing the polarity of V_{PP}. It is not possible to do this with a flash scheme since another Fowler-Nordheim pulse would re-preset the entire array and loose all previous programming.

A second problem was found with the 'flash' method: *over-erasure*. During the Fowler-Nordheim erase, some cells are erased more than others due to the inter-device variability studied on *TARDIS*. In order to ensure that the hardest-to-erase device is completely preset, the easiest-to-erase device may become preset $\gg$ 3V. Since the parameterised Euclidean cell limits the range of floating gate voltages which can be distinguished, it is not possible to know exactly what this voltage is. It therefore becomes impossible to apply the correct V_{PP}^{che} according to the model of equation 6.24. If the estimated V_{PP}^{che} is then sufficiently poor, the hot electron current will be greatly reduced and programming will be very slow. Neither is it possible to simply apply a long CHE pulse (a secondary preset), to move the centre back into the range of measurement since during this pulse the floating gate may enter the positive feedback regime of the hot electron curve and thus massively undershoot both 3V and the intended tar-

get during the long period of this pulse. Secondary preset must then consist of many very short CHE pulses with re-evaluations between each which introduce significant time overhead. This problem does not prevent 'flash' erase working, it simply makes it more complicated and much slower than would be desirable.

Looking ahead to downloading a neural weight set, it might be desirable to perturb the downloaded weights according to a chip-in-the-loop algorithm, rather than having to completely rewrite them for each training update. In this case, again, a 'flash' method is unsuitable.

Undershooting, overshooting (hence overerasing) is a fundamental limitation of iterative SLV-CMOS floating gate programming: the devices are highly non-linear, badly matched and somewhat stochastic (due to trapping and detrapping events). No reasonable algorithm will be able to avoid missing targets to a certain extent and no algorithm can properly preset several badly matched floating gates with a single connection. Thus the lack of bi-polar programming functionality is thus a serious deficiency. 'Flash' block erase can therefore be concluded to be inappropriate for an iteratively programmed array unless the required accuracy is low. Notice however, that 'flash' erase *can* be applied if the floating gates are being used digitally (single bit accuracy is suitably low!) or if programming is done in continuous time (by feedback) – in a non weight-perturbation context – rather than iteratively, so that instantaneous events can be responded to immediately, rather than at the end of a programming pulse period.

From the experimental evidence from the dual-phase array it is straightforward to incorporate cell-wise erase into the CHE array without relaxing the constraint on positive-only substrate-referenced programming; the digital logic can be reconfigured to couple HE_COUPLE onto $C_{control}$ of a deselected device without turning on $M1$. Cells could then be deselected by coupling a large voltage onto the floating gate during the preset pulse.

6.10.3 CHE Program Disturbance

To determine inter-device programming disturbance, the cells in an array were preset and then programmed sequentially, with the programming of the last cell traversing the entire floating gate operating range. The centre of all cells were continuously monitored through programming. Due to overerasure, preset was problematic and only a couple of trials were attempted. One such trace is shown in figure 6.53. As expected, the inter-device disturbance was found to be negligible (below the measurement precision). This is because $M1$ acts digitally and entirely prevents any hot electron currents (and hence parasitic reprogramming) in deselected cells. This is an improvement on the slight disturbance noticed in the dual-phase array; it should be noted that this would be reintroduced if a differential Fowler-Nordheim increment was incorporated

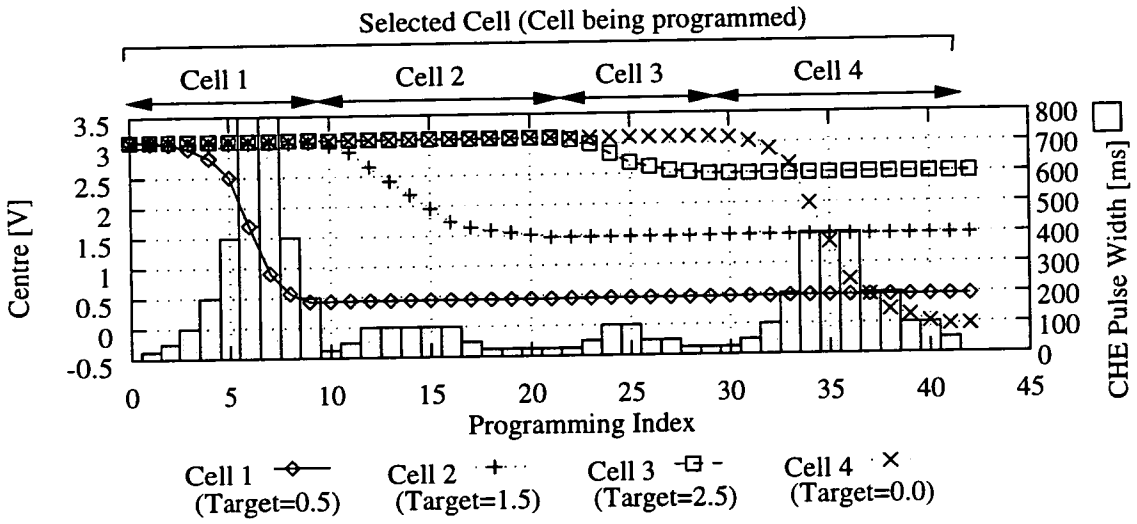


Figure 6.53: Euclidean array-wide CHE programming trial.

as proposed. However, this would seem to be a minor cost to allow bi-polar iterative programming.

6.10.4 CHE Multiplier Array

A CHE Multiplier Array was also constructed using the multiplier design coupled to the floating gate arrangement shown in figure 6.46. It was originally intended to test this in a similar way to the Euclidean array.

However, due to the 'companding' of the differential stage in the Euclidean array this would further hide the true value of ϕ_{fg} and thus exacerbate the overerase problem. Additionally, whilst the basic functionality of the floating gate cells would be expected to be the same as for the Euclidean distance array (since the programming range, 0V to 3V, is the same) and the parameterised cell functionality is expected to be the same as for the dual-phase array, little additional information is likely to be gained (at the expense of a significant further investment of time; it was preferred to start testing *PARAFIN* instead). For this reason no experiments were done on the CHE multiplier array.

6.10.5 Programming Retention

To briefly examine the retention characteristics, the array programmed in section 6.10.3 was held in continuous READ for just over two days. The data collected is plotted in figure 6.54. The rapid $\sim$ 1 day leakage followed by the slower leakage was found as

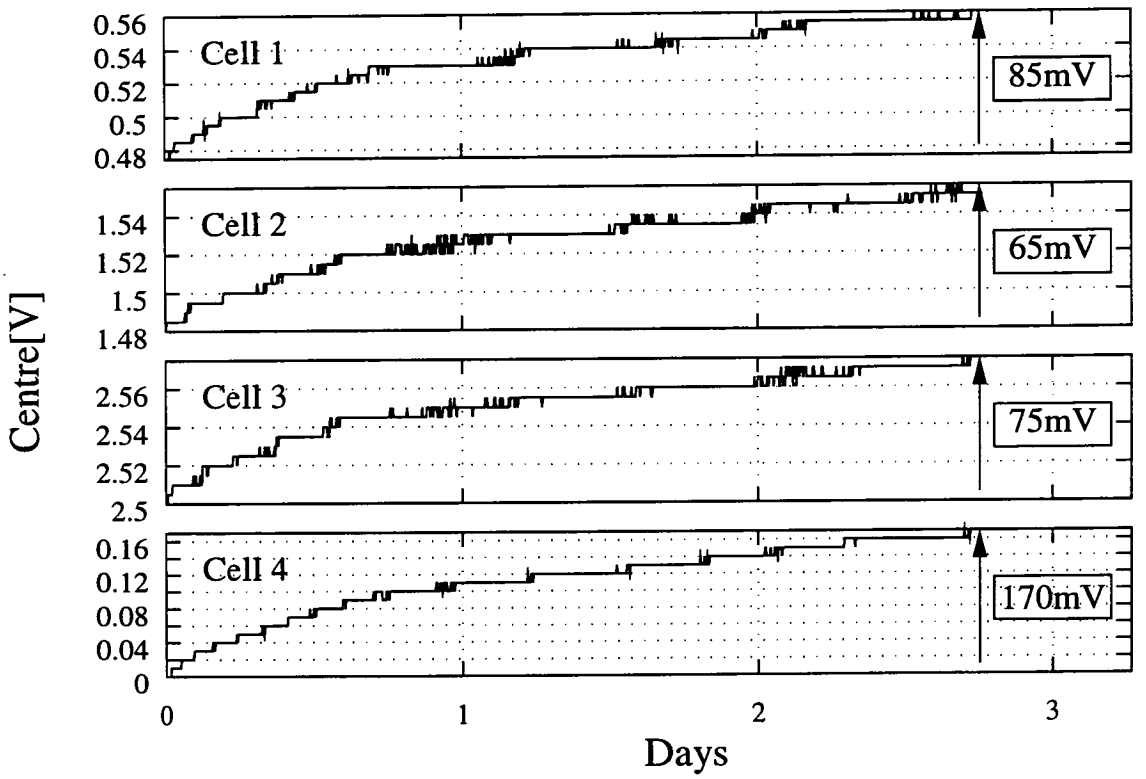


Figure 6.54: Continuous READ operation on programmed Euclidean array.

in the dual-phase array. This was to be expected as the layout around the floating gate is similar and its bias in READ mode is the same.

6.11 Conclusions

The work of this chapter demonstrated the feasibility of floating gate pulse-stream RBF subcircuits. Their programming and programming fidelity have also been investigated. Furthermore, three approaches to on-chip addressing have been shown to work to various degrees of success. However various implementation difficulties have been highlighted which would be required to be overcome. In particular problems with leakage have prevented any further work on long-term precision and non-volatility, so these questions remain open. Therefore it is reasonable to claim that two of the three initial motivations for *NEMO* have been fulfilled.

In this section various topics which have arisen from work with *NEMO* are briefly discussed.

6.11.1 RBF Subcircuits

It has been demonstrated that the analogue Euclidean distance calculators and multipliers developed for the RBF project are amenable to adaption to non-volatile parameterisation using SLV-CMOS floating gates. The network weights can be iteratively established with reasonable accuracy.

A large storage capacitor has been used in the case of the Euclidean distance calculator and a buffer in the case of the multipliers, to improve the functional characteristics of these cells. Without a more detailed study of complete RBF performance it is not possible to determine whether these are essential to RBF functionality, or whether they may be removed to implement more compact cells.

However it has been demonstrated that the compensation transistors in the Euclidean distance calculator are not successful in the presence of inter-device mis-match. It has been shown how these may therefore be removed leading to a more compact hidden layer implementation.

6.11.2 Comparison of Programming Techniques

A summary of the three programming systems tested on *NEMO* is given in table 6.4. All have various advantages and disadvantages which would have to be considered in the context of a specific application: power consumption, layout area, process tolerance, external support, yield, lifetime, etc.

	High-Voltage Switches	Dual-Phase Programming Scheme	CHE Flash
Features	High voltage devices	Global programming signal line	High drain voltages and currents
DRC Violations	Yes	No	No
Cell Programming	Bi-directional	Bi-directional	Uni-directional (Flash erase). Can be modified.
Power Consumption (Programming)	Low (Can be improved)	Low	High
Disturb Protection of Deselected Cells	Good with cross-coupled switch acting as local high voltage driver, poorer with differential stage switch or with banked addressing	Fair: depends on $AMAX$ and inter-device variations	Good
Extensible to Feedback Programming	Yes	Yes	No (V_{fg}(READ) $\neq V_{fg}$(INJECT)†.

Table 6.4: Comparison of programming systems tested on *NEMO* (†Modification to allow cell-selectable Fowler-Nordheim electron-removal *would* permit extension to continuous-time feedback).

The high-voltage switches provide the simplest solution to programming floating gates; floating gates may be addressed directly using digital circuits. For best performance two switches are required to act as high voltage drivers at each floating gate (one on the **T** terminal and one on the **C** terminal). However the penalty for this is the large circuit area and the inconvenience of using DRC non-compliant layout.

The dual-phase scheme provides a more compact and compliant alternative although its resolution depends on the inter-device variability, $AMAX$ and the array size (thus disturbance of non-selected cells is possible since FNT characteristics are not those of an ideal diode) . Here the optimal $AMAX$ was found to be $\sim$ 10V which may be beyond the capabilities of some other processes. This value is much higher than was originally expected would be necessary to cover the range of tunnelling capacitor behaviours (by way of comparison Diorio *et al* [55] implemented a programming array which also used a tunnelling potential differential to deselect cells; in this paper $< 0.01\%$ disturbance of deselected cells was reported for a differential of only 4V). The nature of this poor performance is not certain but it probably attributable to two causes:

1. The large range of valid ϕ_{fg} allowed by the design.
2. The large $V_{fe}^{\pm}$ distribution due both to inherent mis-match and trap-up.

3. The asperity-dominated nature of tunnelling.

To expand on the third item, refer again to figure 5.15. For tunnelling from a planar surface, the emission probability increases over orders of magnitude for a small increase in the applied electric field in the oxide (in the range of interest of $\sim$ 2MV/cm to $\sim$ 5MV/cm). However, for tunnelling dominated at sharp asperities (small radii of curvature), the emission probability has a sharp knee above which the increase in emission probability is slow. Tunnelling current can therefore *not* be expected to vary over orders of magnitude for a small number of volts change in V_{tunnel}. Additionally, the random nature of the asperities and their greater trap-up susceptibility have a strong impact on the second item in the list.

This observation has important implications for design of circuits using interpoly tunnelling capacitors; use of a differential tunnelling potential scheme for deselection is likely to be poor or demand very high differential voltages to work successfully as shown here. If neither high voltage capabilities nor a more planar oxide[24] are available then an alternative programming scheme must be used.

CHE flash allows good deselected device disturbance with a DRC compliant design and uses voltages within the maximum tolerance of the fabrication process which gives scope for total on-chip integration. The high power consumption during programming may be a problem in some applications and the large drain-switch transistor is certainly inconvenient (although this could be eliminated it would compromise reliability). Subthreshold CHE injection in p-channel transistors may be an interesting alternative to investigate but programming has been shown to be very slow [55]. The flash erase scheme implemented was clearly not well suited to iterative analogue programming with poor device matching. It is possible to introduce cell-selectable erase into this system without much overhead but at the cost of reducing the deselected disturbance to dual-phase levels.

It was found that the evaluation phase dominates the programming time requirement, and so the slightly slower injection of hot electrons compared to tunnelling is negligible. Depending, again, on the application it may be preferable to loosen the constraints on programming mode specific circuits such that floating gate voltages may be buffered. Evaluation time was longest on the Euclidean distance cells and since the floating gate in these was large to account for the coupling from the source of the sense transistor, it may instead be desirable to buffer it in any case.

[24]It would be interesting to build a dual-phase array based on gate oxide tunnelling capacitors to compare their performance.

6.11.3 Layout

It is now obvious that the floating gates must *not* run through metal1 since this is the most likely cause of the poor retention performance. This does not prevent any circuits being implemented but does make the layout less convenient and may impede integration density. Only then can a reasonable analysis of retention capability be sensibly attempted.

Chapter 7

PARAFIN: Feedback Programming of RBF Arrays

7.1 Introduction

This chapter describes the *PARAFIN*[1] chip which was developed to investigate continuous-time feedback programming of the floating gates in pulse-stream VLSI RBF arrays. This approach was interesting because it avoided the two disadvantages of iterative programming mentioned in section 6.3.1: slow speed and reduced lifetime (caused by aging due to multiple programming pulses). It was therefore desired to extend the iteratively programmed RBF subcircuit arrays of *NEMO* to incorporate continuous-time feedback so that they could be programmed with the minimum of time and charge transport such that the two approaches could be practically compared.

7.1.1 Design Motivations

The motivations for the *PARAFIN* chip were to:

1. Build on the successful floating gate RBF sub-circuit arrays designed on *NEMO* to incorporate continuous-time feedback such that downloading of weight sets from software models can be achieved in a short time with minimum device degradation.

2. Investigate whether the behaviour of continuous-time feedback programmed circuits are suitable for chip-in-the-loop training memories.

3. Practically compare the continuous-time feedback test arrays with the iteratively programmed test arrays on *NEMO* to establish the most promising approach to the construction of a complete VLSI RBF sensor processor.

[1] *P*ulsed *A*nalogue *R*BF *A*rrays using *F*owler-Nordheim *I*nduced *N*on-Volatility.

7.1.2 Choice of Addressing Scheme

Due to the limited time available to design and test *PARAFIN*, work was initiated before the addressing schemes on *NEMO* had been fully evaluated. The Dual-Phase scheme had been demonstrated as functional whereas the high voltage switches and CHE flash array had not yet been tested. Therefore the dual-phase scheme was incorporated as the addressing scheme on *PARAFIN* and expanded to support continuous-time feedback. In retrospect this was still a reasonable decision since the high voltage switches would require more area and CHE flash suffered from over-erasure. However both techniques would also be applicable to the continuous-time feedback circuits proposed and could be incorporated with only modest changes to the design.

7.2 Review of Programming Techniques

Before describing the design, some circuit-based floating gate programming techniques which have already been proposed are outlined briefly, along with explanation as to why they are unsuitable for *PARAFIN*.

7.2.1 Decaying Sinusoid

Vittoz *et al* [166] propose an ingenious method based on a simple floating gate structure similar to the test elements on *TARDIS* (see figure 7.1(a)). The target value, ϕ_{fg}^{T} is applied to the TUNNEL terminal whilst a decaying sinusoid, A, is applied to the COUPLE terminal:

$$A = |V_{PP}| \cdot \exp\left[-\left(t - t_0\right)\theta\right] \cdot H\left(t - t_0\right) \cdot \sin\left(2\pi f\right) \tag{7.1}$$

where $|V_{PP}|$ is the initial sinusoid magnitude, t_0 is the programming start time, θ is the damping factor, f is the signal frequency and $H(\cdot)$ is the Heaviside step function.

During a rising cycle of the sinusoid, V_{fg} follows A by capacitive coupling until it reaches $V_{fe}^{+} + \phi_{fg}^{T}$. Thereafter, since the potential across the tunnelling capacitor is now V_{fe}^{+} it is sufficient to cause significant tunnelling of electrons onto the floating gate preventing it from rising any further. Therefore removing the component of V_{fg} due to capacitive coupling gives

$$\phi_{fg}^{+} = V_{fe}^{+} + \phi_{fg}^{T} - m_c|V_{PP}| \exp\left[-\left(t - t_0\right)\theta\right] \tag{7.2}$$

Similarly, when A swings negative, V_{fg} follows until it reaches $V_{fe}^{-} + \phi_{fg}^{T}$ and then

$$\phi_{fg}^{-} = V_{fe}^{-} + \phi_{fg}^{T} + m_c|V_{PP}| \exp\left[-\left(t - t_0\right)\theta\right] \tag{7.3}$$

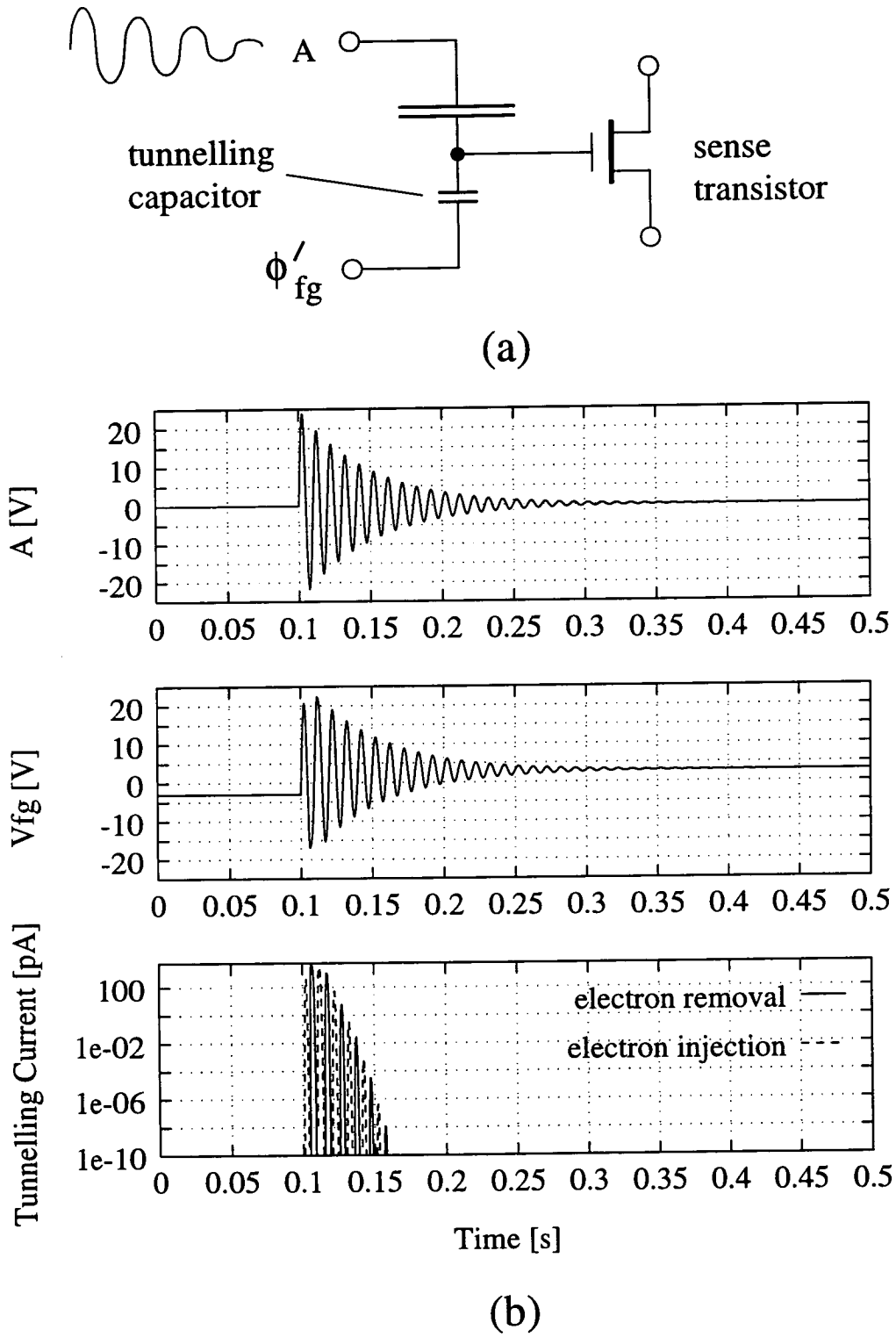


Figure 7.1: (a) Floating gate element for decaying sinusoid programming, (b) `Hspice` simulation of programming ϕ_{fg} from -3V to +3V using a decaying sinusoid. Here $V_{PP} = 25$V, $t_0 = 100$ms, $f = 100$Hz and $\theta = 20$.

As t increases, A declines, tunnelling declines and then stops and so ϕ_{fg} stabilises. At this point $\phi_{fg} = \phi_{fg}^{+} = \phi_{fg}^{-}$. This can be calculated by equating expressions 7.2 and 7.3:

$$V_{fe}^{+} - V_{fe}^{-} = 2m_c|V_{PP}| \exp\left[-(t - t_0)\,\theta\right] \tag{7.4}$$

If symmetric tunnelling is assumed, $V_{fe}^{+} = -V_{fe}^{-}$, then equation 7.4 can be expanded to give

$$|V_{PP}| \exp\left[-(t - t_0)\,\theta\right] = \frac{V_{fe}^{+}}{m_c} \tag{7.5}$$

Substitution of equation 7.5 into equations 7.2 (or 7.3 with $V_{fe}^{-} = -V_{fe}^{+}$) then gives

$$\phi_{fg} = \phi_{fg}^{T} \tag{7.6}$$

which is as required. This allows ϕ_{fg} to be programmed across a wide range of values[2]. Figure 7.1(b) shows the simulated programming of ϕ_{fg} using the `Hspice` model of the tunnelling capacitor developed in section 5.5.

However, as shown with *TARDIS*, tunnelling asymmetry means that $V_{fe}^{+} \neq -V_{fe}^{-}$. Under these circumstances equation 7.5 becomes

$$|V_{PP}| \exp\left[-(t - t_0)\,\theta\right] = \frac{V_{fe}^{+} - V_{fe}^{-}}{2m_c} \tag{7.7}$$

which causes programming to halt at

$$\phi_{fg} = \phi_{fg}^{T} + \frac{V_{fe}^{+} + V_{fe}^{-}}{2} \tag{7.8}$$

Unfortunately this prevents this scheme being used with *PARAFIN* since *TARDIS* experiments have shown that the programming offset, $\frac{1}{2}\left(V_{fe}^{+} + V_{fe}^{-}\right)$, is randomly distributed, of a magnitude of up to a few volts and constantly changing such that no form of offset cancellation can accommodate it.

A secondary difficulty is the need to load weights in parallel since deselection of cells is not possible (A is globally distributed). Whilst parallel weight downloading is advantageous in terms of speed it presents a problem here since all ϕ_{fg}^{T} values must

[2]Although equation 7.5 can be re-arranged to yield an exact required programming time, programming need not be halted then, since no further tunnelling occurs during subsequent oscillation periods. In practice, it is probably easier simply to let the sinusoid to decay to low levels before halting programming. Note that there will be a slight capacitively-coupled fall in ϕ_{fg} when ϕ_{fg}^{T} is reset after programming (depending in magnitude on m_t.)

be held dynamically at the TUNNEL terminal and area-intensive buffering may be necessary to prevent this node from capacitively following V_{fg}. Dynamic storage of ϕ_{fg}^T must also be able to accommodate with precision the long programming phases (several hundred milliseconds) due to the relatively slow speed of tunnelling.

7.2.2 Threshold Drain Switch

Lanzoni *et al* [97] describe an elegantly simple method of establishing a sense transistor target threshold, V_T^T, as viewed from the CTRL terminal of the circuit shown in figure 7.2.

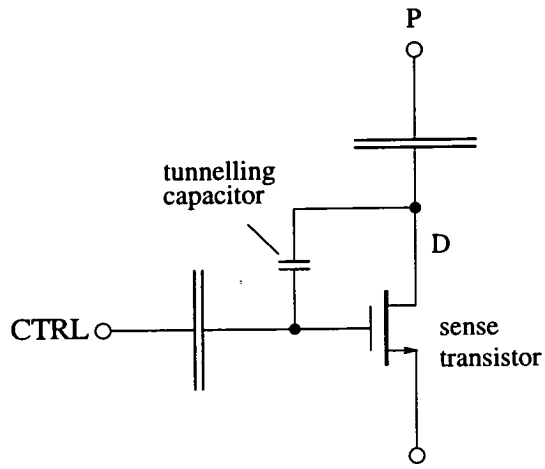


Figure 7.2: Schematic of Lanzoni *et al*'s circuit

The floating gate is preset by applying V_{PP} to CTRL such that electrons are injected onto the floating gate through the tunnelling capacitor from node D (which will be grounded through the sense transistor), long enough until it is certain that the threshold has been *overwritten* ($V_T > V_T^T(max)$)[3] . V_T^T is then applied to CTRL and a voltage ramp applied to node P. Since the sense transistor is off, the voltage on node D follows the ramp by capacitive coupling causing a potential to be established across the tunnelling capacitor such that electron removal by tunnelling begins decreasing V_T. Once $V_T = V_T^T$ is reached, the sense transistor turns on, collapsing the voltage on node D such that tunnelling then halts. Lanzoni *et al* have not fabricated this design but have demonstrated it by Spice simulation. Gotou [69] proposes a modification of this circuit to prevent over-erase in CHE-programmed Flash memories.

[3]Lanzoni *et al* also describe a 4-transistor iterative scheme to increase V_T towards V_T^T so that unnecessary aging due to overwriting may be avoided.

The obvious disadvantage is that the sense transistor must be able to withstand the high programming voltages. This can only be achieved in SLV-CMOS by use of a LDDNMOS device (although, of course, this technique had not been proven at design time). Programming of positive $\phi_{fg} > V_T$ as required by the RBF subcircuits would require programming of negative V_T^T. Since negative voltages cannot be conveniently switched on chip, a high voltage switch would be required to disconnect the high voltage programming ramp connected to P for deselected devices. This, combined with the need for a second large coupling capacitor, could make the cell inconveniently large for a dense analogue data array and matching of LDDNMOS devices may be poor. A more minor problem is the need to program in terms of threshold voltages rather than directly on ϕ_{fg} or, for example, multiplier weights, which fails to exploit the trimming potential of feedback programming of the floating gate to implicitly eliminate some of the offset and mismatch errors.

7.2.3 Continuous-Time Injection

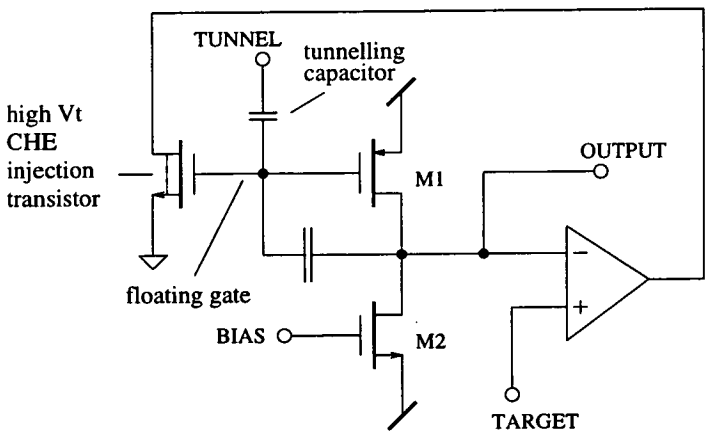


Figure 7.3: Schematic of Diorio *et al*'s circuit

Diorio *et al* [54] propose an alternative approach using CHE. Instead of coupling a high voltage onto the floating gate, ϕ_{fg} is preset high by tunnelling such that the stored charge alone is sufficient to make V_{fg} support injection. Therefore simultaneous read/write is possible and continuous-time feedback programming can be implemented by switching off V_{drain} when a target V_{fg} is reached.

The complete circuit is shown in figure 7.3. A high voltage is applied to the TUNNEL terminal to preset the floating gate with a deficit of electrons (ie. a high ϕ_{fg}). The comparator circuit then drives the drain of the CHE injection transistor sufficiently high to induce CHE; electrons being swept onto the gate by the high ϕ_{fg} which

is gradually diminished. M1 and M2 serve to map the high ϕ_{fg} into a more reasonable circuit voltage which appears at the OUTPUT terminal. When this falls sufficiently to match TARGET, the comparator flips and halts any further electron injection.

The p-implant of a BiCMOS process was used to modify the injection transistor threshold voltage such that even with a sufficiently large V_{fg} to allow injection to occur, the transistor itself is operating in weak-inversion, thus limiting the operating current dissipation. The p-implant is not available in SLV-CMOS but an alternative approach is to use a p-channel injection transistor [55]. Here the injection process is different: a fraction of channel holes may collide with the semiconductor lattice liberating electron-hole pairs; ionised electrons promoted to the conduction band are expelled from the drain by the high electric field, some of which gaining sufficient kinetic energy to surmount the oxide potential barrier. To attract the electrons onto the floating gate, a high gate potential is required which naturally drives the transistor into weak-inversion. The slow rate of this process means large weight changes can take tens of minutes. This is not attractive for the weight-downloading intention of *PARAFIN* but ideal for the gradually adaptive neural system developed by Diorio, Hasler, Minch and Mead (since adaptation should integrate over many presentations of inputs or several periods of the input); in this sense floating gates are not being used as analogue EEPROMs but circuit elements with important time-domain dynamics [53, 55].

All these designs use a single pulse application to instigate programming which is in contrast to pulse-trains with inter-pulse evaluation. Trap-up can be reduced in this way since the peak electric field which occurs on a rising pulse edge occurs only once instead of multiple times within a single programming cycle.

7.3 *PARAFIN* Programming

Having reviewed some alternative approaches to continuous-time analogue programming of floating gates it is possible to assess some desirable qualities for the design of *PARAFIN*:

- Accommodation of tunnelling asymmetry and inter-device mis-match.
- Accommodation of aging due to trap-up to a reasonable extent such that multiple programming cycles are available for CIL trimming before window closure.
- Easy selection/deselection such that individual elements may be trimmed and sequential weight downloading is possible.
- Feedback based on parameterised cell output (eg. weights, centres) rather than explicitly on ϕ_{fg}, eliminating some of the offset and mismatch error.

- Fast weight downloading.
- Limited cell size, with the avoidance of LDDNMOS elements.

The last point requires that feedback circuit elements are distributed between several RBF subcircuits rather than at each individual cell. This results in a *semi-parallel* implementation of weight downloading (eg. by column). Thus the speed of downloading must be traded off between the overall chip size. For *PARAFIN*, a completely serial approach has been taken for simplicity, although this may easily be expanded if desired.

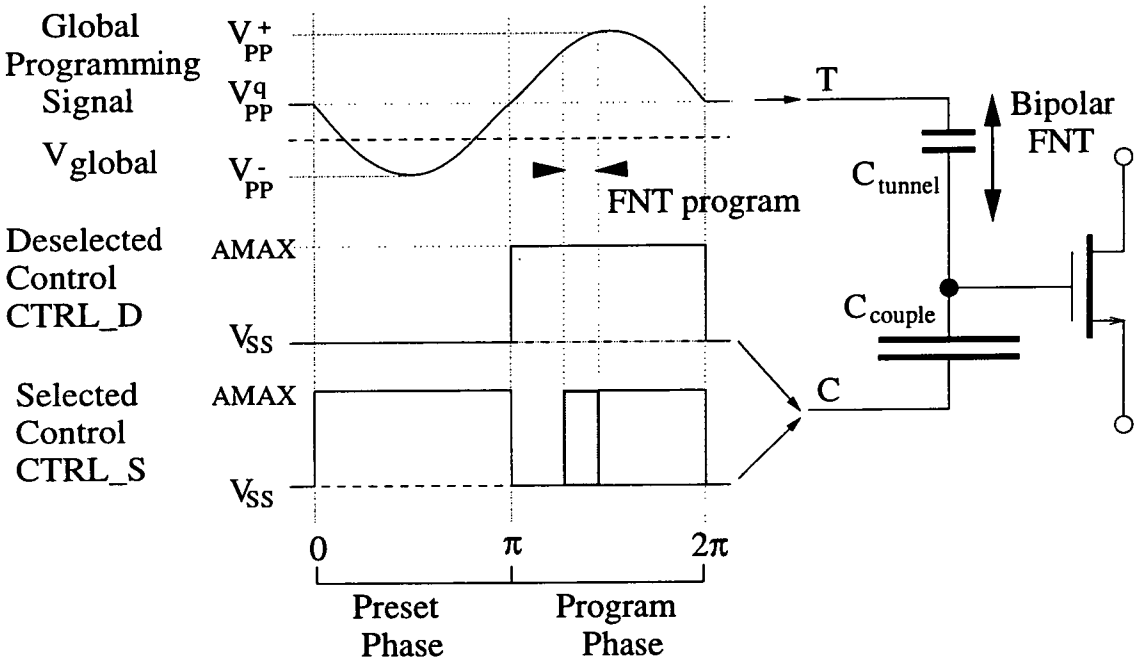


Figure 7.4: Dual-phase FNT continuous-feedback programming scheme signals

Figure 7.4 shows how the control signals of the iterative dual-phase programming scheme are modified for continuous-time feedback. Instead of pulses, a continuous waveform is applied to the global signal line sweeping V_{global} between V_{PP}^{-} and V_{PP}^{+} (here it is shown as an offset sinusoid for the purposes of clarity). Programming is a two stage process:

1. During the *preset* phase $[0 \rightarrow \pi]$, V_{global} sweeps down to V_{PP}^{-}. Control of deselected devices remains in phase with V_{global} and thus remained biased at V_{ss} while selected devices are biased out of phase at $AMAX$. This establishes an electric field in selected tunnelling capacitors only, sufficient that electrons are injected

onto the floating gate, so:

$$\phi_{fg} \rightarrow \phi_{fg}(\pi) \leq \phi_{fg}^{min.} \tag{7.9}$$

where $\phi_{fg}^{min.}$ is the lowest value of ϕ_{fg} which a programmed device may take up and still validly parameterise the RBF subcircuit cell.

2. During the *programming* phase $[\pi \rightarrow 2\pi]$, V_{global} sweeps up to V_{PP}^{+}, with control of deselected devices following to $AMAX$. Control of selected devices remains out of phase and so biased around V_{ss}. Now electrons are removed from selected devices until $\phi_{fg} = \phi_{fg}^{T}$ where ϕ_{fg}^{T} is the target value of ϕ_{fg} such that the parameterised RBF subcircuit cell exhibits the correct output value of weight, centre, etc. Upon reaching this *stopping condition*, the control of the selected cell switches to follow that of the deselected cells. The actual switch point depends on the characteristics of the tunnelling capacitor and ϕ_{fg}^{T}.

During the programming phase, V_{ss} is applied to terminal C as would be the case during read-out, so provided $C_{tunnel} << C_{fg}$ (ie. a very small m_t):

$$\phi_{fg}^{PROGRAM} \approx V_{fg}^{PROGRAM} \approx \phi_{fg}^{READ} \approx V_{fg}^{READ} \tag{7.10}$$

So the behaviour of the parameterised circuit connected to the floating gate can be used to determine the stopping condition for programming. Thus continuous feedback can only be applied to the electron *removal* operation; this is why the preset phase exists - to add surplus electrons to the floating gate if required such that ϕ_{fg}^{T} can always be attained by the removal of electrons. This basic concept is illustrated in figure 7.5 for the case of a multiply/accumulate array.

In practice, obtaining a sufficiently small m_t is not a trivial problem. It did not matter much in $NEMO$ because V_{global} was always zeroed for the interpulse re-evaluations but this is not possible during continuous-time operation. To give an indication of the magnitude of this problem, consider a $NEMO$-sized tunnelling capacitor of 46fF and a floating gate with a large total capacitance of 1pF. Here m_t is 0.046, which means that $V_{PP}^{+} = 24$V would cause V_{fg} to be $\sim$ 1.1V higher than ϕ_{fg} due to capacitive coupling. Correct operation requires that $V_{fg} = \phi_{fg}$ during the programming phase. Capacitive coupling then leads to a large offset (which would vary between devices due to different values of V_{global} at the stopping condition). It may even be sufficient to push the parameterised cell well out its operating range thus producing invalid results. This problem is common to the previously described continuous-time programming circuits in decreasing order of seriousness:

1. Lanzoni *et al* [97]: not only the tunnelling capacitor, but the drain capacitance of the sense transistor are connected to V_{PP}^{+} during programming (measured

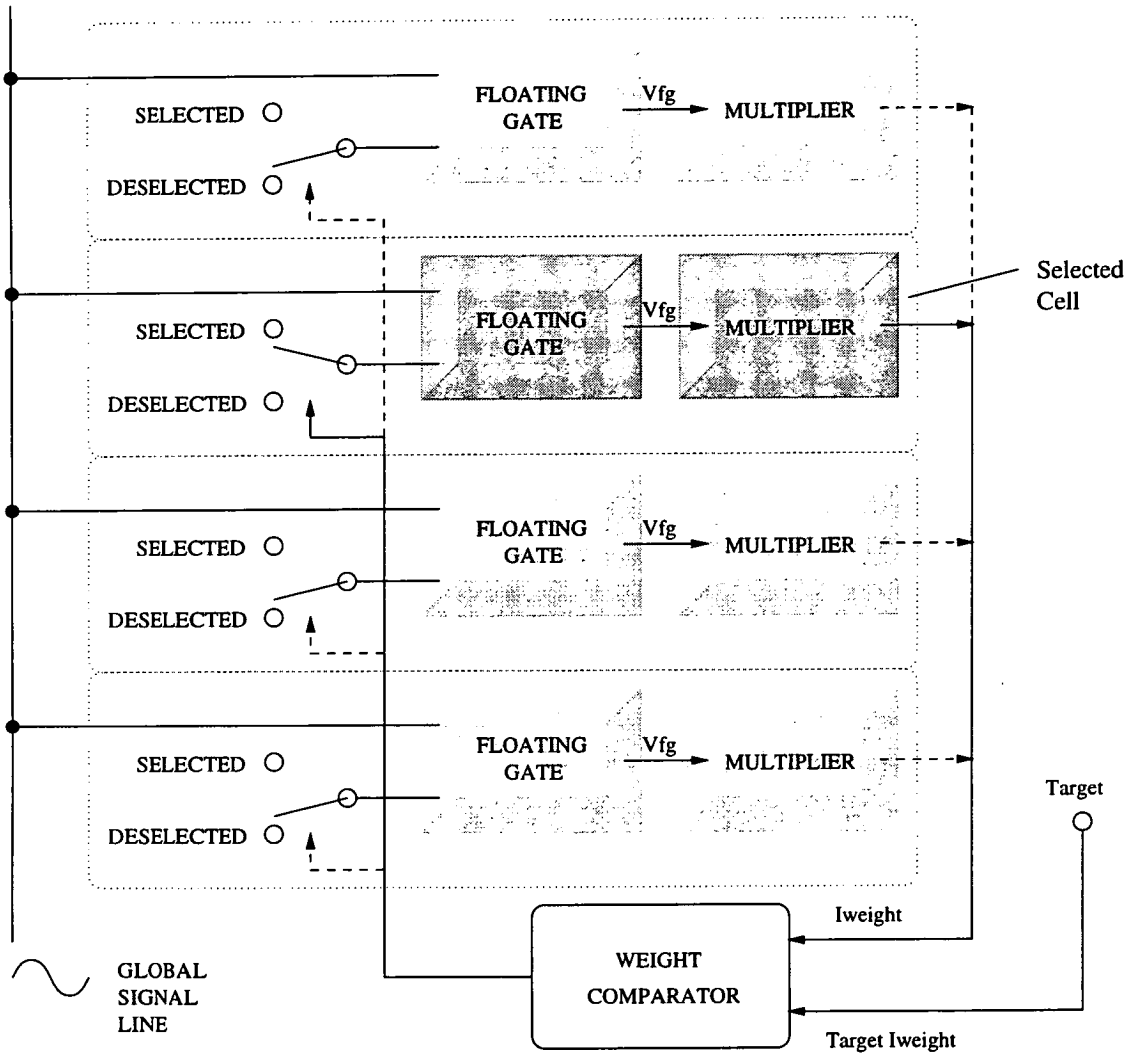


Figure 7.5: Concept of feedback-based dual-phase programming for array of multipliers. The comparator trips when $I_{weight} = I^T_{weight}$, deselecting the selected cell and halting any further programming.

threshold offset $\sim$ 2V).

2. Diorio *et al* [54]: the drain capacitance of the injection transistor is connected to the high drain voltage required for injection, but this capacitance *and* this voltage are smaller than in Lanzoni *et al*'s case.

3. Vittoz *et al* [166]: the tunnelling capacitor is connected to ϕ_{fg}^{T} which would be removed after programming. However this is likely to be a small voltage, smaller than the tunnelling or injection programming voltages and so the effect is even smaller.

A simple solution was taken here: $C_{control}$ was increased from 1pF to 2pF and C_{tunnel} reduced to 3.8fF by minimising the tunnelling capacitor to the form of two crossing minimum polysilicon strips. This reduced m_t to 0.0019, meaning the $V_{PP}^{+} =$ 24V would cause V_{fg} to be $\sim$ 45mV higher than ϕ_{fg} due to capacitive coupling. However, notice also that the $V_{PP}^{+}(min)$ required to program ϕ_{fg} is likely to be in excess of, say, 18V in all cases. This decomposes the offset into a -35mV systematic offset and only -10mV random offset. While this makes the offset tolerable it is not an ideal solution due to (i) the area overhead of $C_{control}$, and (ii) programming a large capacitor through a minimum tunnelling area is both slow and prone to rapid aging due to a concentration of charge transport. Clearly this is a matter which could benefit from further study.

7.4 *PARAFIN* Floorplan

To test the programming scheme introduced in section 7.3, the test chip *PARAFIN* was designed, comprising three test arrays for each RBF subcircuit: Euclidean distance calculator array, multiply/accumulate array and non-linearity array. In addition, a second multiply/accumulate array was included which did not contain the differential stage thus significantly reducing the cell size. These designs will be described in the following sections.

A complete floorplan block diagram for *PARAFIN* is shown in figure 7.6.

7.5 Feedback Programming of Euclidean Array

The *PARAFIN* Euclidean distance calculator cell is shown in figure 7.7[4]. The addressing scheme is the same as for the *NEMO* test array.

[4]The compensation circuit *was* included in the design but is not shown here for clarity. As discussed in chapters 2 and 6 this circuit has been found to be redundant and would be removed in future design iterations.

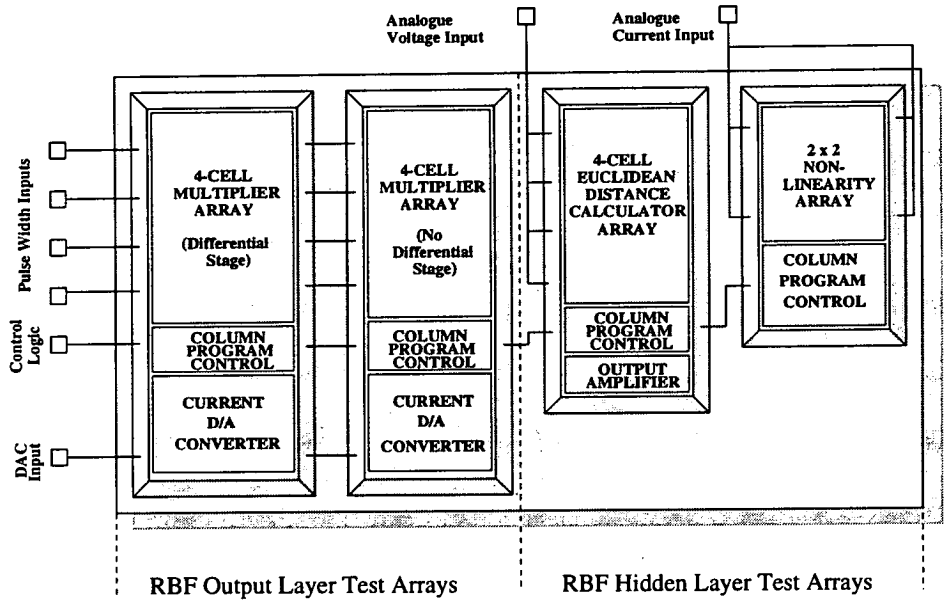


Figure 7.6: *PARAFIN* Block Diagram

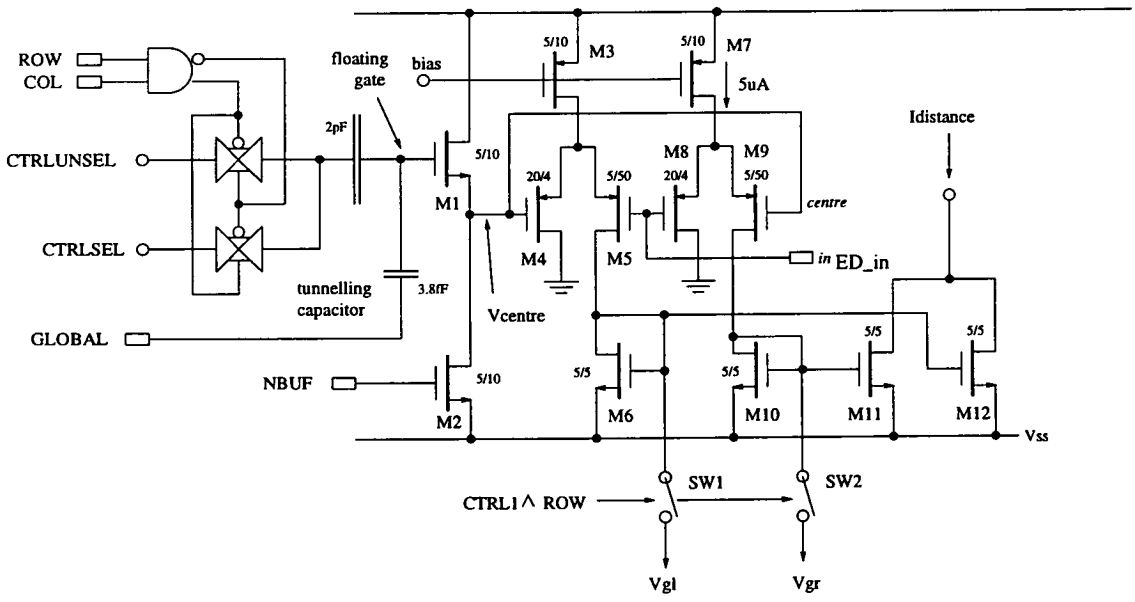


Figure 7.7: *PARAFIN* Euclidean distance calculator cell

Instead of the large floating gate capacitor used in *NEMO*, a smaller capacitor was used with a source-follower to buffer the voltage and so eliminate drain coupling. While this did not significantly reduce the overall cell size and required an additional signal voltage to be generated, the reduction in capacitor meant a reduction in charge to be injected during programming. This should lead to both faster programming and reduced trap-up. With NBUF set at 1V, ϕ_{fg} ranged from 1V to 4V to cover the same centre range as in *NEMO*.

Programming by feedback was based on the currents in the two ratioed pairs. In the preset state, surplus electrons have been injected onto the floating gate such that

$$\phi_{fg}(preset) \leq \phi_{fg}(min) < \phi_{fg}^{T} \tag{7.11}$$

where ϕ_{fg}^{T} is the target ϕ_{fg} required to set up the desired centre location, $centre^{T}$. With the input set to $centre^{T}$, $centre < in$ and current will flow in the left ratioed pair, turning on M6. As $\phi_{fg}(= centre)$ starts to rise with the removal of electrons in the program phase, the current in M6 will start to fall. Ideally, when $centre = in$, both M6 and M10 will be off, and then M10 will turn on as $centre$ continues to rise. As seen in section 6.5.2, the centre may be non-zero but, as figure 6.10 demonstrates, the centre can always be defined as the point where $centre$ and in are equal.

Thus feedback programming is based on deselecting the selected cell when the currents in M6 and M10 become equal. This is detected by a simple current comparator based on stacked cascode current mirrors [62] (high speed operation [131] is not important since FNT is relatively slow); Switches SW1 and SW2 at each selected cell create current mirror connections of M6 and M10 into the comparator. CTRL1 is used to put the array into Write mode by switching the currents of the selected cell into the array current comparator.

A complete schematic of the Euclidean distance calculator feedback programming array is shown in figure 7.8. The circuit may operate as shown in figure 7.4, or, as described here, with the intervention of external logic to prevent unnecessary preset phases (ie. when $centre(0) < centre^{T}$) reducing aging and speeding programming (using low-pass filtered V_{PP} pulses instead of a continuous global signal).

Initially the array is assumed to be in read mode (with inputs $CTRL1 = 0$, $CTRL2 = 1$ and $LATCHCTRL = 0$. The two NOR gates consequently couple 0V onto all floating gates via $CTRLSEL = CTRLUNSEL = 0$). The sequence of programming events is then as follows:

1. Apply the target centre voltage to the input and assert $CTRL1$. This connects the selected cell to the array current comparator, the output of which, DIR, defines the initial state of the cell. If $DIR = 0$, the initial stored centre value ($centre(0) = \phi_{fg}(0)$) is higher than the target centre and preseting is required

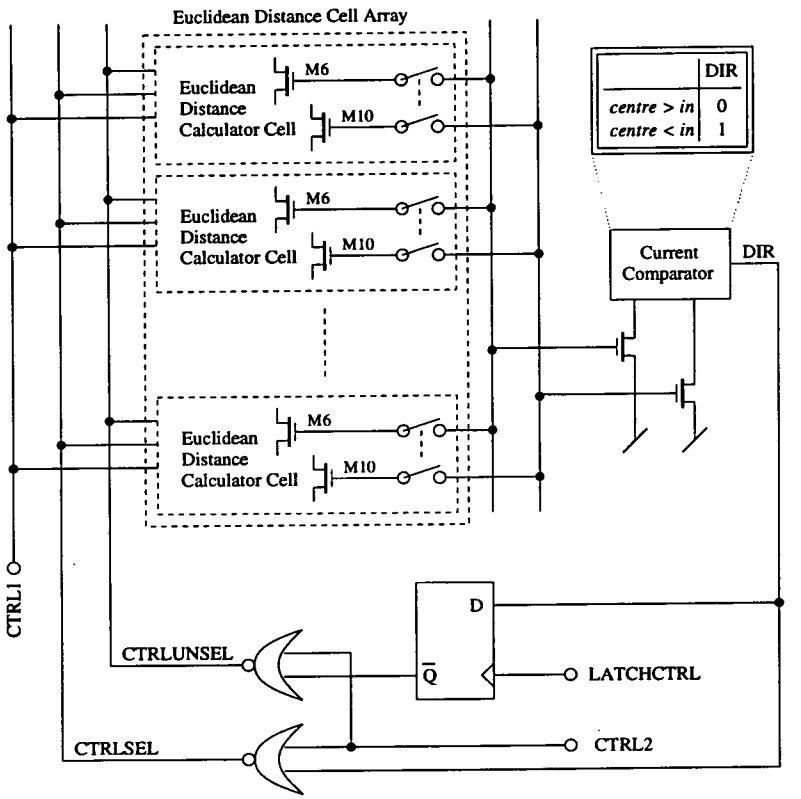


Figure 7.8: *PARAFIN* Euclidean distance calculator feedback programming array

to add surplus electrons (since programming involves the removal of electrons from the floating gate). If $DIR = 1$ the initial stored centre value is already lower than the target centre and no preset is required.

2. If $DIR = 0$ and preset is required, then this is achieved by the following sequence of actions:

   (i). DIR is latched so that when $CTRL2$ is now asserted, $AMAX$ is applied to the $CTRLSEL$ line and coupled onto the *selected* floating gate. Deselected cells are unchanged ($CTRLUNSEL = 0$). The centre value is now even higher due to capacitive coupling and so DIR remains at 0.

   (ii). V_{global} swings low such that electrons are injected onto the selected cell's floating gate. The voltage differential imposed by the coupling of $AMAX$ ensures that negligible injection occurs at deselected cells. Although the stored centre now decreases due to the injection of electrons DIR does not flip since the coupling effect of $AMAX$ is much larger.

   (iii). After a predetermined period $CTRL2$ is deasserted and the cell state (DIR) is re-evaluated to verify a successful preset. If verification fails (DIR still 0, which implies that the stored centre is still higher than the target centre) then it is assumed that trap-up has reduced the number of electrons injected during the preset period such that preset is incomplete. V_{PP}^{-} is therefore reduced by 1V (whilst within predetermined limits to prevent oxide breakdown) and the preset is repeated. The preset is repeated until either successful ($DIR = 1$) or the device is deemed failed due to trap-up (V_{PP}^{-} cannot be reduced without violating oxide breakdown limits).

3. The selected cell may now be programmed since the centre is now less than the target centre ($DIR = 1$). Programming may now be achieved by the removal of electrons. This is accomplished as follows:

   (i). DIR is latched so that when $CTRL2$ is now asserted, $AMAX$ is applied to the $CTRLUNSEL$ line and coupled onto *deselected* floating gates. The selected cell's floating gate remains unchanged ($CTRLSEL = 0$).

   (ii). V_{global} swings high such that electrons are removed from the floating gate. The differential voltage imposed by the coupling of $AMAX$ ensures that negligible removal occurs at the deselected floating gates. $centre$ increases due to the removal of electrons from the selected cell's floating gate. When the stored centre reaches the target centre, DIR flips to 0 causing $CTRLSEL$ to flip to 1 through the NOR gate. This effectively makes the selected cell become deselected and negligible further electron removal occurs. The capacitive coupling of

$AMAX$ onto the selected cell's floating gate will introduce strong hysteresis which prevents instability in DIR.

(iii). After a predetermined period $CTRL2$ is deasserted and the cell state DIR is again re-evaluated to verify successful programming. If the centre has not reached the target ($centre < centre^T - \zeta$, where ζ is a preset tolerance to compensate for capacitive coupling of V_{PP} through the tunnelling capacitor during programming; $\zeta = 100$mV), then V_{PP} is increased by 1V, again to compensate for trap-up and the programming phase is repeated. If programming is unsuccessful and V_{PP} is at the maximum limit, then the device is deemed to have failed due to trap-up.

Based on experience with earlier chips $V_{PP}^{-}(min.)$ was set to $AMAX$ - 27V, and $V_{PP}^{+}(max.)$ was set to 27V.

The following table describes the sequence of events during programming using pseudo-code notion (a $\rightarrow$ indicates a signal transition).

PARAFIN Programming Control Pseudo-Code

	Description	Controls/Inputs	On-Chip Signals
1	Read Mode	$CTRL1 = 0, CTRL2 = 1,$ $LATCHCTRL = 0,$ $V_{global} = 0$	$DIR = X,$ $CTRLSEL = 0,$ $CTRLUNSEL = 0$
2	Select Cell	ADDRESS = {ROW,COL}	
3	Apply Target	$in = centre^T$	
4	Evaluate Cell State	$CTRL1 \rightarrow 1$	$DIR \rightarrow \begin{cases} 0, & \text{if } centre(0) > centre^T \\ 1, & \text{if } centre(0) < centre^T \end{cases}$
5	Preset Required?	If $DIR = 1$ no preset required: go to step 18.	
6	Latch Cell State	$LATCHCTRL \rightarrow 1 \rightarrow 0$	$\overline{Q} \rightarrow 1$
7	Apply Guard Band	$V_{global} \rightarrow V_{PP}^{q}$	
8	Assert Preset Mode	$CTRL2 \rightarrow 0$	$DIR = 0,$ $CTRLSEL \rightarrow AMAX,$ $CTRLUNSEL = 0,$ $centre \rightarrow \gg centre^T$
9	Preset	$V_{global} \rightarrow V_{PP}^{-}$ for duration t_{PP}	$centre$ decreases due to electron injection
10	Apply Guard Band	$V_{global} \rightarrow V_{PP}^{q}$	

PARAFIN Programming Control Pseudo-Code

11	Deassert Preset Mode	$CTRL2 \rightarrow 1$	$CTRLSEL \rightarrow 0$, $CTRLUNSEL = 0$
12	Reset Global Line	$V_{global} \rightarrow 0$	
13	Evaluate Cell State		$DIR \rightarrow \begin{cases} 0, & \text{if } centre(0) > centre^T \\ 1, & \text{if } centre(0) < centre^T \end{cases}$
14	Verify Preset	If $DIR = 1$ preset is complete: go to step 18.	
15	Decrement V_{PP}^- by 1V		
16	Check New V_{PP}^-	If $V_{PP}^- < V_{PP}^-(min.)$ device has failed: HALT.	
17	Repeat Preset	Go to step 6	
18	Latch Cell State	$LATCHCTRL \rightarrow 1 \rightarrow 0$	$\overline{Q} \rightarrow 0$
19	Apply Guard Band	$V_{global} \rightarrow V_{PP}^q$	
20	Assert Program Mode	$CTRL2 \rightarrow 0$	$DIR = 1$, $CTRLSEL = 0$, $CTRLUNSEL \rightarrow AMAX$
21	Program	$V_{global} \rightarrow V_{PP}^+$ for duration t_{PP}	*centre* increases due to electron removal
	Programming Succeeds?		$centre = centre^T$, $DIR \rightarrow 0$, $CTRLSEL \rightarrow AMAX$
22	Apply Guard Band	$V_{global} \rightarrow V_{PP}^q$	
23	Deassert Program Mode	$CTRL2 \rightarrow 0$	$CTRLSEL = 0$, $CTRLUNSEL = 0$
24	Reset Global Line	$V_{global} \rightarrow 0$	
25	Adjust $centre^T$ for Capacitive Coupling	$in = centre^T - \zeta$	
26	Evaluate Cell State		$DIR = 1$, $CTRLSEL = 0$, $CTRLUNSEL \rightarrow AMAX$
27	Verify Programming	If $DIR = 0$ programming complete: go to step 32.	

PARAFIN Programming Control Pseudo-Code

28	Reset $centre^T$	$in = centre^T$	
29	Increment V_{PP}^+ by 1V		
30	Check New V_{PP}^+	If $V_{PP}^+ > V_{PP}^+(max.)$ device has failed: HALT	
31	Repeat Programming	Go to step 18	
32	Reset Read Mode	$CTRL1 \rightarrow 0$	
33	Programming Complete	HALT	

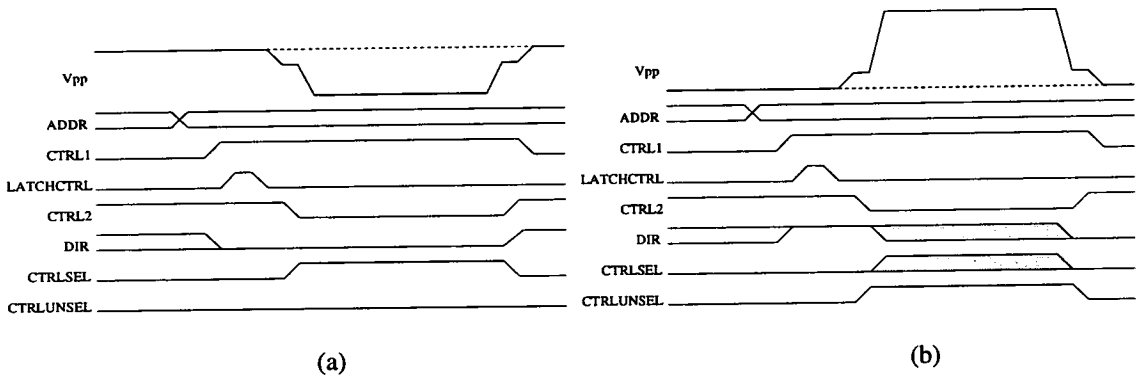


Figure 7.9: Signal timing diagrams for the *PARAFIN* Euclidean distance array. (a) Preset Phase (b) Program Phase.

Figure 7.9 shows timing diagrams corresponding to the scheme described. The shaded area in figure 7.9(b) represents the indeterminate transition time corresponding to the flipping of DIR when $centre$ reaches $centre^T$. This specific point depends on $centre^T$ and the characteristics of the tunnelling capacitor in use.

7.5.1 Experimental Results

Figure 7.10 shows measured traces from the Euclidean distance cell array with centres programmed using the feedback circuit, demonstrating basic functionality. Figure 7.11 shows CRO traces measured during the programming cycle showing the control signals being generated on the board and input to the chip, and the buffered DIR response of the array.

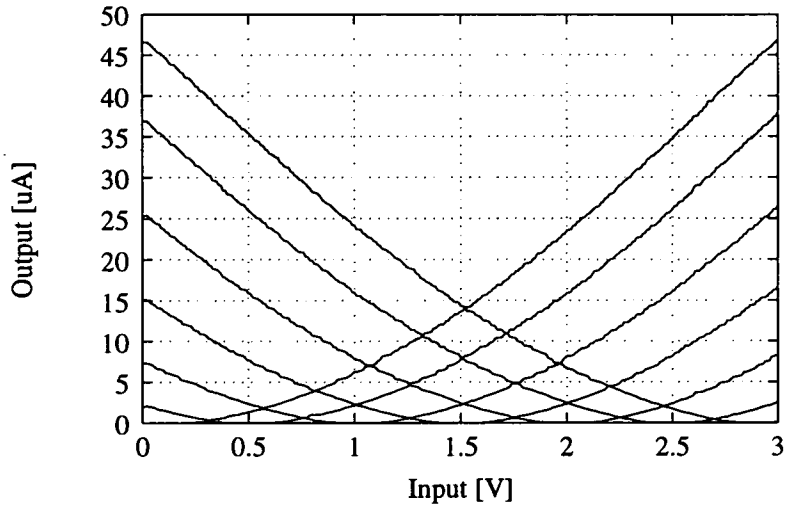


Figure 7.10: Measured traces of Euclidean distance cell programmed to 0,0.5,1,1.5,2,2.5 and 3V centres.

Disturb Protection of Deselected Cells

Before examining the programming performance in detail, the disturb protection of deselected cells was measured since this was an issue of concern with the Dual-Phase arrays on *NEMO*. Cell 1 was continually reprogrammed to pseudo-random 8-bit targets (equation 6.7). Cells 2,3 and 4 were initially programmed once to cover the 0-3V centre range and then not programmed again. Figure 7.12(a) shows how the programming of these cells decays as the programming trials of Cell 1 proceed. A strong correlation between sharp decays in the centre values and increments in V_{PP}^{+} (figure 7.12(b)) is apparent and shown boxed. This indicates a more severe disturb degradation than was expected from *NEMO* results.

This behaviour may be traced to back to the reduction in size of the tunnelling capacitor to reduce m_t. Both the surface area and perimeter have been greatly reduced from the values used in *TARDIS*. In fact the overlap perimeter is as short as that of a *TARDIS underlapped* tunnelling capacitor. It is therefore believed that the number of sharp asperities has been greatly reduced, resulting in the higher V_{PP}^{+} and V_{PP}^{-} magnitudes which are observed to induce tunnelling (higher than expected to support the $\sim$ 1V step in $\phi_{fg}(range)$ to accommodate the source-follower). Referring again to figure 5.15, this higher bulk electric field increases the range of 'bluntness' over which tunnelling currents have a reduced electric field dependency and so greater disturb current may be expected due to the larger number of blunter asperities.

Here it was found that disturb protection increased with increasing $AMAX$, rather than peaking at $AMAX = 10$V as found on *NEMO*. Therefore the maximum

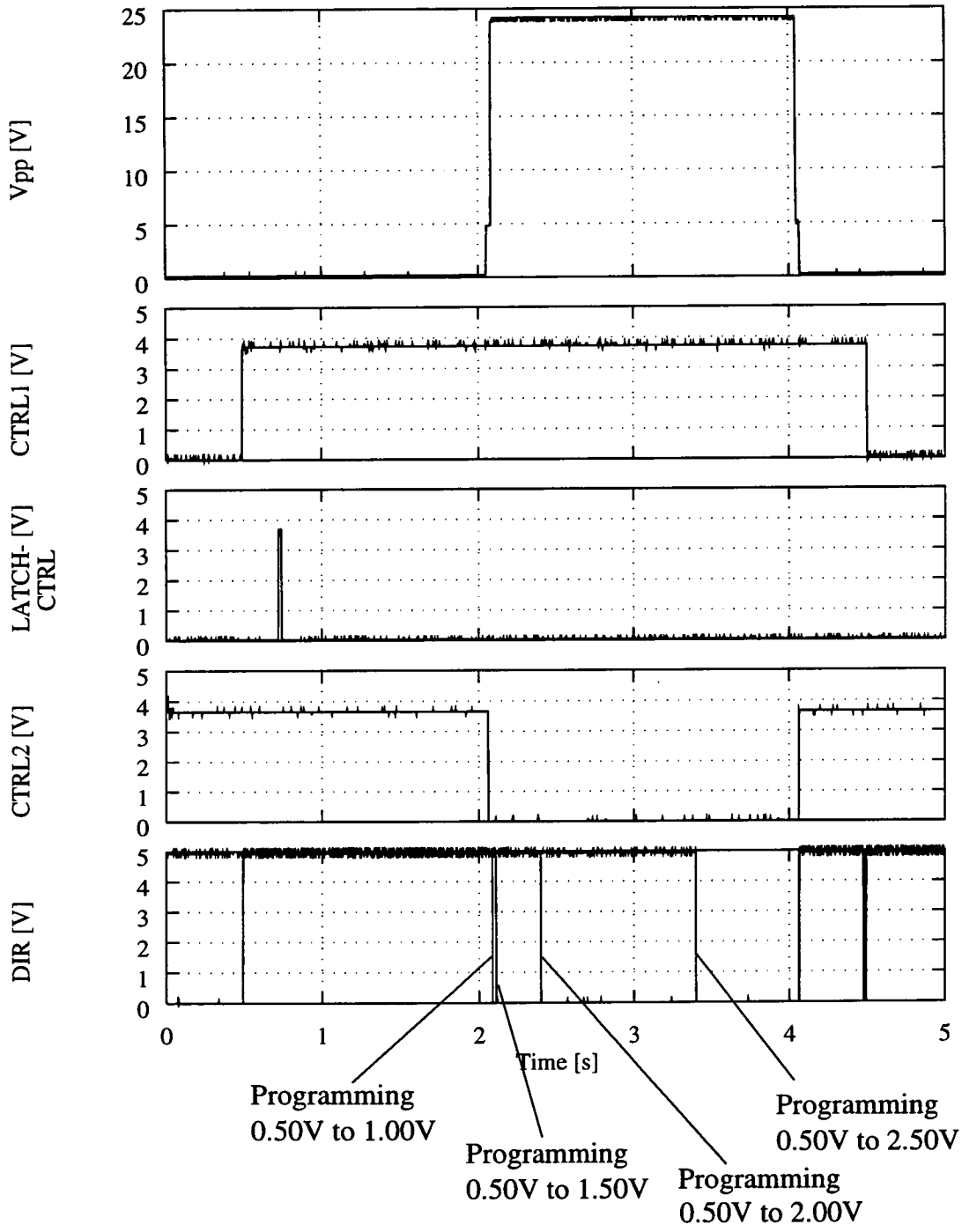


Figure 7.11: Measured CRO traces of control signals and DIR responses during four (overlaid) programming cycles.

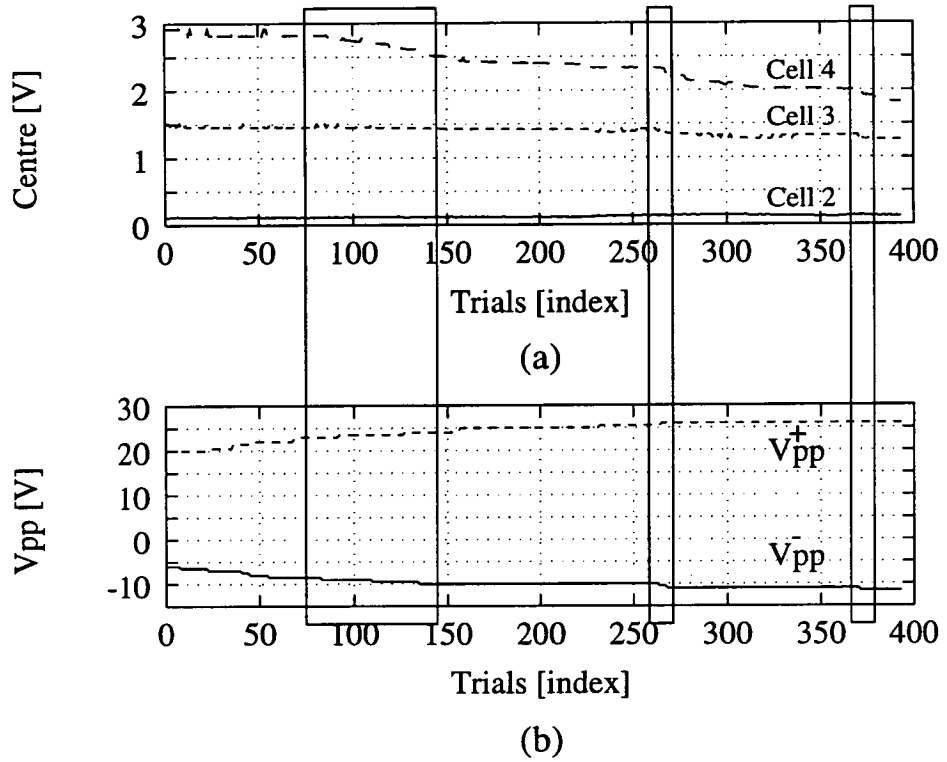


Figure 7.12: (a) Measurement of 3 deselected cells during repeated reprogramming of selected Cell 1, (b) V_{PP} incrementation during repeated reprogramming of selected Cell 1. $AMAX = 12V$.

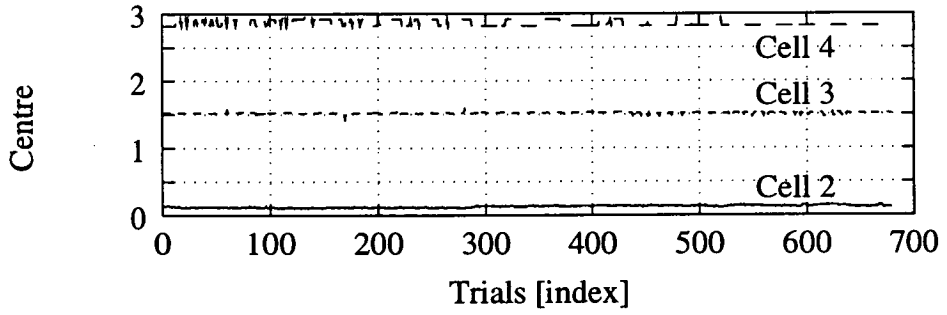


Figure 7.13: Measurement of 3 cells during repeated reprogramming of all cells, Cell 1 to a random target, Cell 2 to 0.15V, Cell 3 to 1.50V and Cell 4 to 2.85V. $AMAX = 12V$.

$AMAX$, 12V, was best for this array. The change from a 3 p-channel to a single n-channel sense transistor is probably important in reducing parasitic gate current. Not only is the injection area reduced but any contribution due to SHE is likely to be diminished due to the lower substrate doping [126].

Thus the only two possible ways to reduce the disturb error were to

1. Always use $AMAX = 12V$.

2. Reduce t_{PP} from 2 seconds. t_{PP} had originally been set so long to compensate for the increased $|V_{fe}^{\pm}|$ of the minimal $PARAFIN$ tunnelling capacitors. However, in the majority of programmings, programming could be achieved *during* the rise time of V_{PP}^{+}, the reset of the 2 seconds contributing only to parasitic reprogramming currents. t_{PP} was therefore reduced to 10ms. The control PC was used to permit a second pulse of 100ms and a third of 2s if necessary (verification failure) before increasing $|V_{PP}^{+}|$, to accommodate the small number of programmings which required long programming times. In practice these were large $centre$ swings across the entire weight range, unlikely to occur often in CIL.

Of course, a simple method to eliminate the disturb error is to re-program all cells in the array at each trial iteration. This is demonstrated in figure 7.13 which shows much improved stability of the 'unchanging' cells. Of course, the disturb error gives an ultimate bound on the acceptable size of a single array, which from figure 7.12(a) is a few 10s of elements. This is not a particularly severe restriction since it is already desirable to partition the chip into small arrays or neural columns which can be programmed in parallel.

Programming Endurance and Accuracy

To determine the programming endurance and accuracy of the feedback scheme, 4 cells of an array were continuously programmed. An error function, $Error = centre - centre^{T}$, was used to monitor performance. As can be seen in figure 7.14, about 500-1000 programming trials could be reasonably completed (depending on the cell) before the error increased substantially due to the inability of an increase in V_{PP} to accommodate the effects of trap-up without causing oxide breakdown (the device is then considered a failure). Since the error is negative, this suggests that V_{PP}^{+} is the limiting voltage in this case as $centre$ continually undershoots $centre^{T}$. Although random programming is not worst-case[5] it is worse than would be expected for CIL, where small changes in $centre$ values would be more likely.

[5]Worst-case programming would involve programming *centre* in the sequence 0V,3V,0V,3V,... to ensure maximum charge transport through the tunnelling oxide.

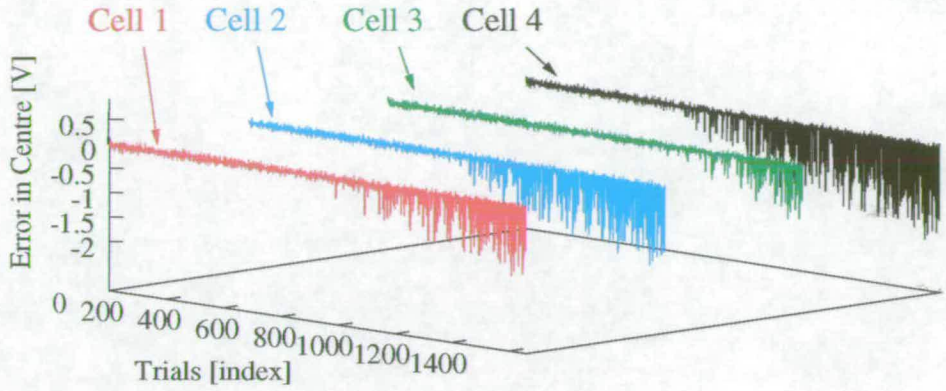


Figure 7.14: Programming error in 4 randomly programmed cells over a contiguous series of trials.

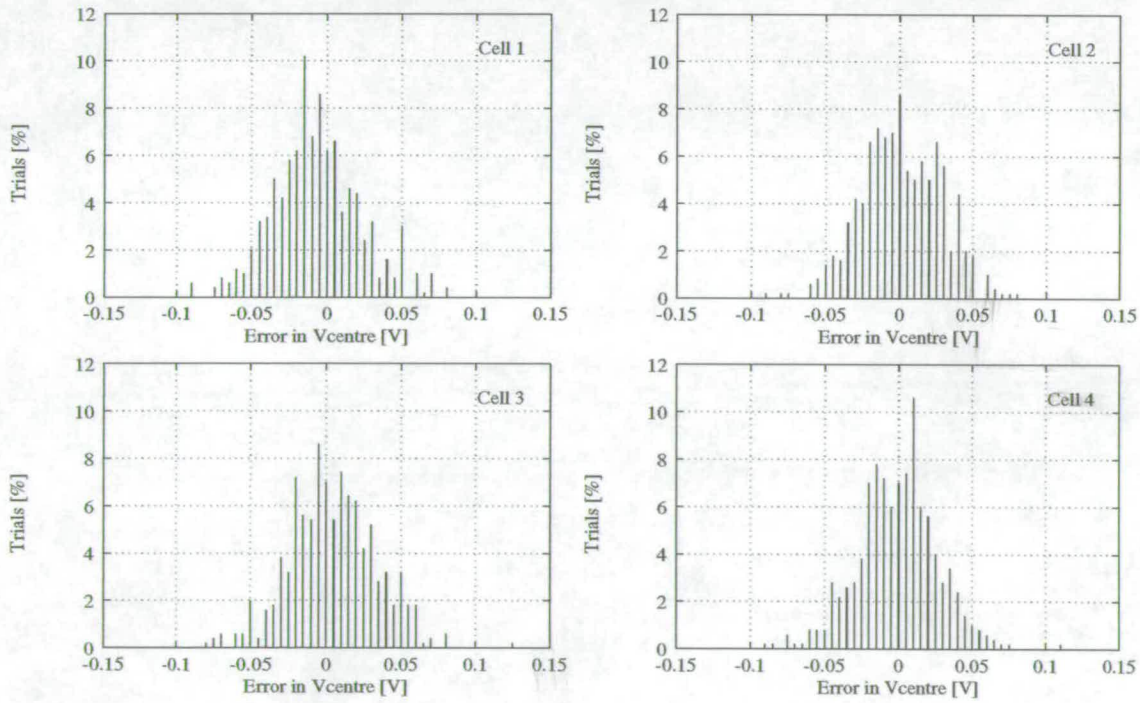


Figure 7.15: Histograms of the programming error for 4 Euclidean distance cells in an array over first 500 reprogramming trials.

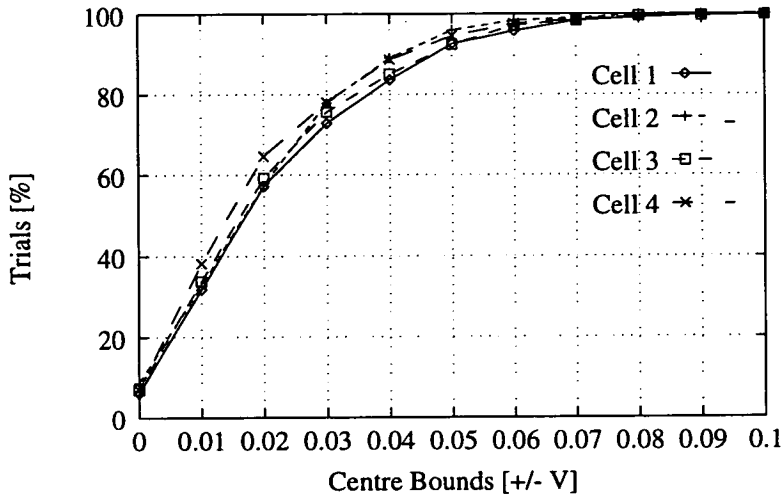


Figure 7.16: Achieved bounds around $centre^T$ obtained by programming 4 Euclidean distance cells in an array over first 500 reprogramming trials.

Figure 7.15 shows the distribution of errors for the first 500 trials (ie. before any cell fails). There are slight systematic offsets but the achieved *centres* are reasonably equally distributed about the $centre^T$s. Figure 7.16 shows how these distributions map into the probability of programming within specific bounds, showing greater than 5 bit equivalence ($\xi < \pm 0.0484$) in more than 90% of trials.

It was found however, that the distributions shown in figure 7.15 masked a data dependency in the mean of the error distributions. This is highlighted in figure 7.17 which shows low $centre^T$ values tend to overshoot while high $centre^T$ values tend to undershoot. Since this behaviour is the opposite of what would be expected due to capacitive coupling through the tunnelling capacitor, it is more likely due to inherent asymmetry within the circuit.

A significant source of mis-match may be cross-chip variation in the two arms of the switched current mirrors which feed the current comparator since these 'matched' transistors may be some physical distance apart. It may be interesting to explore a voltage-mode design which explicitly compares the *centre* generated by the source-follower with $centre^T$ since this this removes the transistor matching requirement of the current mirror approach.

7.6 Feedback Programming of Multiplier Array

Like the Euclidean distance calculator array, the multiplier array on *PARAFIN* is similar to its predecessor on *NEMO*. However, as mentioned in section 7.4, there is also a second multiplier array which does not have the differential stage between the

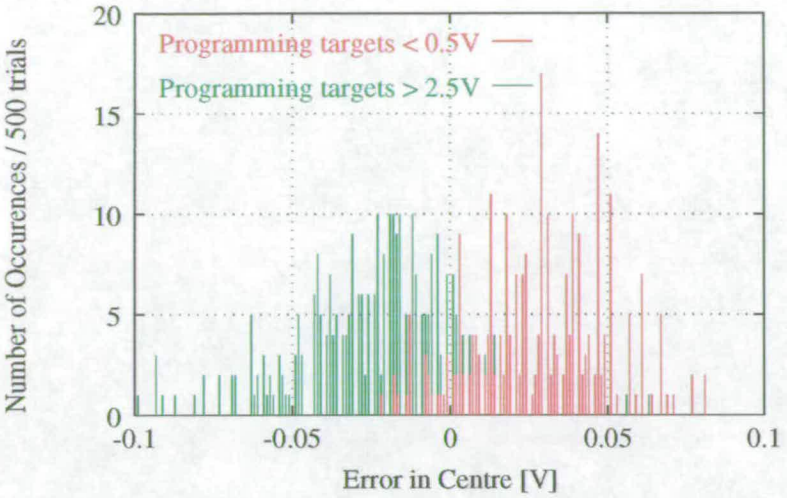


Figure 7.17: Distribution of programming error for targets at the two extremes of the programming range (all four cells).

floating gate and the PWM multiplication circuit. Since a linearized mapping between ϕ_{fg} and I_{weight} is no longer important, the case for such a large component in each cell is considerably weaker. Therefore an array both with (SSYN array) and without this stage (DSYN array) were included to test whether its removal was acceptable.

Figure 7.18 shows the two multiplier cells. Here, the currents through transistors M6, I_{zero}, and M7, I_{set}, Kirchoff sum at node ①, yielding a 2-quadrant I_{weight} current through transistor M11 performing a 2-quadrant multiplication with PWM input τ_i as before. M10 is used to connect node ① to `INIT_VOLTS` when both M8 and M11 are off to eliminate the charge-injection induced offset about zero observed with the *NEMO* multipliers. $\hat{\tau}_i$ is generated on-chip:

$$\hat{\tau}_i = \neg\left(\tau_i \vee CTRL1\right) \tag{7.12}$$

To set I_{weight}, it must be copied from the cell into the array current comparator through transistor switch M8. However, to simplify the comparator, a single quadrant weight current is required. Therefore a copy of I_{zero} is Kirchoff added using transistor M9, matched to M6. This yields a cell output current of

$$I^*_{weight} = I_{weight} + I_{zero} = 2I_{zero} - I_{set} \tag{7.13}$$

which is always non-negative.

As seen in figure 7.19, the feedback control circuitry is very similar to that used in the Euclidean distance calculator array. However, in the Euclidean array $centre^T$ was presented as a voltage to the cell input (using an external 8-bit voltage digital-to-

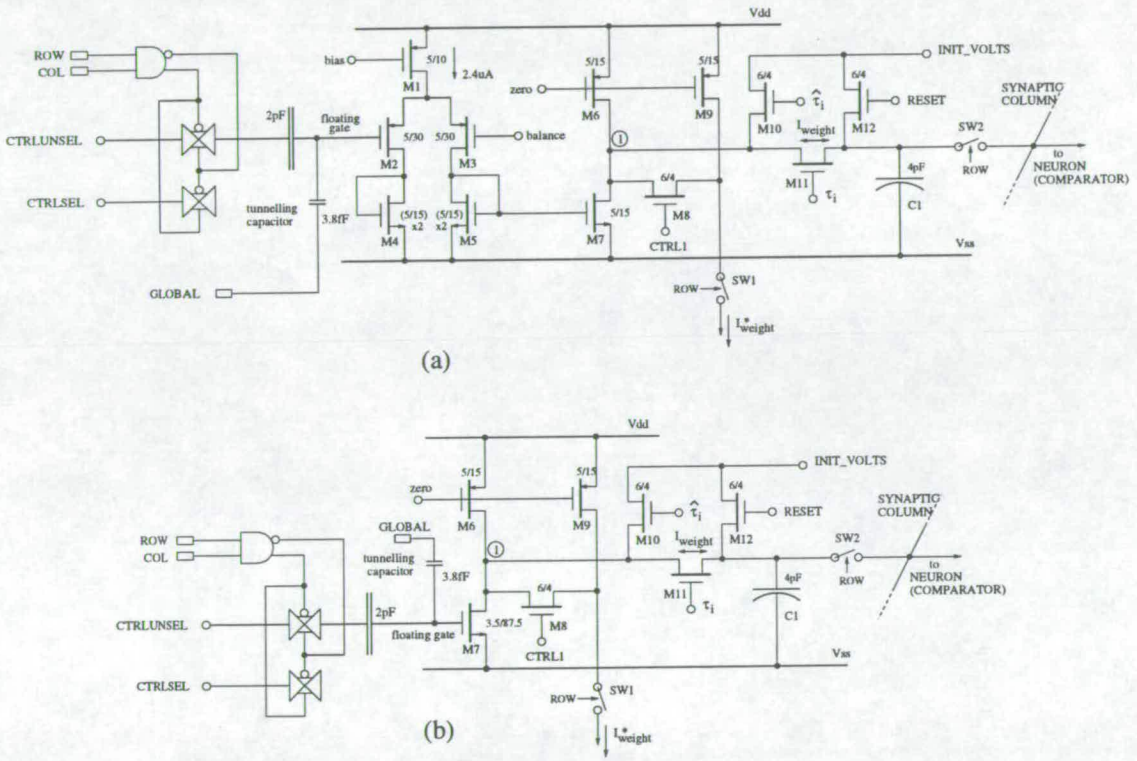


Figure 7.18: (a) Multiplier circuit with differential stage, (b) Multiplier circuit without differential stage

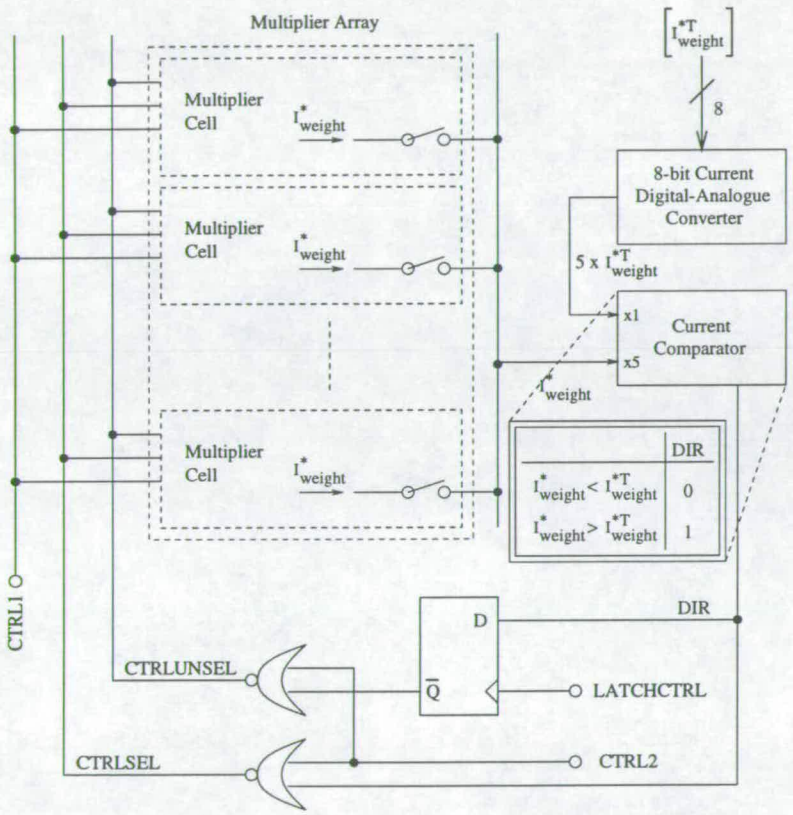


Figure 7.19: *PARAFIN* Multiplier feedback programming array

analogue converter). Here, I^{*T}_{weight} is a current which is more conveniently developed on chip using a current digital-to-analogue converter. Such a component had already been designed by Mayes [110] for use in dynamic current mirror multipliers, and the layout was used here.

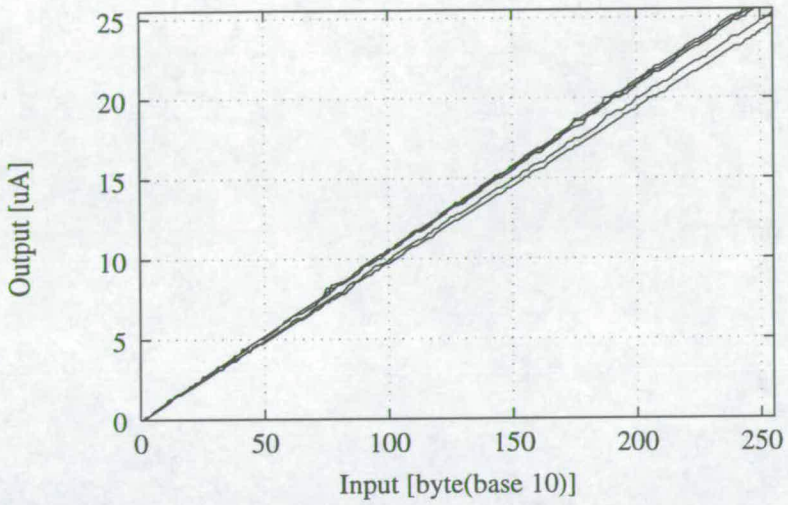


Figure 7.20: Measured current DAC characteristics for five chips

Figure 7.20 shows some characterisation results for the fabricated current DAC on 5 chips biased to perform the function

$$I_{out} = \frac{\text{Input Data}}{10}\mu\text{A} \tag{7.14}$$

There is a slight gain variation between the chips due to mis-match which is similar to that reported in [110], but the operation was considered adequate for testing the multiplication arrays. Note that I_{out} is amplified by ratioed current mirrors for extraction off-chip. The on-chip range was $0 \rightarrow 6$uA; I^{*}_{weight} was multiplied five times by ratioed current mirrors at the input to the current comparator for an equivalent range.

The sequence of events during programming is similar to those for the Euclidean distance calculator array, although, as for $NEMO$, V^{+}_{PP} application results in a decreasing ω compared to an increasing *centre* in the Euclidean array. In this case final verification fails if $\omega > \omega^{T} + \zeta \equiv < weight > + < zeta >$ where $< zeta >$ is 9 for the SSYN array and 15 for the DSYN array. The larger $< zeta >$ requirement in the DSYN array is because of the greater transconductance of Mset compared to the entire differential stage in the SSYN array. In the SSYN array $\phi(range) = 0 \rightarrow 3$V, but in the DSYN array $\phi(range) = 0 \rightarrow\sim 2$V. Therefore capacitive coupling of V_{global} has a proportionately larger influence on ω.

7.6.1 Experimental Results

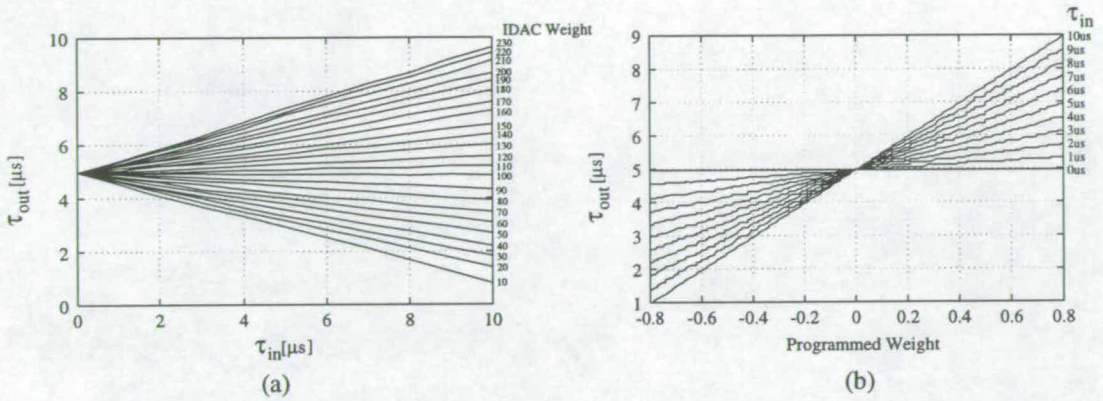


Figure 7.21: Measured multiplication characteristics (a) τ_{out} versus τ_{in} for various programmed weights (τ_{in} ramped from 0 to 10μs in 40ns increments), and (b) τ_{out} versus the programmed weight range for various τ_{in} (ω programmed from -0.8 to 0.8 using 1 bit increments of the digital latch data).

Figure 7.21 shows measured traces from the DSYN array with weights programmed using the feedback circuit, demonstrating basic functionality. Both SSYN and DSYN arrays exhibited equivalent multiplication characteristics, both with the 'kick' about zero seen on *NEMO* (figure 6.13) eliminated.

Weight Mapping Characteristics

In contrast to the Euclidean distance calculator array where V^T_{centre} is applied directly, ω^T is applied indirectly in the form of a current, I^{*T}_{weight}, which in turn is represented as the digital input to the on-chip current D/A converter, `<weight>`^{T}. downloading to function correctly, a simple linear mapping is expected between `<weight>` and ω which should be ensured both by the linearity of the D/A converter, the Kirchoff summation in equation 7.13 and the programmable range of the multiplier[6].

Figure 7.22 shows some measured results for both the SSYN and DSYN multiplier arrays. Both arrays showed reasonably good linearity over the programmable weight range, although in some cases the differential stage bias current (SSYN array) or the zero current bias (DSYN array) had to be altered by up to 1μA from its designed value to trim the mapping offset to the correct range. However this trimming refers to

[6]Clearly, since feedback is based on the value of I^*_{weight} rather than ϕ_{fg}, in the SSYN array the linearity of the differential stage V_{fg} to I(Mset) mapping does *not* impact upon the linearity of the `<weight>` to ω mapping other than to bound the possible range of I^*_{weight}.

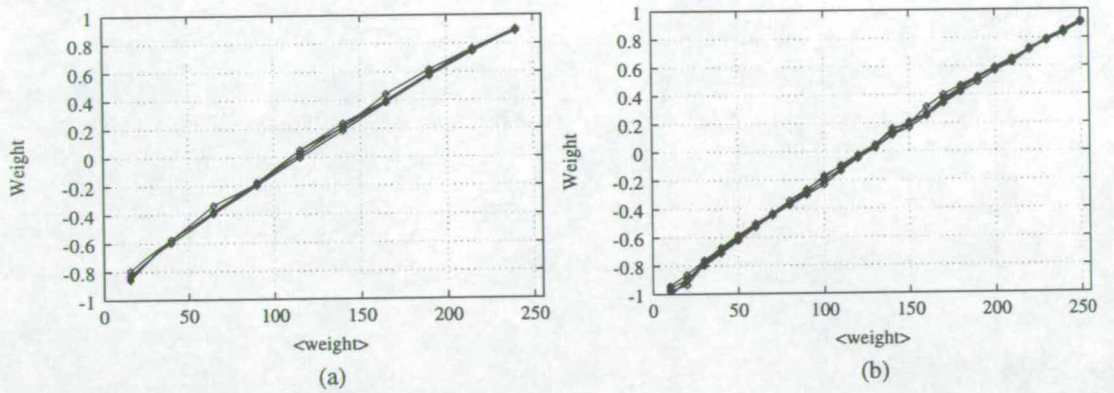


Figure 7.22: (a) Typical measured digital <weight> to ω mapping in SSYN array, (b) Typical measured digital <weight> to ω mapping in DSYN array.

chip-to-chip variations, and intra-array mapping reproducability was good in all chips tested. An approximate mapping function could therefore be introduced:

$$\omega = 2\left(\frac{< weight >}{255_{10}}\right) - 1 \tag{7.15}$$

Which may be conveniently re-arranged to determine the format for weight downloading:

$$< weight >= 127.5\,(\omega + 1) \tag{7.16}$$

(rounded to the nearest integer).

Disturb Protection of Deselected Cells

Figure 7.23 shows the disturb protection performance of the tests arrays while cell 1 is programmed to a different randomly selected target at each trial. The use of reduced t_{PP} shows a much improved disturb protection in figures 7.23(a) and (c) over the Euclidean distance calculator array, with hundreds of reprogrammings of cell 1 possible for negligible degradation in the programmed value of cells 2-4.

Figure 7.23(b) and (d) demonstrates repeated reprogramming of cell 1 in conjunction with repeated reprogramming of the same initial values for cells 2-4. Experiments terminated when cell 1 could not be properly reprogrammed due to trap-up failure.

The original weight set of cells 2-4 was then preserved over hundreds of cycles in the SSYN case, and thousands in the DSYN case. The reason for the longer DSYN operation is due to the reduced $\phi(range)$. This reduces the maximum V_{PP}^{+} required during programming and so permits many more cycles (due to logarithmic nature of

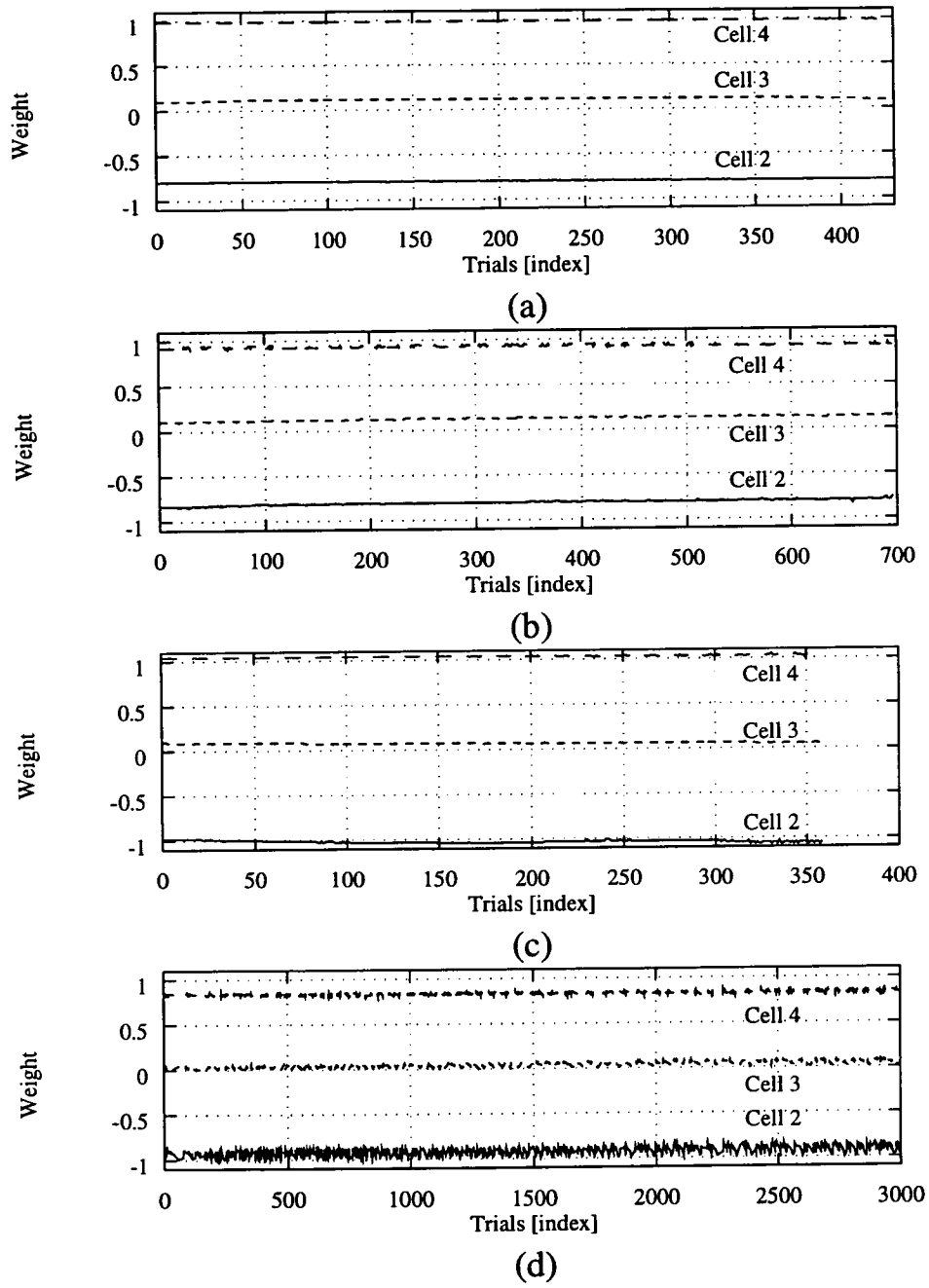


Figure 7.23: (a) SSYN Array disturb performance: deselected cells 2-4 programmed once at trial 0, (b) SSYN Array disturb performance: all cells programmed at each trial, cells 2-4 programmed to the same value each trial, (c) DSYN Array disturb performance: deselected cells 2-4 programmed once at trial 0, (d) DSYN Array disturb performance: all cells programmed at each trial, cells 2-4 programmed to the same value each trial.

trap-up) before it reaches the maximum limit. However, as can be seen in figure 7.23(d) the reduced range impacts on the precision of repeated reprogrammings.

Programming Endurance and Accuracy

As for the Euclidean distance array, the cells of a SSYN and DSYN multiplier array were randomly programmed across the entire weight range to determine endurance and programming accuracy. As figure 7.24(a) shows, the SSYN cells could sustain about 900-1300 programming trials before the weight error became unreasonable. This endurance behaviour is better than for the Euclidean distance array cells since although the same $\phi_{fg}(range)$ is used, it is 0-3V as opposed to 1-4V and so a lower maximum V_{PP}^{+} is required to overcome the stored charge.

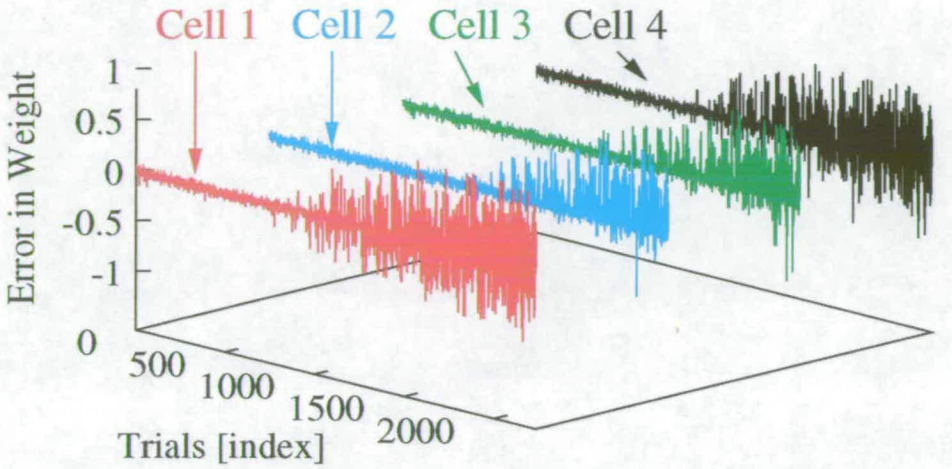
Figure 7.24(b) demonstrates that the reduced $\phi(range)$ of the DSYN array sustains a much higher endurance of upwards of 4000 cycles (approximately an order of magnitude better than for the SSYN array). Again this is due to a smaller $\phi_{fg}(max)$ resulting in a smaller maximum V_{PP}^{+} since the valid ϕ_{fg} range is approximately only 0-2V.

Figure 7.25 shows the error distributions over the programming trials for both arrays. As expected, the SSYN array shows a tighter distribution than DSYN due to the larger $\phi(range)$. Both arrays exhibit a positive offset tendency due to capacitive coupling. This effect was hidden in the Euclidean array due to data dependency of the distribution, but no such dependency was found for the multiplier arrays. Figure 7.26 shows how these distributions map into the probability of programming within specific bounds, showing greater than 4 bit equivalence ($\xi < \pm 0.0588$) in more than 90% of trials for both arrays. Due to its tighter distribution the SSYN array approaches 5 bits equivalence.

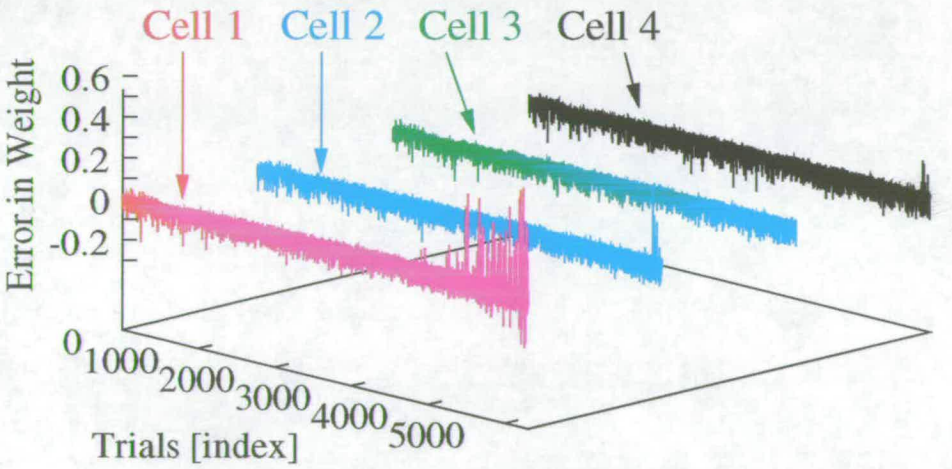
Mapping Endurance

Since $|V_{PP}^{+}|$ is allowed to grow within fixed bounds to accommodate trap-up, the offset due to capacitive coupling through the tunnelling capacitor will change as the device ages. This may have some impact on the mapping characteristics.

However, as demonstrated in figure 7.27, since m_t has been minimised then the actual shift in the mapping is sufficiently negligible to be ignored. In the case shown, after 5200 cycles, cell 1 of the array has been aged more than the other three cells and cannot be programmed across the entire $\omega(range)$.



(a)



(b)

Figure 7.24: (a) Programming error in 4 randomly programmed SSYN array cells over a contiguous series of trials, (b) Programming error in 4 randomly programmed DSYN array cells over a contiguous series of trials.

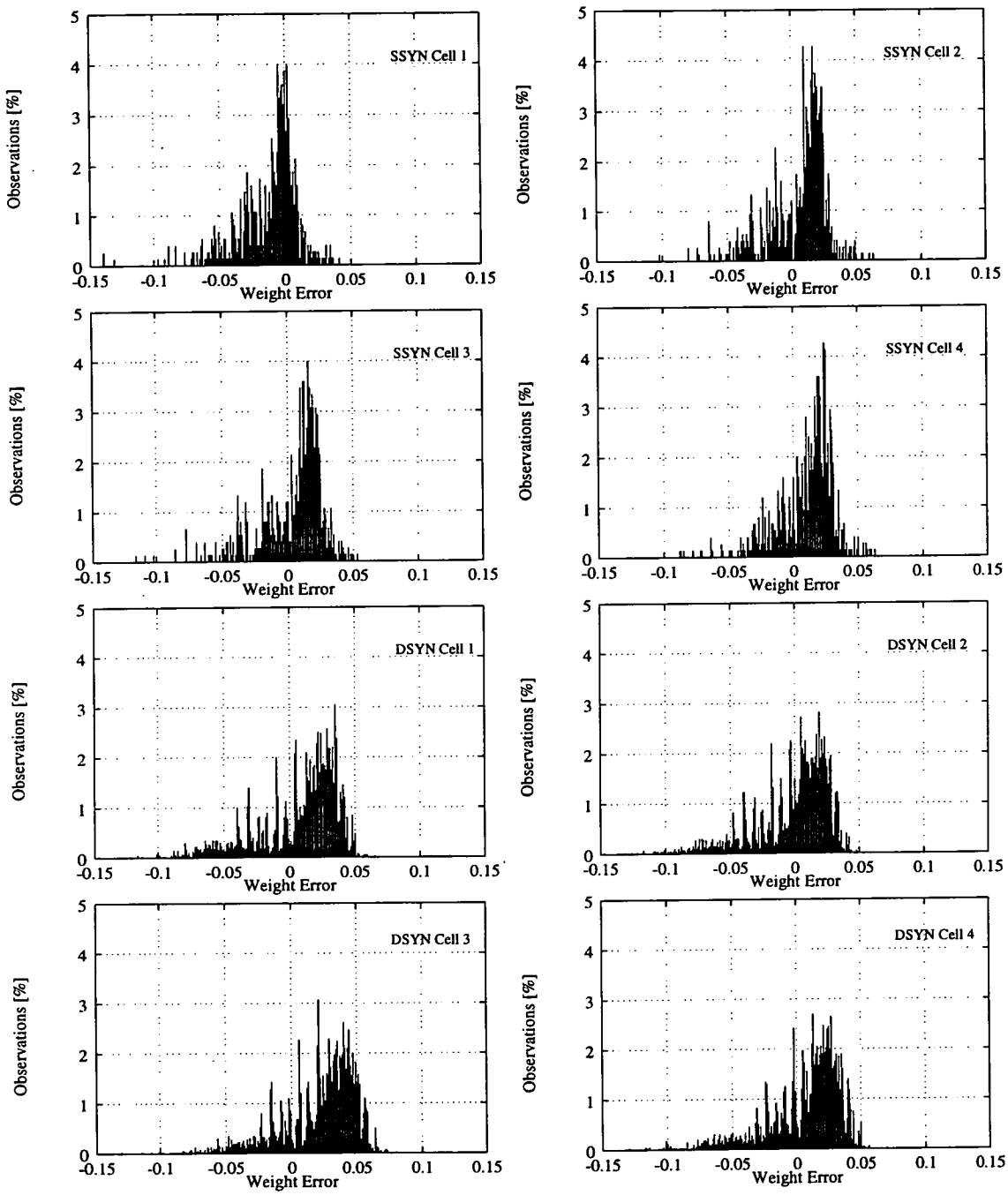


Figure 7.25: Histograms of the programming error for 4 SSYN multiplier cells and 4 DSYN multiplier cells in an array. The SSYN results are over 350 trials, the DSYN results are over 4000 trials.

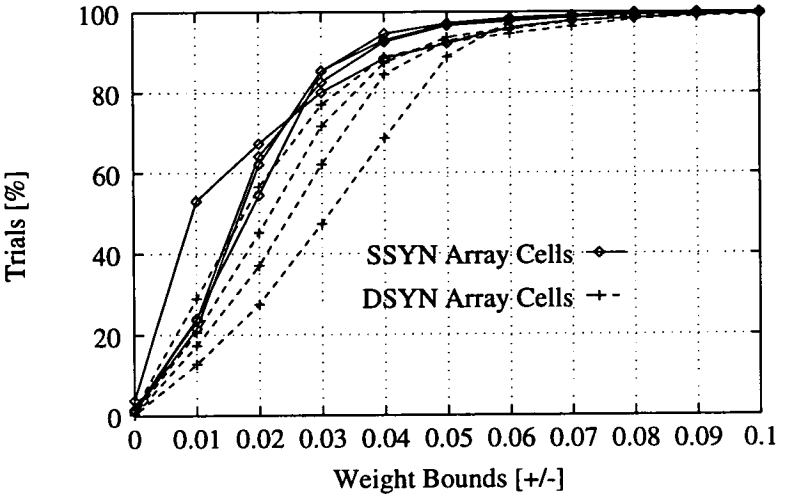


Figure 7.26: Achieved bounds around ω^T obtained by programming 4 SSYN cells and 4 DSYN cells in an array (350 trials for SSYN array, 4000 trials for DSYN array).

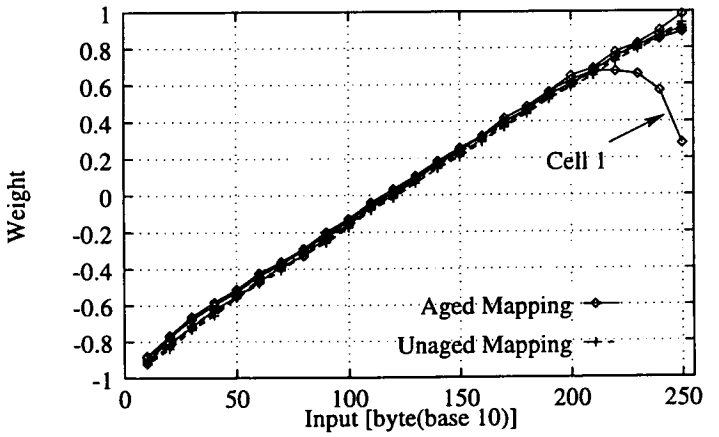
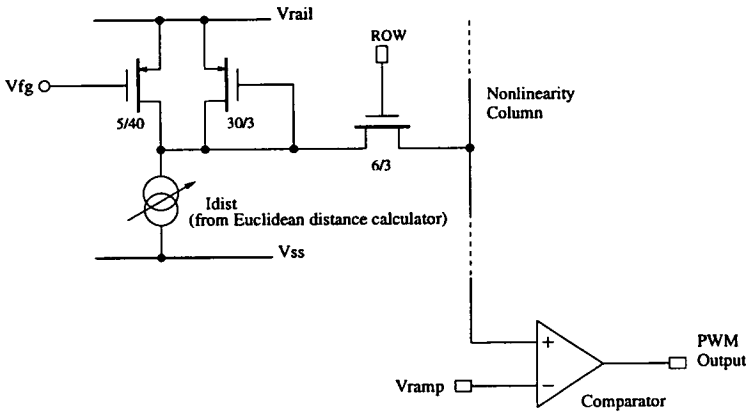


Figure 7.27: Typical measured <weight> to ω mapping in DSYN array which was deemed to have failed after 5200 cycles due to trap-up.

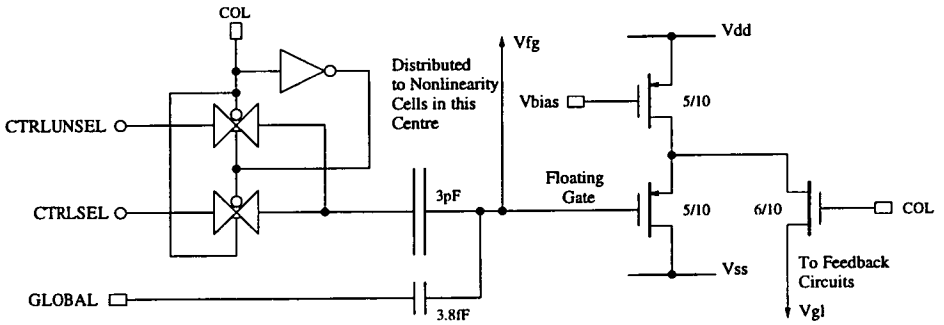
7.7 Feedback Programming of Non-Linearity Array

The fourth test array on *PARAFIN* contained the non-linearity functions which form the final component required for implementing a complete non-volatile pulse-stream VLSI RBF implementation.

7.7.1 Circuit Description



(a)



(b)

Figure 7.28: (a) Non-linearity cell distributed at each dimension of each centre, with output comparator. (b) Non-linearity weight storage cell: floating gate memory and buffer for each centre.

Figure 7.28(a) shows the non-linearity circuit cell which is distributed at each dimension of every centre in the RBF hidden layer. I_{dist} would normally be the output of the Euclidean distance calculator cell although here it is generated externally since the two test arrays had been kept separate. The voltage mode output of each non-linearity cell in the centre is averaged on the non-linearity column using the distributed load

implementation proposed in [110]. This voltage is then pulse-width modulated using a comparator and non-linear ramp as for the multiplier. This PWM signal forms the output of the hidden layer.

Since in this implementation the centres are designed to be radially symmetric, the width of the non-linearity in each dimension should be the same. Therefore there is a single ϕ_{fg} parameter for each centre which sets the width in all dimensions. For feedback operation, ϕ_{fg} is buffered by a p-type source-follower as shown in figure 7.28(b) so that multiple centres may be switched between a single feedback circuit. The source-follower also allows the $\phi_{fg}(range)$ of 0-3V to be stepped up by $V_{dd} - V_{bias}$ (eg. 1V) such that a conventional comparator circuit [3] may be used (ie. with inputs more than a threshold from either power supply). A slightly larger floating gate capacitor (3pF) was used since the floating gate voltage was not required locally but had to be distributed along an array of non-linearity cells; the larger capacitor provided a higher signal immunity from parasitic coupling onto the distribution wire.

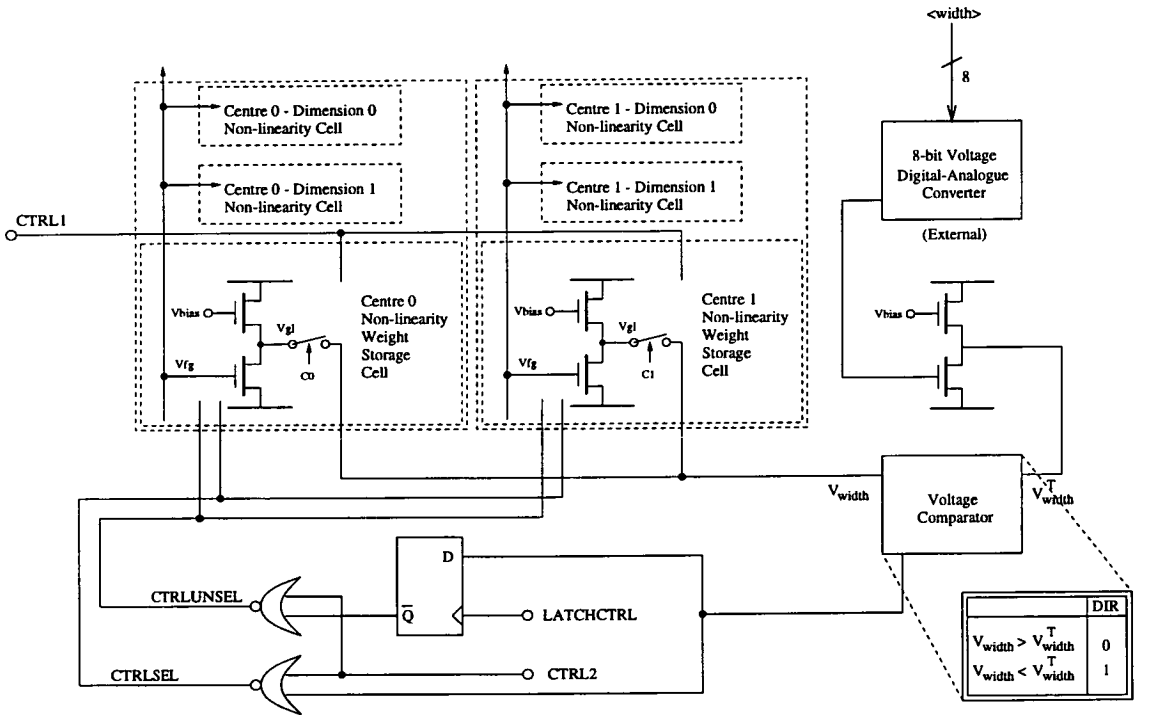


Figure 7.29: *PARAFIN* Non-linearity feedback programming array

The complete programming array is shown in figure 7.29 which implements the non-linearity portion of two 2-dimensional centres. Again the control circuitry is very similar to that already used for the Euclidean distance calculator array and the two multiplier arrays. A 0-3V target voltage is applied (through an external D/A converter), and mapped to 1-4V by a p-type source-follower identical to those used in the non-

linearity weight storage cells.

The sequence of events during programming is also similar to the preceding arrays. V_{PP}^{+} application results in an increasing V_{width} (defined in figure 7.29). Final verification fails if $V_{width} < V_{width}^{T} - \zeta \equiv < width > - < zeta >$ where $< zeta >$ is 9 (ie. the same size (but opposite sign) as used in the SSYN array since $\phi_{fg}(range)$ is the same)).

7.7.2 Experimental Results

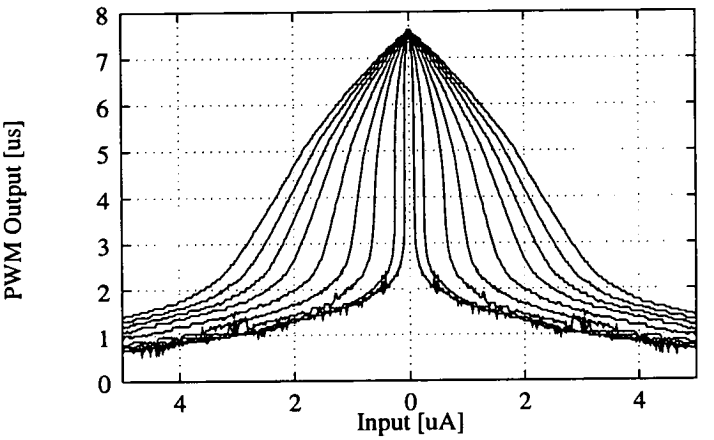


Figure 7.30: Measured non-linearity characteristics for centres programmed to <width> of 025(widest),050,100,125,150,175,200,225,250(narrowest). The trace has been mirrored about the y-axis.

Figure 7.30 shows typical measured traces from one of the non-linearity cells programmed to a variety of widths.

7.7.3 Width Mapping Characteristics

Since the non-linearity function is an arbitrary transistor characteristic, there is no clear metric for it (ie. nothing analogous to sigma in a Gaussian non-linearity) as there was for centre locations or multiplier weights. For measurement purposes, a metric was introduced which is equal to the width (in microamps) of the non-linearity at half of its maximum height (this required pre-calibration of an array to determine the maximum and minimum possible PWM outputs).

A typical mapping measured using this metric is shown in figure 7.31. The mapping is unexpectedly close to linear for <width> values of less than $\sim$ 200. This is not a requirement but may be convenient for determining the optimal non-linearity

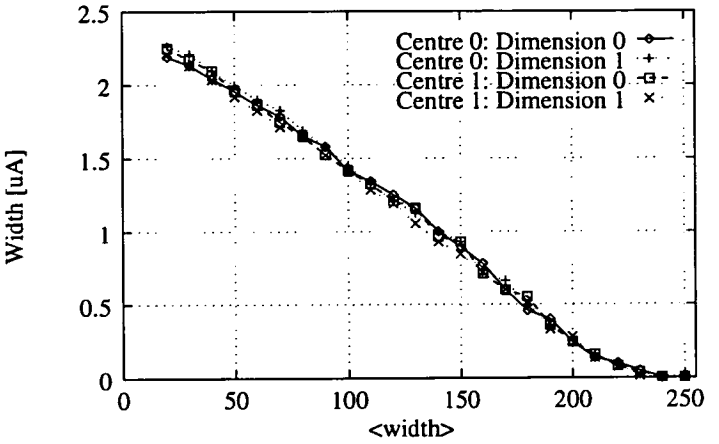


Figure 7.31: Typical measured <width> to width mapping for a non-linearity test array

width. Presently no algorithmic procedure has been developed for this calculation. In the RAM-refreshed RBF [110], the width was selected simply by training on a large set of possible widths (ie. the same width for every centre) and then selecting the best RBF so created.

7.7.4 Disturb Protection and Programming Endurance

One centre was selected and randomly programmed across the entire width range whilst the other centre was held at a preset value. The programming performance of the first centre began to degrade after only $\sim$ 150 cycles, before any disturbance was evident in the other centre. This behaviour was consistent across a number of chips tested. Despite having the same $\phi_{fg}(range)$ as the SSYN array, the observed endurance was only about a third of the number of cycles.

This behaviour must be attributable to the only design change in the floating gate; the move from a 2pF to 3pF $C_{control}$ capacitor. This is because the larger capacitor requires a greater number of electrons to be transported through the tunnelling capacitor oxide. This greater fluence will inevitably lead to more trapping per cycle and this trapping will be at the smaller number of asperities which supports tunnelling (due to the smaller tunnelling capacitor) therefore leading to a more rapid aging of the device. The increase in $C_{control}$ size is therefore detrimental to the floating gate operation and should be reviewed in future designs.

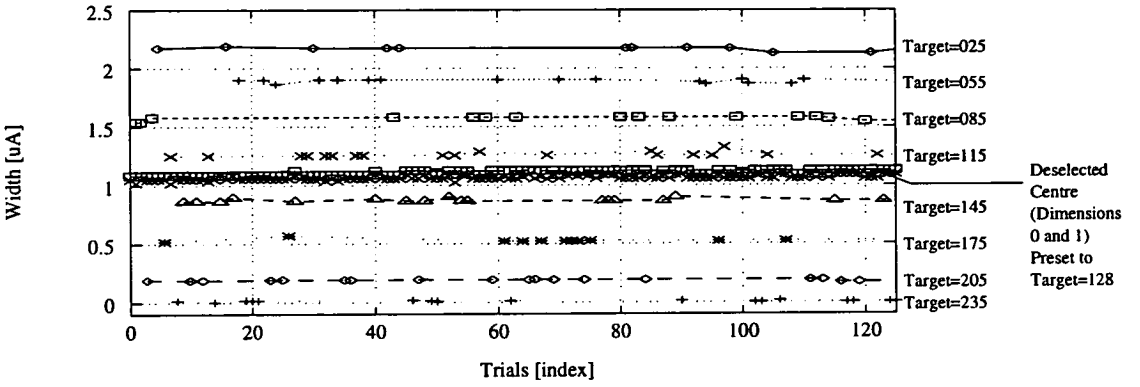


Figure 7.32: Measurement of centre 0 (dimension 0) and centre 1 (dimensions 0 and 1) during repeated programming of centre 0.

7.7.5 Programming Accuracy

Since the <width> to width mapping is non-linear and the number of endurance cycles is comparatively low, it is not possible to make a comparison of accuracy similar to those of the previous test arrays. However, as shown in figure 7.32, a reasonable level of programming precision is visible when 3-bit random programming (ie 8 distinct targets) is used. Note also in the figure that no significant disturb error occurs in the deselected centre.

7.8 Suggested Future Work: Offset Compensation

It has been found undesirable to resort to a large floating gate $C_{control}$ capacitor to limit the influence of parasitic coupling of V_{global} through the tunnelling capacitor as this reduces the endurance of the memory. A better approach would be to develop some form of offset compensation technique which would allow the $C_{control}$ to be reduced in size.

An obvious approach would be to store the target voltage dynamically and allow it to track V_{global} through a capacitor coupling ratio, m_t, the same as for the floating gates. This can be shown to work in simulation but is probably impractical since it is difficult to copy m_t exactly in the presence of real parasitic capacitors, especially when the tunnelling capacitor is comparatively small. Any error in m_t will be greatly magnified by the large amplitude of V_{global}.

An alternative approach is suggested in figure 7.33. Here a number of compensation capacitors, n, are provided (shown here with $n = 4$). These capacitors are matched to the tunnelling capacitor (capacitor matching on chip tends to be good [68]). $V_{compensation}$ is another external voltage which is globally distributed. When V_{global} line

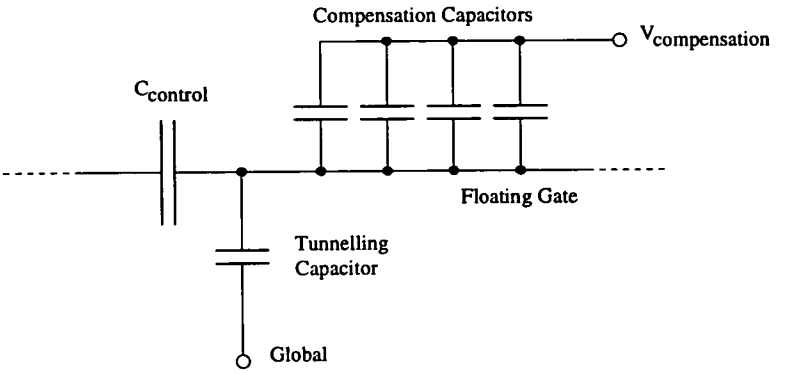


Figure 7.33: Parasitic coupling-compensated floating gate circuit design

runs positive (ie. V_{PP}^{+}), $V_{compensation}$ tracks it in the relationship:

$$V_{compensation} = \begin{cases} V_0 - \frac{1}{n}V_{global} & \text{during programming} \\ V_0 & \text{during preset and read mode} \end{cases} \quad (7.17)$$

where V_0 is a default initial voltage. Using this arrangement, the component of the floating gate voltage due to coupling of V_{global} is cancelled out for any value of V_{PP}^{+}. The division of V_{global} across n compensation capacitors ensures that $V_{compensation}$ is sufficiently low that no tunnelling occurs in the compensation capacitors. n must be chosen to be sufficiently large and V_0 sufficiently small to prevent parasitic tunnelling (including in deselected cells where $AMAX$ will be coupled onto the floating gate through $C_{control}$) and to limit the shift in ϕ_{fg} due to the continuous application of static voltage, V_0, during read mode. The values $n = 4$ and $V_0 = +4$V are suggested as good values. Using the floating gate dimensions of $TARDIS$ and $NEMO$, this would result in a $\phi_{fg}(range)$ shift of -1V due to V_0 in read mode, and $V_{compensation}$ would range from 4V to about -3V during programming which provides adequate margin against reprogramming deselected cells. Of course, if a larger area is available n may be increased to further enhance this margin and reduce the $\phi_{fg}(range)$ shift.

Unfortunately there was no further time available to investigate the issue of offset compensation further.

7.9 Retention

Because $PARAFIN$ was submitted to fabrication before the retention failure on $NEMO$ had been observed, the same floating gate layout (ie. with poly-metal1 contacts) was used. This was expected to result in similar leakage behaviour and this was indeed what was found in a small number of power-up retention tests.

Array	$\phi_{fg}(range)$	$C_{control}$	Min. Endurance	Failure
Euclideans	1V - 4V	2pF	> 500 cycles	V_{PP}^{+} limit
SSYN	0V - 3V	2pF	> 900 cycles	both limits
DSYN	0V - 2V	2pF	> 4000 cycles	V_{PP}^{-} limit
Non-Linearities	0V - 3V	3pF	> 150 cycles	V_{PP}^{-} limit

Table 7.2: Summary of feedback programmable RBF subcircuit arrays performance. Minimum endurance is defined as the minimum number of random reprogramming cycles before an unrecoverable preset or programming verification failure in the worst cell of the worst chip tested.

Some of the floating gates were sensed by n-channel transistors, rather than p-channel transistors as was the case with *NEMO*, and this was not found to result in any significant change in the retention behaviour. This supports the suggestion that SHE injection is not a major component in the leakage behaviour during power-up retention testing.

7.10 Discussion

The test arrays on *PARAFIN* have demonstrated that it is possible to download weights to RBF subcircuits using continuous-time feedback to halt the programming. Typical programming times per weight are 100ms or less which is much faster than for iteration and therefore more practical for CIL trimming. However this is at the expense of more complicated circuits and consequently larger area. Programming accuracy is also less than can be achieved with a large number of iterations.

Table 7.2 summarises the performance of the various RBF subcircuit arrays tested. Due to both the limited time and limited number of samples, the endurance figures quoted should be treated with some caution. However, two observations are clear: endurance is extended by both a smaller $\phi_{fg}(range)$ and a smaller floating gate capacitance. In both cases the reason for this is the reduced amount of charge through the tunnelling capacitor resulting in less trapping and slower aging. Determination of the smallest possible range and capacitance is therefore an important consideration which must be made in any future development of these circuits. Preferably, offset compensation will be investigated as an alternative to using a large floating gate capacitance.

However, even without modification, the endurance is sufficient to implement the same CIL training which was performed on the RAM-refreshed RBF demonstrator chip [110]. Here, the hidden-layer weights (centre locations and non-linearity widths) were loaded onto the chip only once, and CIL was only performed on the output (multiplier) layer. A large learning rate was used and only a few tens of epochs (with

batch-mode LMS updates) were required. The SSYN and especially the DSYN array results suggest plenty of margin is available for this operation. However, the limited reprogrammability of the floating gates does exclude use in sensor processing applications which require frequent complete recalibrations.

It may be possible to implement CIL with parallel capacitor (volatile) storage, only transferring the final weight set to floating gates at the end of training. However, this would add significantly to area and would probably not entirely preclude the need for some CIL training using the floating gates since offsets and inaccuracy in the transferral are inevitable.

Chapter 8

Discussion and Conclusions

8.1 Introduction

The original objectives of the project were detailed in the list on page 3. In this concluding chapter, progress made towards these aims is discussed followed by a more general discussion of the appropriateness of SLV-CMOS floating gates for non-volatile weight storage in analogue VLSI ANNs.

8.2 Experimental Observations and Conclusions

Three test chips were successfully designed and fabricated during the course of this work. Most of the project objectives can be straightforwardly assigned to specific chips. The discussion in this section will therefore be partitioned by chip.

8.2.1 *TARDIS*

Summary

TARDIS comprised a number of SLV-CMOS floating gate test structures. The emphasis of interest was on corner, edge and poly2-poly1 overlap FNT enhancement behaviour in interpoly capacitors since these were reported as the 'best' tunnelling structures in the literature available at the time. Experiments were carried out to test these enhancement effects and compare enhanced FNT with gate oxide tunnelling and with CHE injection. An `Hspice` tunnelling capacitor model was developed, and iterative programming of a floating gate current reference and Euclidean distance calculator were demonstrated.

Conclusions

The *TARDIS* test chip demonstrated that analogue floating gate memories with interpoly tunnelling capacitors were feasible in the available SLV-CMOS process, and that they could be successfully programmed using bi-directional FNT. Tunnelling voltages were a few volts beneath oxide breakdown voltages but high enough to cause damage to CMOS diode structures.

- Corner and edge dependency was investigated by the use of a number of test structures of various peripheries and numbers of corners. Neither feature was found to be significant. However poly2-poly1 overlaps did appear to reduce the required tunnelling potentials by a few volts. The optimal design of interpoly tunnelling capacitor is therefore one consisting of a poly2-poly1 overlap but without the need for area-hungry long peripheries of elaborate multi-cornered layout.

- A large random spread of tunnelling thresholds, rapid trap-up and observations of other researches suggested that the tunnelling current was most likely concentrated at surface asperities, although the exact nature of this behaviour was unclear - specifically the action of the poly2-poly1 overlap. The issue of plate area in the tunnelling capacitor has not been satisfactorily explored primarily because early literature described a tunnelling current concentrated at specific layout features and not distributed across the bulk of the capacitor plate. A future experimental study of this aspect may be worthwhile.

- Bi-directional FNT was also investigated in the gate oxide of the floating gate sense transistor. Despite the thinner oxide, tunnelling thresholds were higher than for interpoly tunnelling capacitors. This is explicable by the higher quality of the gate oxide, since its smoother injection surfaces are not prone to asperity-related electric field enhancement. Similarly, a reduced trap-up was observed. However, the initial inherent spread of tunnelling thresholds was actually worse than for the interpoly tunnelling capacitors, perhaps due to significant oxide thickness gradients.

- CHE was investigated as an alternative injection mechanism. While there was a straightforward, if extremely non-linear relationship between tunnelling potential and current (ie. higher voltages, higher currents), CHE injection required balancing drain and gate voltages to achieve best performance; even when this was done (by use of a simple system model) programming was still about an order of magnitude slower than FNT. High currents were generated by the injection biases, and current limiting precautions were necessary to prevent permanent damage in many cases.

- An FNT model was developed by fitting parameters to the Fowler-Nordheim expression. The predictive power of this model was limited since it (i) did not explicitly incorporate trap-up, and (ii) did not attempt to account for the distortion to the Fowler-Nordheim expression by the asperities. However it was demonstrated as sufficient for simple evaluation modelling of future floating gate RBF

subcircuit designs during programming. The complexity of CHE behaviour was better suited to a piecewise database approach.

- Finally, analogue programmability was successfully demonstrated in two simple circuit applications. Some initial drift was observed in the programmed value but good long term stability over 1 year was seen. Rigorous retention testing was deferred due to uncertainties over ESD damage.

8.2.2 *NEMO*

Summary

NEMO comprised RBF test subcircuits, and three different schemes for addressing and programming the floating gates used to store network weights in the RBF subcircuits and subcircuit arrays: high-voltage switches, the 'dual-phase' programming scheme and the 'CHE flash' programming scheme. Experiments were carried out to determine the suitability of the SLV-CMOS floating gates for weight storage within the ANN subcircuits and to compare the performance of the three floating gate addressing/programming schemes. The iterative programming algorithm developed for *TARDIS* was upgraded to implement an automatic weight downloading system analogous to that of ETANN.

Conclusions

During design, it was found to be remarkably easy to convert the RBF subcircuits developed for capacitor refresh to use floating gate storage; a floating gate is, after all, simply a non-leaky capacitor. Whilst such a direct replacement did not take advantage of the full functionality of floating gates, being instead simply a fixed gate voltage reference, it meant a rapid migration from preceding designs and, in some cases, direct re-use of cell layout. Successfully addressing and programming the floating gates, however, was a more difficult undertaking.

- High-voltage switching in SLV-CMOS has already been demonstrated by other researchers. Such switches were also found successful here in the available fabrication process, and a novel configuration was also demonstrated as successful although less ideal than the established design since the switch could not be fully turned off. Despite DRC-violations, fabrication reliability appeared to be promising. However the area requirement was significant (although this may be improved to an extent in future by a less cautious approach to protection circuitry).

- A more compact, and DRC-compliant, circuit design called the 'dual-phase' scheme was demonstrated to be successful given sufficiently high control signal voltages to provide a high tunnelling potential differential allowing for disturb errors in deselected cells over a large time or number of programming iterations. The relatively high potential differential required was unexpected given the exponential tunnelling current dependency on tunnelling voltage predicted by the Fowler-Nordheim equation. However, the believed asperity-dominated nature of tunnelling greatly detracts from the validity of this equation leading to a reduced voltage dependency. Asperity-dominated tunnelling oxides are therefore not desirable for use with a potential differential tunnelling scheme. Use of high control voltages is not an ideal solution since gate oxide tunnelling and/or SHE injection in the sense transistor is induced. Also, this behaviour demonstrated that whilst the `Hspice` FNT model developed using *TARDIS* is adequate for modelling programming (ie. large tunnelling potentials), the behaviour of deselected floating gates (ie. low tunnelling potentials and parasitic tunnelling currents) is not adequately encapsulated.

- Programming with switched CHE was also possible but required large currents, cumbersome current surge protection and large drain switches. Injection required algorithmic optimisation based on knowledge of the present floating gate charge making 'flash' erase techniques awkward since programmed charge may be outwith the sensing range of read-out circuits (over-erasing).

- Iterative programming of the RBF circuits was possible using the algorithm modified from *TARDIS*. However downloading of weights was slow, not necessarily because of the algorithm but because of the time overhead of evaluating the intermediate states of the parameterised cells. Direct measurement of floating gate voltages through a buffer would reduce this inter-pulse evaluation time but at the expense of larger circuits and reduced implicit trimming.

  The algorithm was tested with up to 8-bits precision during download. However, it is suspected that there is subsequent rapid drift in the programmed value followed by gradual decay over long periods of the stored data which would degrade this value. The extent of this effect was undetermined since these intrinsic effects were overwhelmed by a relatively fast leakage, probably due to a thin conducting layer between two oxide layers. Floating gates must be kept within a single polysilicon layer and not run through contacts.

8.2.3 *PARAFIN*

Summary

PARAFIN comprised RBF test subcircuit arrays which could be programmed by the dual-phase programming scheme. The circuits themselves had been modified to allow continuous-time monitoring of a circuit parameter during FNT programming and feedback circuits exploiting this monitored parameter were designed to allow programming of RBF parameters within a single cycle, rather than a number of pulse iterations, speeding up downloading and reducing aging through trap-up.

Conclusions

Despite problems with higher tunnelling potentials probably induced by a redesign of the tunnelling capacitor, the RBF arrays in *PARAFIN* could be successfully programmed and re-programmed 100s or even 1000s of times (sufficient for CIL). Device failure was due to trap-up limiting the programming range and so may be considered more graceful than abrupt oxide failure which causes immediate and irrecoverable loss of network characteristics. Programming precision was generally better than 5-bits equivalence which is believed to be sufficient for recall mode operation, although the effects of short and long term drift remain undetermined for the same reasons as on *NEMO*. Batch-mode CIL should be used to limit the weight downloading (still much slower than capacitor refresh) and to allow reprogramming of all cells in an array at each epoch to prevent accumulation of disturb errors. Alternatively the dual-phase scheme could be replaced with high-voltage switches to remove this source of error.

8.2.4 Project Summary

Overall, the project has succeeded in pursuit of the three specific goals described in the abstract:

1. SLV-CMOS floating gates have been found feasible, analogue programmability has been demonstrated, conclusions on layout optimisation drawn and a model constructed which gives a reasonable description of tunnelling currents at high potentials. Good yield of floating gates was obtained (ESD allowed for) despite DRC-violation of poly2-poly1 overlaps.

2. Floating gates have been interfaced to RBF subcircuits with minimum changes to the RBF library cells, and programming schemes demonstrated which can accommodate the high voltages, currents, device mis-match and device aging inherent in the floating gate memories.

3. Modified RBF circuits have been demonstrated which can be programmed by continuous-time feedback which allow setting of non-volatile parameters on-chip in a way which is as transparent to the user as analogue capacitor storage, if somewhat slower.

These results allow optimism that, if desired, a complete RBF chip can be constructed which can be automatically set up and trimmed by CIL in a manner similar to the volatile RBF chip demonstrated by Mayes [110].

Amorphous Silicon Comparison

Since the examination of SLV-CMOS floating gates in this work was largely prompted by difficulties in previous work with amorphous silicon (a-Si:h) non-volatile pulse-stream ANNs [80], it is worth comparing the two devices here.

Some attributes of the two memory devices related to the context of this work are listed in table 8.1. Poor manufacturability severely affected the practicality of a-Si:h and it clear from the table that in its present state SLV-CMOS floating gates outperform a-Si:h resistors in most respects. Although deselection of a-Si:h resistors is easier because programming depends on high voltages *and* currents rather than simply high voltages, the need for a milliamp rated switch transistor detracts from the benefits of simpler circuitry. Additionally the other principal benefit, programming speed, is denigrated by the unreliable and unpredictable programming characteristics. As was seen with *NEMO*, the long iterative programming times are dominated by inter-pulse state evaluations rather than pulse widths themselves; the erratic a-Si:h characteristics would most likely make programming *slower* than for SLV-CMOS floating gates.

Until the yield, consistency and reliability of a-Si:h can be markedly improved it can be reasonably argued that SLV-CMOS floating gates offer a much more convenient analogue memory device in terms of availability and convenience.

Iteration versus Feedback

NEMO and *PARAFIN* demonstrated both iterative and feedback analogue programming of SLV-CMOS floating gates. Iteration allows smaller chips, and immediate programming precision can be very good but the programming time can be long, and, if the programming algorithm is complex may require extensive intervention by a control PC. By contrast programming by feedback demands a larger chip area (both for larger individual cells and for the inclusion of feedback circuits) and the immediate programming precision depends on the performance of these circuits. However programming can be much faster, cause less aging and require minimal external intervention.

The choice of approach depends on the application. Programming by iteration is preferable if the chip size must be absolutely minimal. However iterative program-

Technology	Amorphous Silicon	SLV-CMOS Floating Gates
Parameter	Resistance	Voltage / Threshold
Fabrication	Standard CMOS + Post Processing	Standard CMOS
Yield	Poor/Unreliable (some devices could not be formed; others formed but not programmable over wide range)	Very good
Size	Small 'active filament' (possibly as small as 0.1μm) but actual pore size limited by quality of photolithographic equipment. Main size limitation is 1-10mA passing access transistor (deselection easy by limiting programming current)	Floating gate capacitor size determined by need for capacitive coupling ratio; ratio extreme for continuous-time feedback (C_{fg} > 2pF). Main size limitation due to cumbersome access and programming circuits (deselection difficult due to spread of characteristics)
Typical Programming Voltages	Erratic: 10V-18.5V forming pulses. 2-5V required for programming giving rise to 6-12V programming pulses allowing for drop across access transistors.	11-15V required for programming giving rise to > 20V programming pulses allowing for coupling ratios, stored charge and trapped charge; higher for gate oxide. Lower voltages but higher currents for CHE injection
Programming Consistency	often unpredictable	predictable: tunnelling direction always consistent with applied potential
Programming Speed	Fast (nanoseconds)	Slow (milliseconds)
Programming Method	Iteration (poor control of programming led to programming by trial-and-error)	Iteration or Feedback
Programming Precision Equivalence	4-bits quoted	> 5-bits (subject to allowance for drift)
Retention Tests	Power-down for at least 1 week	Power-down for at least 1 year; Power-up for at least 2 weeks (leakage problems in later chips)
ANN Built in Project	MLP	RBF Subcircuit Arrays

Table 8.1: Comparison of a-Si:h resistors and SLV-CMOS floating gates as pulse-stream ANN memory elements

ming soon becomes impractically slow if the network size is large and multiple CIL epochs are necessary from trimming. A fast compromise may be to incorporate both volatile *and* non-volatile storage at each cell. Capacitors may be used for quick weight updates during CIL with the final values transferred to floating gates once the ANN's performance has been optimised.

Future Work

Several issues have arisen during the course of the project which could not be adequately investigated at the time. These issues have been labelled 'future work' since they may be worthy of investigation in any possible furthering of this work.

- **Retention:** While $TARDIS$ demonstrated both short and long term holds the quantity of information obtained was not particularly satisfactory. Further investigation in the later chips was thwarted by the running of floating gate through poly-metal contacts which shorted them to a parasitic conducting surface. There are a number of issues here. Short term drift must be examined to determine the ultimate programming precision; consistent drift may be accommodated by software fixes, random drift cannot. Long term drift may be examined by bake retention testing to give an operating lifetime for the chip (a large number of samples and high temperature packaging are required). It is also necessary to investigate power-up hold behaviour - in the context of neural sensors, data must more likely be held during continuous operation rather than in storage. Degradation due to operating effects, such as hot carrier injection cannot be discounted. Special consideration must be taken of p-channel transistors which may be more susceptible to SHE injection. Additionally, possible deterioration in retention due to repeated cycling must also be investigated.

- **Gate Oxide:** Tunnelling in the gate oxide was little investigated because the literature available at the start of the project strongly favoured the use of inter-poly tunnelling capacitors due to their lower tunnelling voltages. However, the reduced tunnelling enhancement for the gate oxide tunnelling that *was* observed suggested that the behaviour was not asperity-dominated. The use of gate oxides *may* therefore provide better disturb protection in the dual-phase scheme (although the wider spread of tunnelling thresholds observed on $TARDIS$ for gate oxides may conspire to annul this advantage) and may lead to reduced trap-up over protracted number of cycles (allowing longer CIL phases). None of the addressing schemes demonstrated on $NEMO$ exhibited characteristics which would prevent gate oxide implementation. Use of gate oxide would permit migration to a pure digital CMOS process without the need for a second polysilicon layer.

- **Temperature Compensation:** No attempt has been made to devise temperature compensated circuits since this was similarly not a feature of the RAM-refreshed RBF demonstrator. However, temperature compensation may become important if neural sensors are to be applied in hostile environments. Fortunately compensation techniques are no more complicated than for ordinary MOSFET circuits (eg. differential configurations) and so should not prevent any additional difficulties. Care must be taken not to over-engineer the circuit, however, since the suitability of analogue VLSI for ANNs is in part due to ANN tolerance to imperfections and non-idealities: it should be ascertained whether drift over a specific temperature range *does* lead to an unacceptable loss of network performance.

8.3 Observations on Non-Volatile ANN Weights

Given circuits with good dynamic range it might be possible to download tolerant weight sets and so eliminate the need for CIL [60]. Thus with hard-wired weights a working product will come off the production line without time consuming and complicated training procedures eliminating entirely the problem of analogue non-volatile weights.

Unfortunately analogue circuits with the required characteristics are difficult to design, especially to design compactly. Even if such circuits existed it is still desirable to allow post-manufacture weight changing, for example to provide a more generic product or, in the context of this project, provide recalibration to compensate for manufacturing spread or time-dependent drift of input sensor characteristics. Use of tolerant weight sets would be more likely to reduce the need for CIL, rather than the need for weight programmability.

8.4 Observations on the Applications of Analogue SLV-CMOS Floating Gate Memories to ANNs

Although it has been shown both here and in the work of other researchers that non-volatile memories *can* be implemented in a SLV-CMOS process, the practicalities of doing so require larger and more complicated circuits (impacting on both silicon area required *and* manufacturing yield) than would be required if a non-volatile process with lower tunnelling thresholds, higher voltage handling capabilities and better quality oxides were specifically used. The higher engineering design costs of the more complicated circuits, the more convoluted programming procedures, and the larger area of circuits all conspire to quickly overcome any cost benefit gained by using a standard process. Here more sophisticated processes are being traded for area, speed

and density and so using a process for something for which it was not intended quickly becomes costly. (A specific exception may be designs which embrace SLV-CMOS shortcomings rather than attempt to fight them. For example Diorio *et al* use the very slow programming times to implement a gradually adaptive system).

Although non-volatile memory processes require extra masking steps, due to multiple gate oxide steps, which also increases the time through fabrication, the number of additional masks is small with CMOS processes increasingly taking on non-volatile options as simple variations. In some cases these extra masks may even reduce cost by improving yield on an otherwise hard to control process. In fact, the majority of the addition cost of non-volatile technology is now due to complex test procedures and burn-in[1] . Some customer specifications, such as read/write endurance and data retention are slow and hard to test. These costs are not removed by using a SLV-CMOS, and in fact, are probably greater since design rules are being broken, undocumented and unguaranteed characteristics being used and out-of-specification voltages being applied. Additionally, the onus on non-volatility testing is removed from the foundry with its dedicated test equipment, specialist knowledge and aggregated in-house experience and placed on the VLSI designer who may have none of these advantages yet would be expected to attain the same levels of confidence about reliability in a commercially specified product. And before starting the design cycle, the VLSI designer must also fully characterise and build simulation models for the standard process when these would normally be provided as a matter of course when subscribing to a EEPROM process. Obviously, since these features are undocumented and unguaranteed, there is also no way of being certain that an identical design on a subsequent run will behave in the same way.

In the limit, it might be expected that the actual fabrication of non-volatile processes would tend to cost little more than SLV-CMOS, whilst greatly outperforming it in functionality, run-to-run reliability, yield and testability. It must, however, be noted that the market for EEPROM technologies is currently much smaller than that for RAM which results in a relative monopoly (meaning that for commercial rather than technical reasons price comparability may be unlikely to be achieved). Additionally, integration density is hindered by the lag of EEPROM behind state-of-the-art CMOS. The situation may change in future with the growth of the market for mobile telephones, digital cameras and smartcards (eg. e-money or pay-TV) which incorpor-

[1]This paragraph is based on e-mail discussions with a number of observers. Special acknowledgement is due to the contributions of Gregory Sabin of Intel, Hank Walker of Texas A&M University and Achim Gratz of the Technical University of Dresden. There is little published data to back these claims but, for some quantification, a casual look through a recent semiconductor products catalogue shows stand-alone Flash chips retailing at similar prices to DRAM of the same capacity, and significantly cheaper than equivalently sized SRAM.

ate non-volatile memory technology.

However, given the currently existing price differentials and lack of general availability of dedicated floating gate technologies for VLSI researchers, SLV-CMOS has been shown to fill a role in providing a cheap and reasonably convenient electrically programmable non-volatile memory storage in test circuits, where exploration of the functionality and utility of various ideas is more important than attaining a high yield, meeting tight commercial specifications or having thorough knowledge of long-term reliability.

These observations lead to the following arguments:

- Programmable/trimmable floating gates ANNs should not be used unless these are a justified means of implementing a sensor system. A hardwired system or volatile storage are preferable if suitable.
- SLV-CMOS should not be used in preference to an EEPROM process in a commercial product for the simple replacement of volatile capacitor storage because any savings in cost are effectively lost in counter measures such as extra silicon, design time and reliability assurance problems. SLV-CMOS may be attractive only if its characteristics may be made useful rather than problematic.
- SLV-CMOS floating gates may be useful as memory in a modular prototype design at the early stages of development where the inconvenience and high cost of obtaining an EEPROM process cannot be justified. At this stage the use of SLV-CMOS high voltage switches is best as it provides the closest functionality to that of a special EEPROM process. If the design progresses to development, the modular memory can be replaced by more compact, reliable and pre-tested cells.

8.5 Summary

In this project three operational SLV-CMOS floating gate chips have been demonstrated. *TARDIS* demonstrated that, in interpoly tunnelling, there were no edge/corner effects expected from the literature, and that FNT was faster than CHE injection. *NEMO* demonstrated interfacing of analogue floating gate memories with pulse-stream RBF circuits and also three addressing schemes including a new differential voltage approach but highlighted deselection problems due to the non-ideality of tunnelling characteristics with asperities. Flash methods of bulk erase were found unsuitable for analogue applications. Finally, *PARAFIN* used modified RBF subcircuit designs and support circuits and showed that it was possible to use continuous time feedback to optimise programming speed.

Analogue floating gates have therefore been demonstrated to be a useful memory element in CMOS VLSI circuits, particularly for weight storage in ANNs as investigated. Such devices can be fabricated with dedicated floating gate fabrication technologies or in SLV-CMOS processes. It is believed that use of a SLV-CMOS process commercially, despite its nominally lower cost, could actually result in a more expensive and less reliable product. However, SLV-CMOS can be appropriate as a tool for investigating circuit concepts and development of circuit prototypes given its lower cost price, easy availability and compatibility with small manufacturing runs[2].

Thus SLV-CMOS floating gates would seem to be more appropriate to small scale prototyping rather than forming dense analogue memories in hardware neural network smart sensors as was originally envisaged. Finally, one other promising area of use would be to form a small number of trimming elements for large analogue circuits (where the floating gates' larger size is proportionately less important) such as amplifiers where the whole circuit would otherwise be cost penalised by the need for dedicated processing.

[2]Europractice, for example, consolidates the designs from many users onto a single wafer (Multi Project Wafer) thus sharing fabrication cost.

Appendix A

Layout, Bonding and Pin Diagrams

A.1 Introduction

Three chips were fabricated during the course of this project: *TARDIS*, *NEMO* and *PARAFIN*. *TARDIS* was processed by Eurochip, *NEMO* and *PARAFIN* by Europractice. All chips were fabricated in the Alcatel-Mietec 2.4μm double-poly double-metal n-well CMOS process. This appendix provides chip plots, bonding diagrams and pin-out listings for the three chips.

- The *TARDIS* chip had an area of 3.6072mm by 2.704mm (post-shrink). It was assembled in a 40-pin DIL package. Ten packaged samples were delivered in June 1995. A bonding diagram is shown in figure A.1.
- The *NEMO* chip had an area of 4.5736mm by 4.0712mm (post-shrink). It was assembled in an 84-pin PGA package. Ten packaged samples and a small number of unpackaged samples were delivered in August 1996. A pin-out list is shown in table A.1 and a bonding diagram is shown in figure A.2.

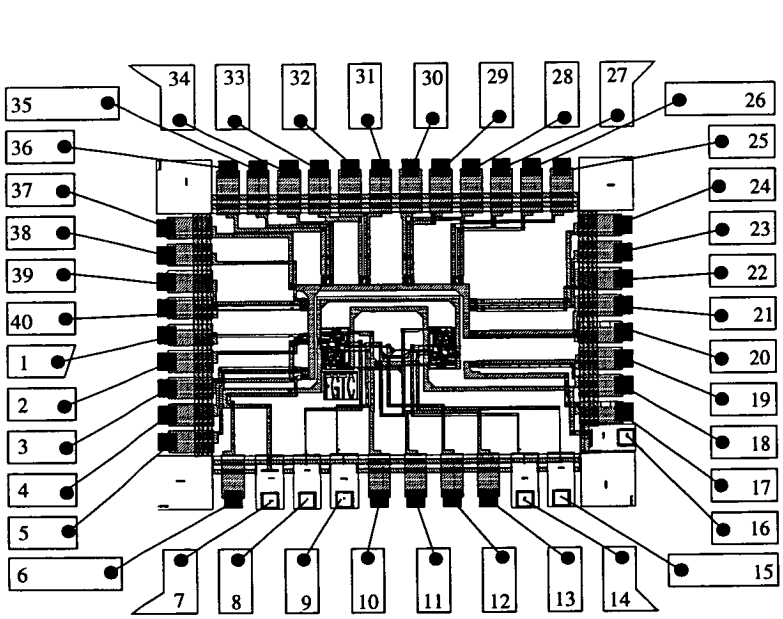
North†		East†		South†		West†	
81	VSS	60	VSS	38	HVS2_OUT	17	C0
80	AMAX	59	VDD	37	HVS_BAL	16	_HOTWR
79	MD_IN/TAU_1	58	EDHE_OUT	36	ONEUA	15	CTRLUNSEL
78	TAU_2	57	ED_HE_COUPLE	35	ED_TUNNEL	14	CTRLSEL
77	TAU_3	56	ED_HOTLINE	34	ED_COUPLE	13	BALANCE
76	MD_SCS_IN/TAU_4	55	EDP_OUTPUT	33	FIVEUA	12	SIXUA
75	MDHE_OUT	54	ED_OUTPUT	32	RAMP	11	INIT_VOLTS
74	MDP_OUT	53	HVS_CTRL	31	MD_SCS_TUNNEL	10	RESET
73	EDP_TUNNEL	52	HVS1_BB	30	MD_SCS_COUPLE	9	MD_PWM_OUT
72	ED_HE_TUNNEL	51	HVS1_BIAS	29	MD_TUNNEL	8	MD_SCS_PWM_OUT
71	ED_IN/IN1	50	HVS1_OUT	28	MD_COUPLE	7	MDP_TUNNEL
70	IN1	49	HVS3_BB	27	R1	6	M_HOTLINE
69	IN2	48	HVS3_OUT	26	R0	5	M_HE_COUPLE
68	IN3	47	HVS2_BB	25	C1	4	M_HE_TUNNEL

Table A.1: *NEMO*: Pin List (†Bond-side View)

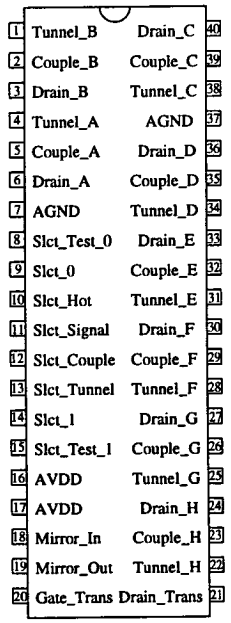
- The *PARAFIN* chip had an area of 5.3284mm by 4.208mm (post-shrink). It was assembled in an 84-pin PGA package. Ten packaged samples and a small number of unpackaged samples were delivered in January 1997. A pin-out list is shown in table A.2 and a bonding diagram is shown in figure A.3.

North†		East†		South†		West†	
83	SSYN_TUNN	60	NL_TARGET	41	FOURTEENFOURUA	18	BIT_1
82	BALANCE	59	NL_TUNNEL	40	AMAX	17	BIT_2
81	CTRL1	58	NL_CURRENT_0	39	VSS	16	BIT_3
80	INIT_VOLTS	57	NL_CURRENT_1	38	ED_LDIR	15	BIT_4
79	RESET	56	NL_CURRENT_2	37	ED_DIR	14	BIT_5
78	DSYN_C1	55	NL_CURRENT_3	36	ED_CTRL2	13	BIT_6
77	R1	54	NL_LDIR	35	ED_OUTPUT	12	BIT_7
76	DSYN_C0	53	NL_DIR	34	DSYN_LDIR	11	TAU_4
75	R0	52	NL_CTRL2	33	DSYN_DIR	10	SSYN_DIR
74	DSYN_TUNNEL	51	NL_PWM_OUT_A	32	VSS	9	TAU_3
73	SIXUA	50	NL_PWM_OUT_B	31	VDD	8	SSYN_LDIR
72	IN4	49	NL_COL	30	DSYN_CTRL2	7	TAU_2
71	IN3	48	RAMP	29	DSYN_IDAC_OUT	6	TAU_1
70	IN2	47	NL_RAIL	28	DSYN_PWM_OUT	5	SSYN_C1
69	IN1	46	EIGHTUA	27	SSYN_IDAC_OUT	4	SSYN_C0
68	ED_TUNNEL			26	SSYN_PWM_OUT		
67	ED_C0			25	LATCHCTRL		
66	ED_C1			24	SSYN_CTRL2		
65	FIVEUA			23	BIT_0		
64	NBUF						

Table A.2: *PARAFIN*: Pin List(†Bond-side View)



Bonding Diagram



40-Pin DIL Package

Figure A.1: *TARDIS* FGTC: Bonding Diagram

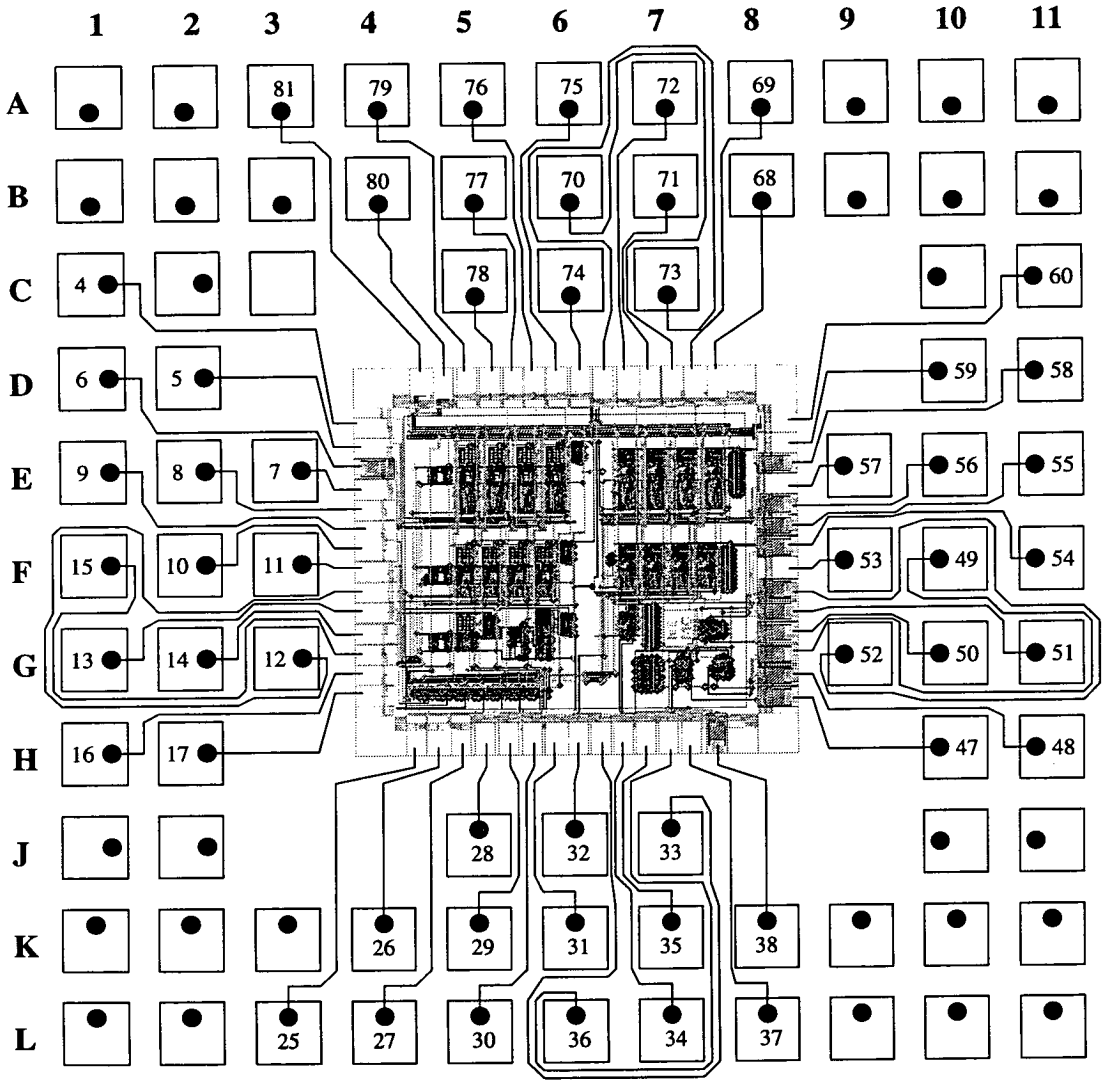


Figure A.2: *NEMO*: Bonding Diagram

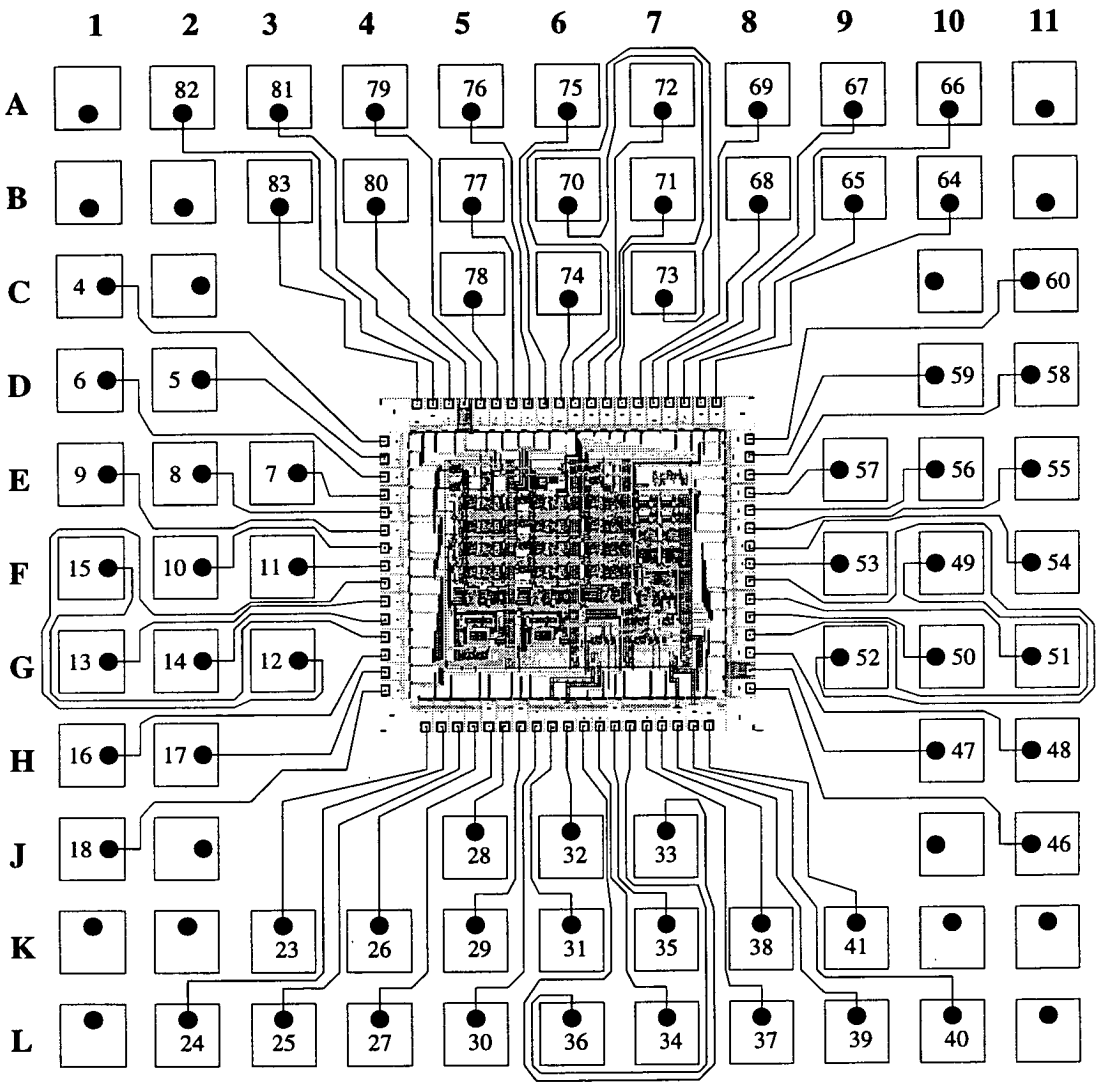


Figure A.3: *PARAFIN*: Bonding Diagram

Cell Name	Length	Height
NEMO		
Directly pinned-out floating gate Euclidean	260μm	265μm
Directly pinned-out floating gate Multiplier	265μm	450μm
Voltage comparator	200μm	245μm
High voltage cross-coupled digital switch	245μm	380μm
High voltage differential switch	200μm	330μm
Dual-phase 4-cell Euclidean array	610μm	1620μm
Dual-phase 4-cell Multiplier array	720μm	1945μm
CHE Flash 4-cell Euclidean array	820μm	1680μm
CHE Flash 4-cell Multiplier array	900μm	1800μm
PARAFIN		
SSYN single cell	275μm	650μm
SSYN 4-cell array	975μm	2150 μm
DSYN single cell	260μm	580μm
DSYN 4-cell array	705μm	2070μm
Current DAC	470μm	500μm
Euclidean single cell	260μm	610μm
Euclidean array	750μm	2400μm
Non-linearity floating gate cell	210μm	290μm
Non-linearity 2x2 array	740μm	1720μm

Table A.3: Approximate (pre-shrink) cell layout sizes for the *NEMO* and *PARAFIN* test chips

A.2 Cell Layout Sizes

Table A.3 lists the approximate test cell layout sizes for the *NEMO* and *PARAFIN* chips. The sizes are pre-shrink and are so slightly larger than the actual fabricated cells. Note however that little effort was given at design time to minimisation of the cell area: quick and correct design were considered more important at these prototyping stages. Therefore the sizes listed could conceivably be somewhat reduced with more aggressive layout and are only provided as a means of qualitative comparison.

Appendix B

Test Equipment and Programmer Boards

B.1 Introduction

To test the three chips developed during this project a number of circuit boards were designed and constructed. This appendix briefly catalogues the different test boards used during the course of the project.

B.2 *TARDIS*

The test board for *TARDIS* floating gate test chips is shown in figure B.1. It comprises principally of two pulse amplifiers: one negative going and one positive going, to generate the programming pulses. The gain of these amplifiers is determined by the op-amp supply voltages which are set from the GPIB power supply under the control of the test program running on the PC. Suitably calibrated, this arrangement can provide very accurate control of pulse heights. PCB relays are used to direct the pulses, and other high voltages to control the programming configuration. Individual floating gate structures were selected for observation by moving a set of miniature jump connectors on the board. Also, on the board is a transresistance amplifier and analogue-to-digital converter, which allows the PC to measure the drain current in the memory device and thus implement iterative programming algorithms.

B.3 *NEMO*

On *NEMO* there were a number of different test blocks. Instead of constructing a huge test board capable of interfacing with all these blocks, several smaller test boards were constructed independently. Each board tested one or more blocks depending on similarity of signal and interfacing requirements. The boards, grouped by test block, were:

- Board 1: Euclidean Distance Cell and Dual-Phase 4-input Euclidean Distance Calculator
- Board 2: Multiplier Cell and Dual-Phase 4-input Perceptron
- Board 3: Hot Electron Euclidean Distance Calculator

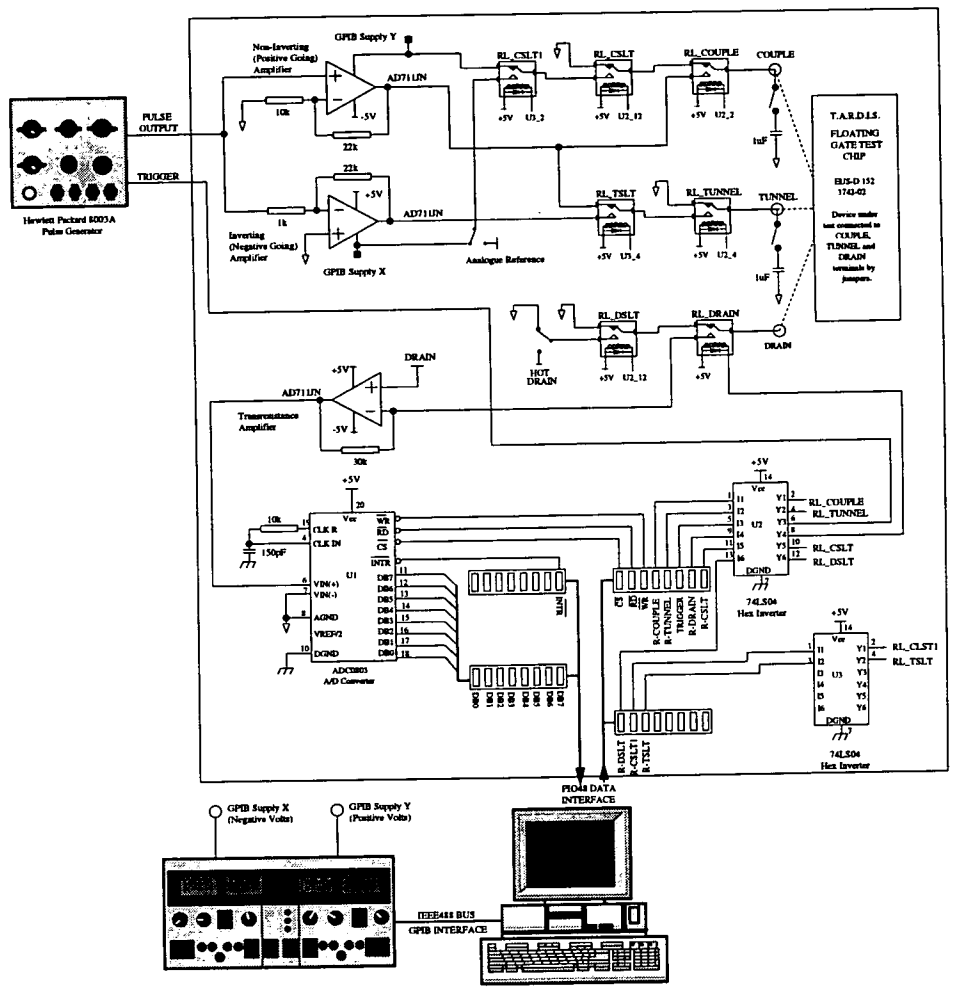


Figure B.1: *TARDIS* FGTC: Test Board

- Board 4: Hot Electron 4-input Perceptron
- Board 5: High Voltage Driver Test Blocks

TARDIS experiments had shown that programming pulse durations of the order of tens or hundred of milliseconds were required. This was within the operating capabilities of the control PC directly switching board relays (with an internal latency of about 2ms); the precision pulse generator could therefore be freed for other users and on-board pulse amplification was no longer necessary.

B.3.1 *NEMO* Board 1

The first *NEMO* test board was used to test the Euclidean distance calculator (pinned out and Dual-Phase array). A circuit diagram for the board is presented in figure B.2.

Relays to direct gate pulses onto *NEMO* from PC-controlled voltage outputs from the GPIB power supply. The positive power supply was also used to provide the analogue input voltage to the distance calculator during centre evaluation. Measurement of the distance current was via a A/D converter on the board or via a GPIB current meter when available.

B.3.2 *NEMO* Board 2

The second board is described in more detail due to its somewhat greater complexity. This board was used for pulse generation and measurement and was later modified to form several future test boards. The requirements of this board were to:

- Provide bi-polar variable magnitude tunnelling pulses (time scale of milliseconds)
- Provide dual-phase logic signals: addressing and CTRLSEL & CTRLUNSEL
- Input PWM test signals (time scale of microseconds) to the chip
- Read PWM output signals (time scale of microseconds) from the chip
- Generate appropriate bias voltages, currents and ramps

The first two requirements are identical to those of *NEMO* Board 1. However downloading and uploading of PWM signals requires a more complicated test board, especially since the times needed are faster than can be driven directly by the PC; more computational power must be devolved to the board itself. The bias voltages and currents can be derived simply from potential divider circuits with unity-gain buffers if required; potentiometers were included for fine tuning of these analogue parameters. Figure B.3 shows how the analogue biases and the ramp can be adjusted to modify the PWM multiplication characteristics.

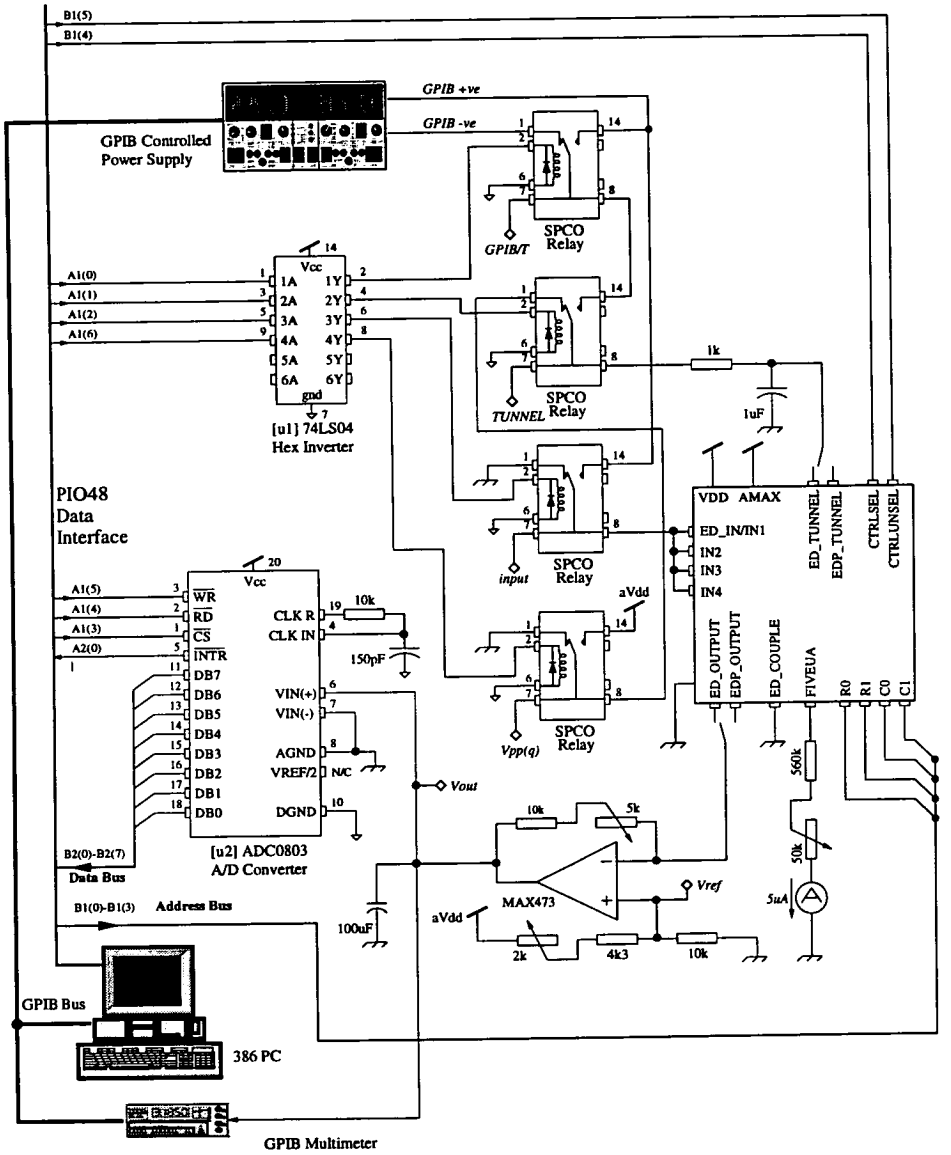
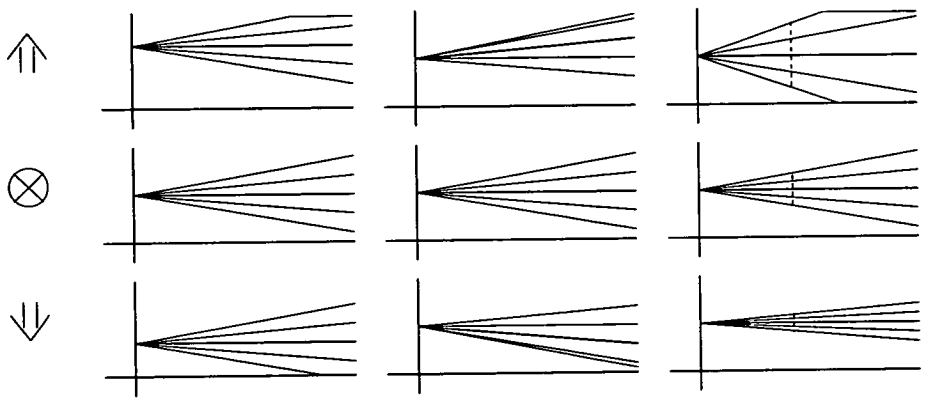


Figure B.2: *NEMO* Test Board 1: Euclidean Distance Cell and Dual-Phase 4-input Euclidean Distance Cell Array



	INIT_VOLTS		BALANCE		Isixua		RAMP (offset)		RAMP (height)	
	reduce	*increase*	*reduce*	*increase*	*reduce*	*increase*	*reduce*	*increase*	*reduce*	*increase*
Zero Offset	⇓	⇑	⊗	⊗	⊗	⊗	⇑	⇓	⇑	⇓
Zero Symmetry	⊗	⊗	⇓	⇑	⊗	⊗	⊗	⊗	⊗	⊗
Output Range	⊗	⊗	⇓	⇓	⇓	⇑	⊗	⊗	⇑	⇓

Figure B.3: Effect of varying analogue bias parameters on the offset, symmetry and range of the PWM encoded multiplier outputs

Generation and Reading of PWM Signals

In common with the original *EPSILON* test board, pulses were generated by loading an SRAM with the required bit patterns and then clocking through using an incremental counter to address the SRAM. This is illustrated in figure B.4 for the four PWM inputs of *NEMO*. A 25MHz clock was used to drive the address counter giving a clock period of 40ns. Therefore a maximum 10μs pulse could be produced by loading (the specific bit of) 250 of the 256 SRAM locations.

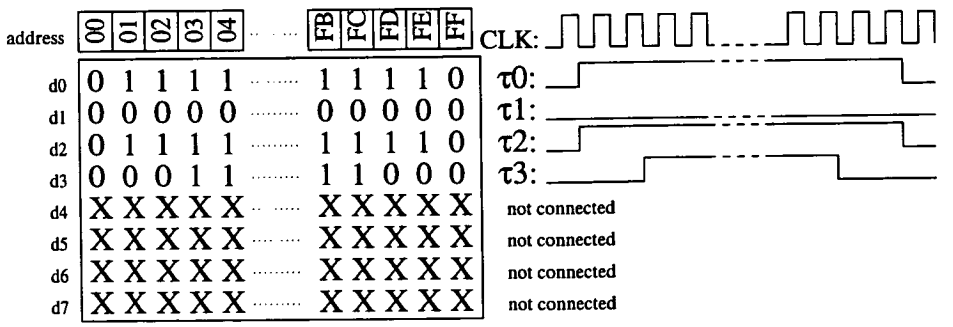


Figure B.4: Generation of four PWM encoded inputs to multipliers/perceptrons by counting through an appropriately loaded SRAM

Since the multiplication/perceptron evaluations performed by *NEMO* are also

PWM encoded, these could be measured by write enabling a separate SRAM which is also incrementally addressed by a counter (effectively the inverse of figure B.4). Following a forward pass, the output data can then be read back to the PC from the output SRAM. The width of the output pulse was determined by the formula

$$(address(last\_1) - address(first\_1) + 1) \times 0.04\mu s \tag{B.1}$$

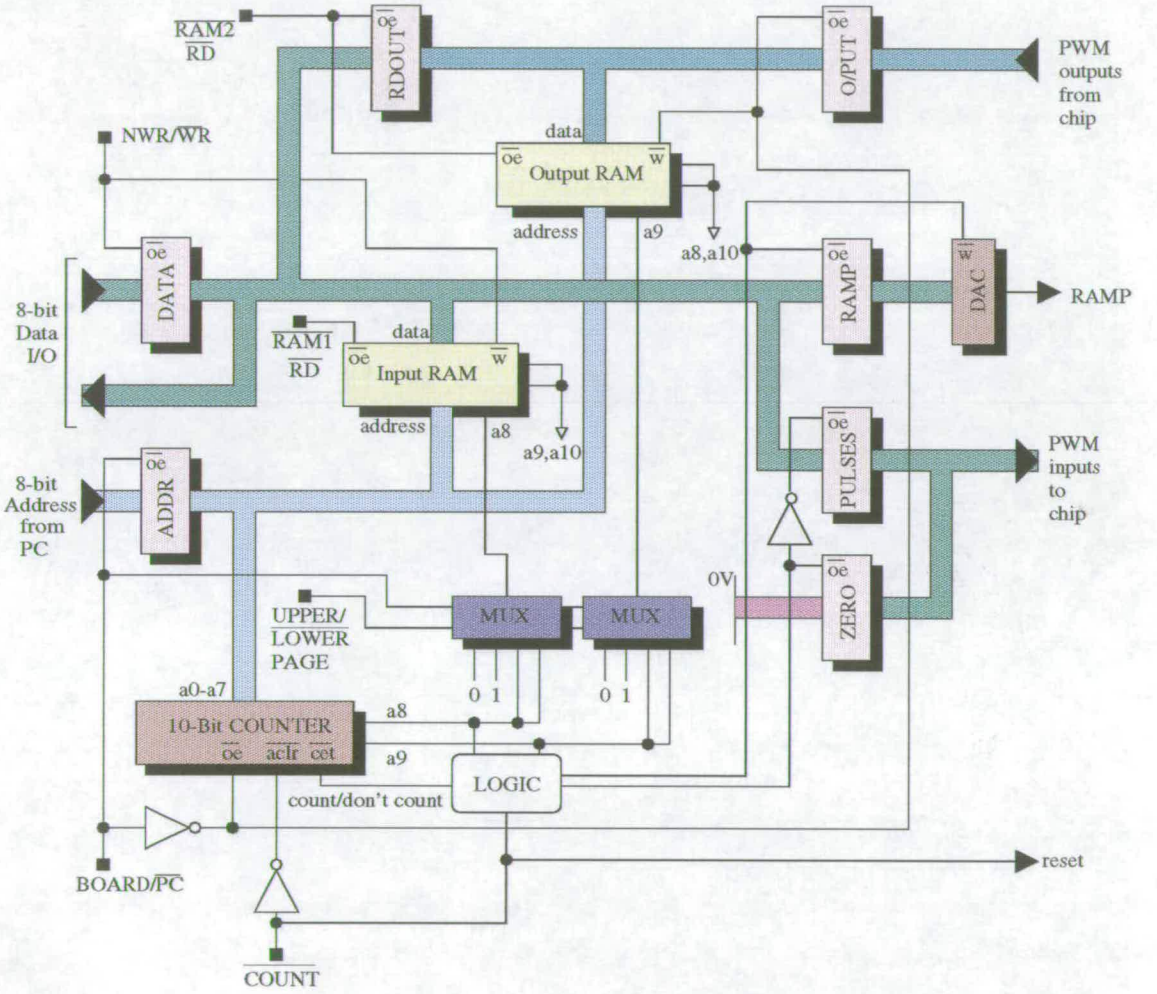


Figure B.5: Schematic for *NEMO* board 2 excluding programming and addressing circuits

Generation of Integration Ramp

The integration ramp could be generated by a triggered signal generator but it was more convenient, in terms of equipment, signal delays and programmability to again

use an incrementally addressed SRAM, this time with an appropriately biased DAC to generate the ramp. The SRAM was initialised as follows:

$$data = \begin{cases} \#FF - (2 \times address), & \text{when address} < (\#FF/2) \\ 2 \times (address - (\#FF/2)), & \text{when address} \geq (\#FF/2) \end{cases} \quad \text{(B.2)}$$

This results in a ramp which starts maximally at address #00, decreases linearly to address #FF/2 (minimal value), and then increases linearly to #FF (maximal value again). Other initialisations could be used for, say, 'on-the-side' sigmoidal ramps for use in MLPs. Since the step between neighbouring data locations is only #2, the generally slow settling time of readily available DACs is not of concern and the use of a small smoothing capacitor led to good linear ramps.

Operation of Test Board

A schematic for the test board is shown in figure B.5. Notice that address bits 9 and 10 of the input S-RAM and 8 and 10 of the output S-RAM are tied to logic low. This restricts the pages of memory which may be accessed, as detailed in figure B.6. During downloading and reading back, when the PC controls the address bus, access to the appropriate page can be obtained by toggling the `UPPER/LOWER PAGE` control line. For both S-RAMs, the lower page is page 0. For the input S-RAM, the upper page is page 1, and for the output S-RAM, the upper page is page 2.

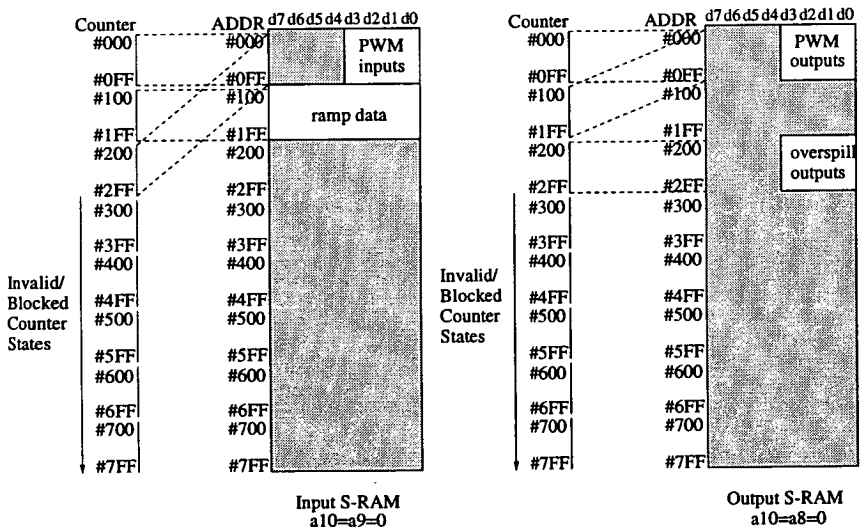


Figure B.6: Memory map for input S-RAM (left) and output S-RAM (right), showing mapping from 10-bit counter values to RAM page accesses.

The operating modes of the control board are listed in table B.1 along with their PC-driven control signals. The forward pass operation, which is controlled by the on-board 10-bit counter (actually three 4-bit counters in look-ahead carry configuration)

$BOARD/\overline{PC}$	$COUNT$	$UPPER/\overline{LOWER}$ $PAGE$	$NWR/\overline{WR}$	$\overline{RAM1}$ $\overline{RD}$	$\overline{RAM2}$ $\overline{RD}$	Description
Downloading Data						
0	1	0	0	1	1	Allows PC to control both address and data bus. The input RAM is write enabled and PWM signal data may be downloading into page 0
0	1	1	0	1	1	Download ramp data into page 1
0	1	1	1	1	1	Disable write on input RAM to finish downloading
Check Downloaded Data						
0	1	0	1	0	1	Allows PC to control both address and data bus. Read back data from page 0
0	1	1	1	0	1	Read back data from page 1
Run Forward Pass						
1	1	X	1	1	1	Set up board for counter control
1	0	X	1	1	1	Activate counter - run forward pass
Read Back Outputed Data						
0	1	0	1	1	0	Read back output pulse data from page 0
0	1	1	1	1	0	Read back any overspill pulse data from page 2

Table B.1: Modes of operation of $NEMO$ test board with requisite control signals

Counter	Input RAM Page	Input Operation	Output RAM Page	Output Operation
#000-#0FF	0	PWM outputs clocked through. Pulse transceiver ON. PWM signals fed into chip. Ramp DAC latched from previous pass to hold analogue ramp signal high.	0	Defunct values clocked in from output of chip.
#100-#1FF	1	RAMP data clocked through. Ramp transceiver ON. Ramp data fed to DAC and analogue ramp fed to chip. Zero transceiver ON /Pulse transceiver OFF ensuring no further PWM input signals to chip.	0	Output PWM signals clocked in from chip overwritting defunct data from previous counter page.
#200-#2FF	0	PWM outputs clocked through but Zero transceiver still ON/Pulse transceiver OFF blocking their passage to chip. Ramp DAC latched to hold analogue ramp signal high.	2	Output PWM signals clocked in from chip, allowing overspill from propagation delay through circuits.

Table B.2: Three stages of forward pass evaluation

is described in more detail by table B.2. There are three distinct stages to this: *(i)* application of PWM inputs, *(ii)* application of ramp and measurement of PWM outputs, and *(iii)* measurement of overspill PWM outputs due to propagation delay. During the forward pass, the MUX controls are set such that page addressing of the two S-RAMs can be performed directly by the counter. When address bits, a8 and a9 are both equal to one (at the end of overspill recording), the counter is forced to stop by on-board logic which prevents unused areas of RAM being addressed. The PC then may reassert control; no interrupts are required as the forward pass occurs much faster than the PC can respond.

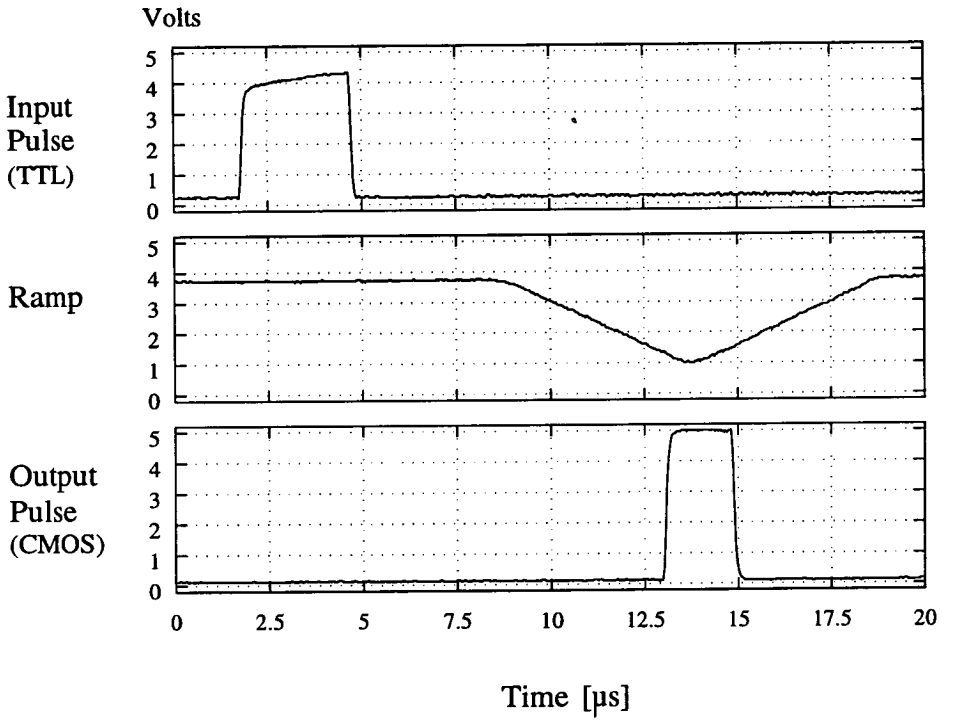


Figure B.7: Oscilloscope measurements of a typical input pulse, ramp and output pulse on the *NEMO* testboard.

Figure B.7 shows typical oscilloscope measurements of input/output signals on the *NEMO* testboard.

The full circuit schematic for the second *NEMO* test board is shown in figure B.8 which also includes the three relays used to apply Fowler-Nordheim programming pulses, and the analogue bias circuits.

B.3.3 *NEMO* Board 3

The test board for the CHE Flash Euclidean array is actually a simple re-wiring of Board 1. The appropriate control signals were changed on the PIO line and ED_HOTLINE

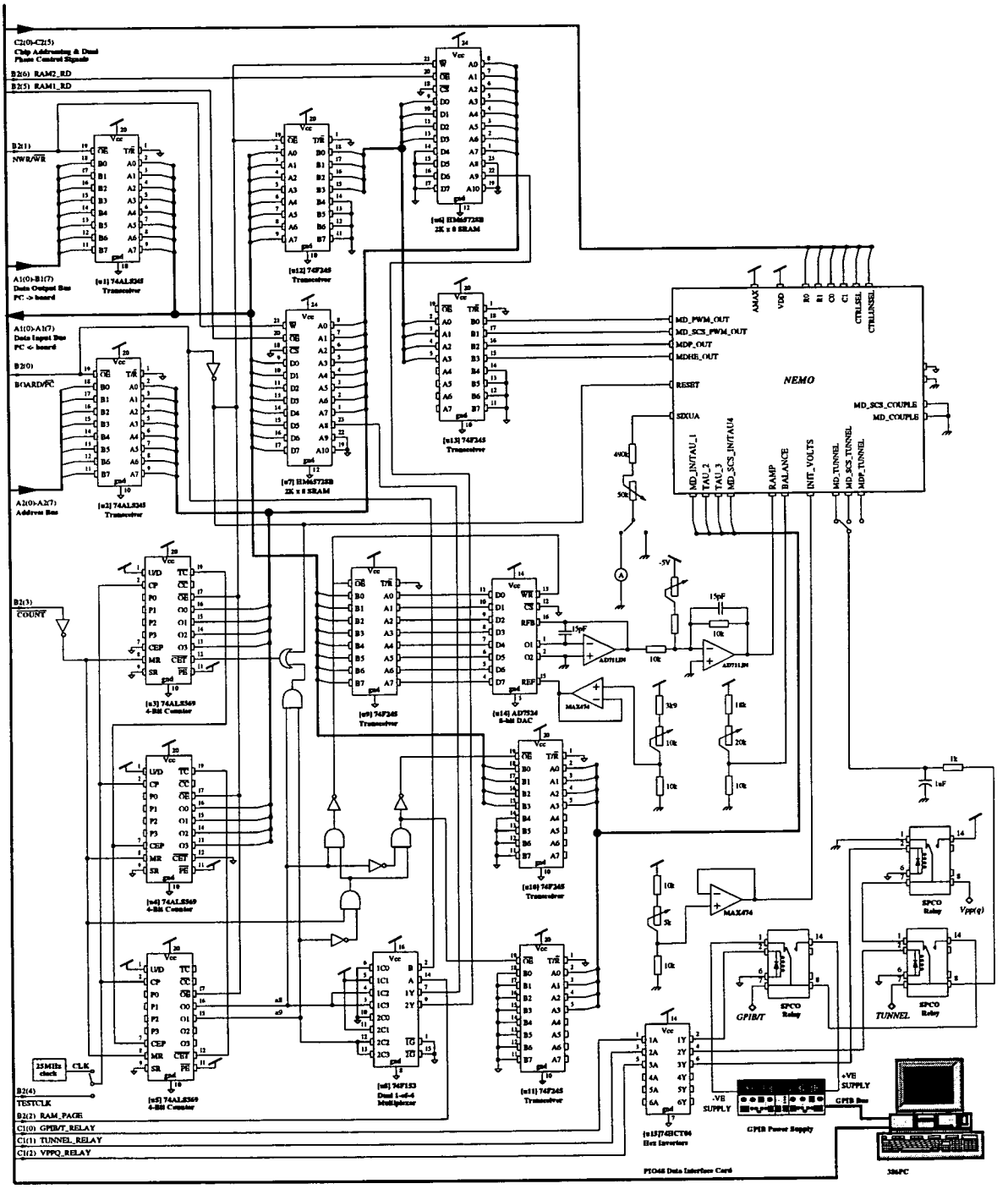


Figure B.8: *NEMO* Test Board 2: Multiplier Cell, Switched Current Source Multiplier Cell and Dual-Phase 4-input Perceptron

and ED_HE_COUPLE connected in from a power supply. ED_HOTLINE ran through the Siliconix 2.5mA current limiter as in *TARDIS*.

B.3.4 *NEMO* Board 4a & 4b

The high voltage switches were tested last in case of any damage to surrounding circuits due to application of power to the then unknown DRC violation structures.

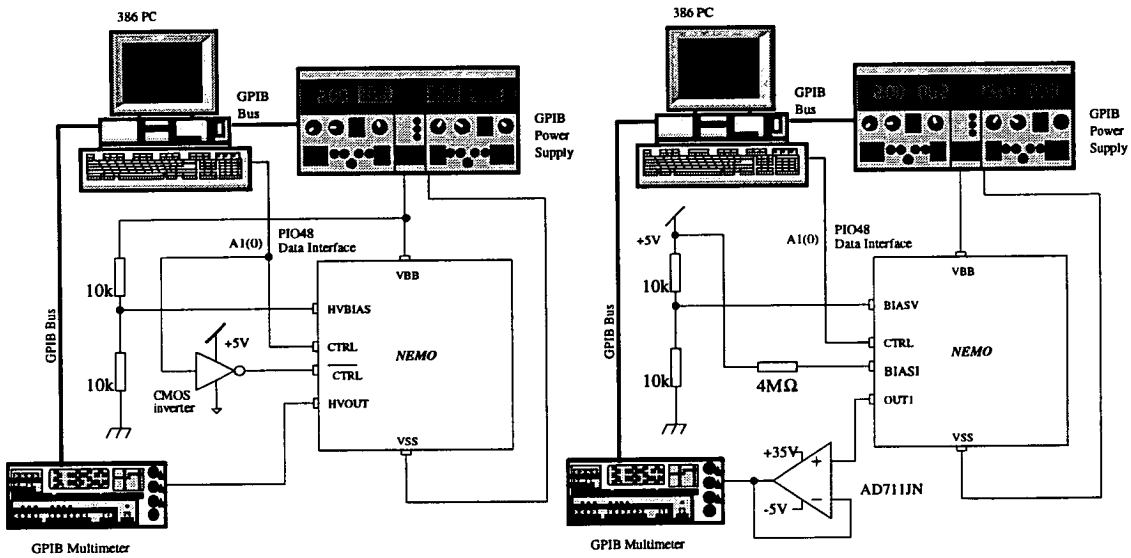


Figure B.9: *NEMO* Test Board 4a & 4b: High Voltage Switches

Test boards for the cross-coupled digital switch and differential stage switch are shown in figure B.9. Notice that no potentiometers have been used to trim out resistor mismatch or set the bias current to exactly $1\mu A$. This is because circuit operation was sensitive to neither the bias voltages nor currents.

B.4 *PARAFIN*

B.4.1 *PARAFIN* Board 1

The first *PARAFIN* board shown in figure B.10 was similar in most respects to the first *NEMO* board and used for programming of the Euclidean distance calculator array. However, the input voltages were now generated from a D/A converter since they had to provided simultaneously with programming signals from the GPIB power supply to allow feedback programming. Current read out was now through the GPIB multimeter since it was readily available at this time.

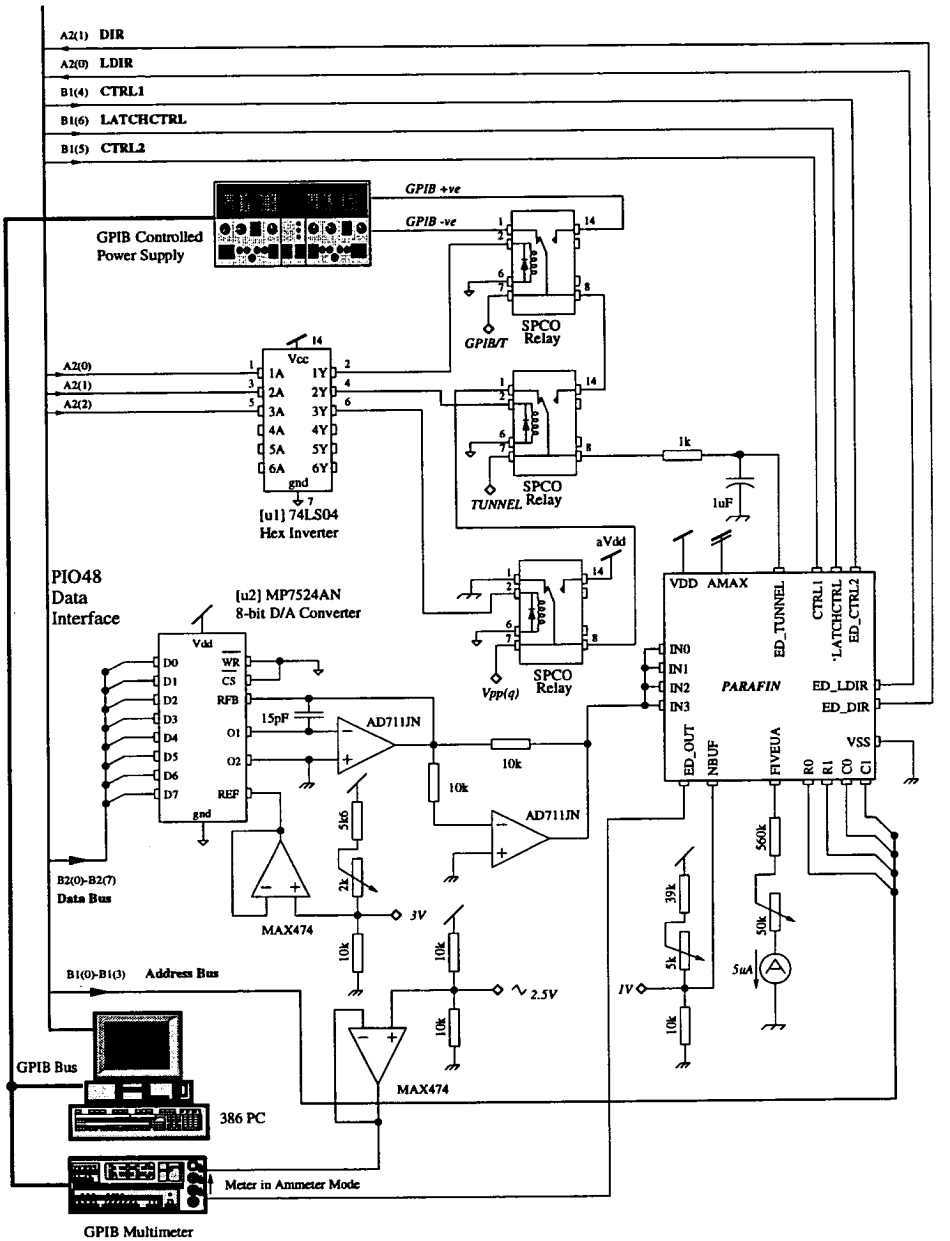


Figure B.10: *PARAFIN* Test Board 1: 4-input Euclidean Distance Cell Array

B.4.2 *PARAFIN* Board 2

PARAFIN board 2 was used for programming the SSYN and DSYN multiplier arrays. The pulse/ramp generation/measurement requirements were very similar to those used on *NEMO* board 2, and therefore this board was modified to become *PARAFIN* board 2 with the replacement of a small number of current and voltage reference sources and the addition of a the extra control logic lines on the PIO interface to the control PC for implementation of the feedback programming mode. This was both quicker and cheaper than building an entirely new board for this purpose.

B.4.3 *PARAFIN* Board 3

The final test board was used with the nonlinearity array. The requirements of an input ramp and output PWM measurement again meant that this board was most easily constructed by redesign of the preceeding board. The complete circuit diagram is shown in figure B.11. The obvious changes are the removal of the input PWM transceivers since there were no pulse-mode inputs. Input was in the form a distance current which was generated by ramping the GPIB PSU across a resistor and measuring the current (distributed to cell 4 inputs) through a multimeter. The ramp biases were changed to cover a new voltage range, and a second D/A converter was added to provide the voltage programming target.

B.5 PC Interfaces

Initial frustrations with *TARDIS* when long experiments would be run incorrectly (for example due to an accidentally disconnected jumper or a wrong parameter in the control software) prompted the use of Borland C's BGI libraries to allow construction of a graphical interface to *NEMO* and *PARAFIN* which would allow continuously updated graphing of experiments' progress and real time status display, allowing immediate treatment of problems, rather than having to wait until the end of the experiment. Some screen-shots of the graphical intefaces are shown in figure B.12.

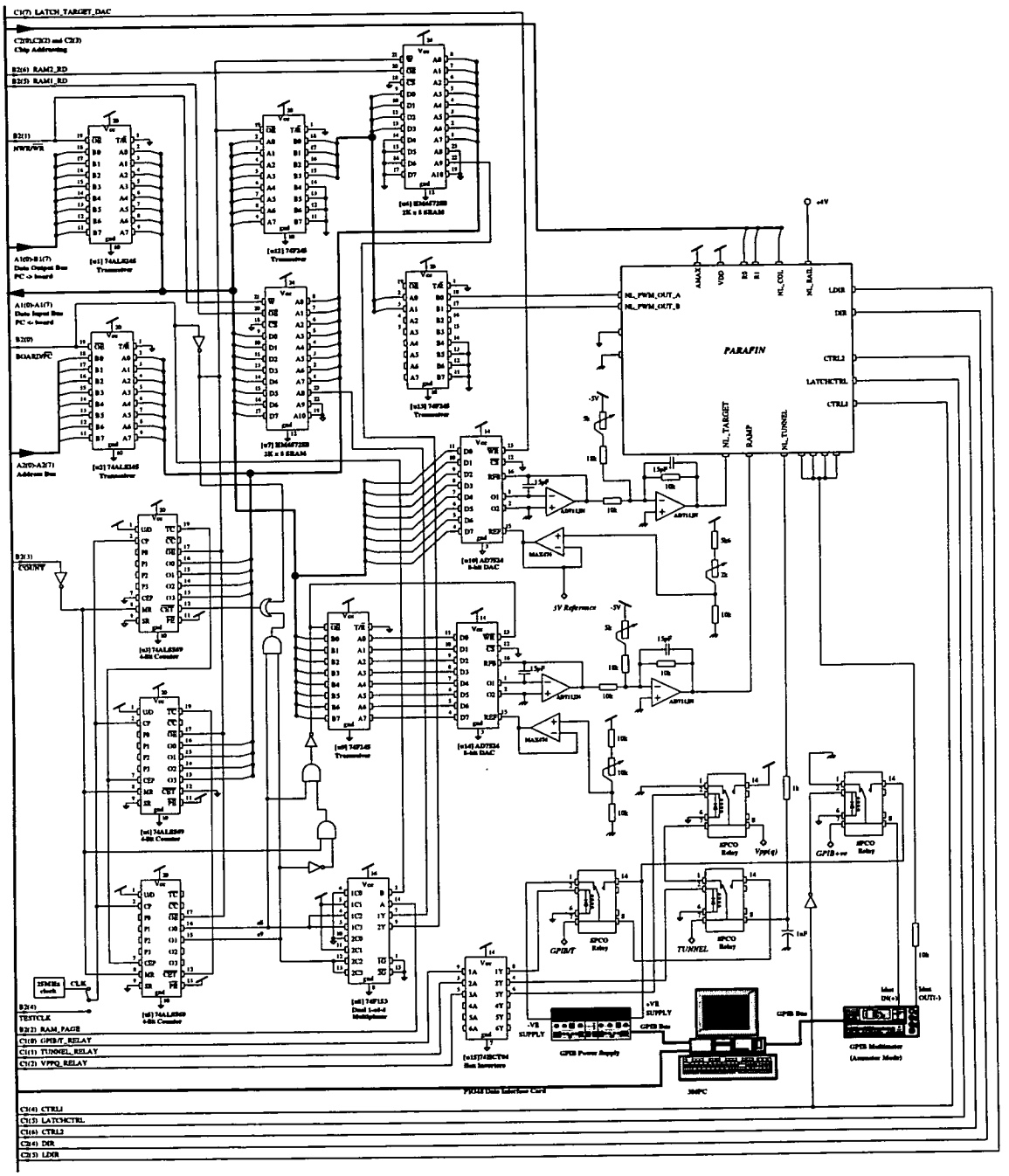


Figure B.11: *PARAFIN* Test Board 3: Nonlinearities

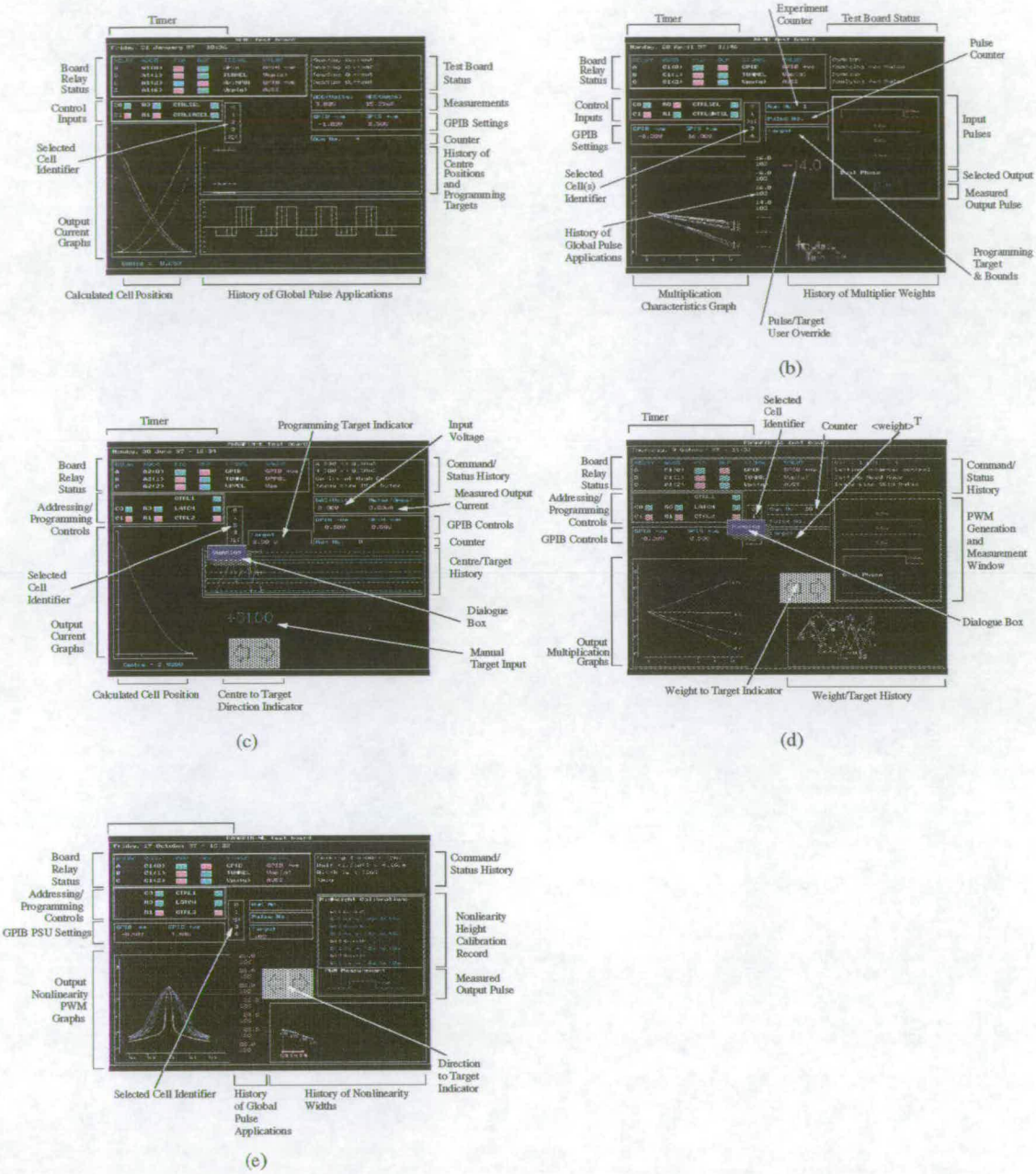


Figure B.12: BGI inteface to test-board control and measurement software: (a) *NEMO* Dual-Phase Euclidean Array, (b) *NEMO* Dual-Phase Multiplier Array, (c) *PARAFIN* Euclidean Array and (d) *PARAFIN* Euclidean Array, (e) *PARAFIN* Nonlinearity Array

Appendix C

Dickson Charge Pump

Many modern EPROM, EEPROM and Flash memory ICs are available with an on-board charge pump. These circuits allow the high voltages required for programming the memories to be generated from the chip power supply (eg. 2.7V, 3.3V or 5V) and so eliminate the need for a special high voltage supply and V_{PP} pin. For the prototyping purposes of *TARDIS*, *NEMO* and *PARAFIN*, it was more convenient, in a laboratory situation, to directly provide high voltages rather than use charge pumps. However these circuits may be very useful to further integration efforts and so will be described briefly in this appendix.

The charge pump circuit proposed by Dickson in 1976 [51] was based on improvements to the much earlier Cockcroft-Walton voltage multiplier [38] to make it more amenable to VLSI integration. The form of the circuit is shown in figure C.1. It comprises a diode chain with each alternate internal node capacitively coupled to the clock, CLK, or its anti-phase, $\overline{CLK}$. The high voltage is developed at node *out* with, typically, the supply voltage applied to node *in*.

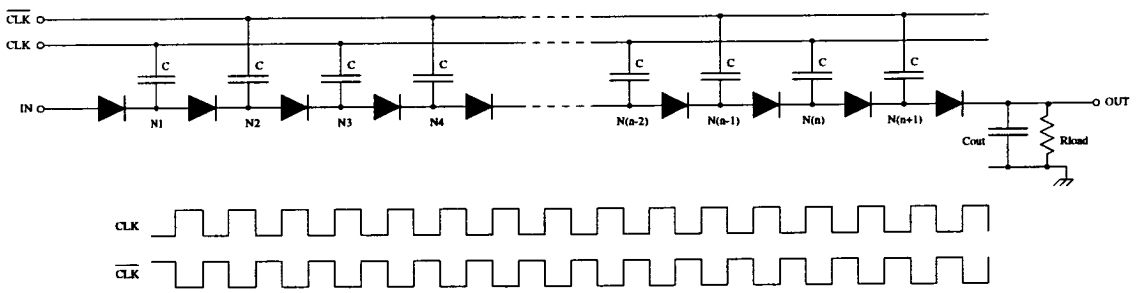


Figure C.1: Circuit schematic of Dickson charge pump

When the supply voltage is first applied, charge propagates along the diode chain, causing each node to charge to a potential one forward-bias diode drop, V_D, lower than the proceeding node.

When the anti-phase clocks start to pulse, each node, say n, is alternately capacitively pulled up whilst both its neighbours, $n-1$ and $n+1$, are capacitively pulled down. For any reasonable coupling coefficient and clock voltage magnitude, this ensures that node n goes to a higher voltage than the falling nodes, $n-1$ and $n+1$. The diodes then act as one-way valves: node n cannot discharge to node $n-1$ because this diode is reverse biased. However it can discharge to node $n+1$ ensuring that the *minimum*

voltage of node $n+1$ is never more than V_D less than the *maximum* voltage of node n. In this way the maximum voltage of each node is always significantly higher than that of the preceding node.

Equilibrium is achieved when the maximum voltage becomes exactly V_D higher than the minimum voltage of the subsequent node, and no further current need enter the chain from node *in* (assuming open-circuit load). The smoothed output at node *out* is then

$$V_{out} = V_{in} + m \cdot V'_{CLK} - (m+1) \cdot V_D \tag{C.1}$$

where V'_{CLK} is the voltage swing at each node due to capacitive coupling from the clock. m is the number of internal nodes, $m+1$ is the total number of diodes in the chain. Thus the maximum voltage is dependent upon the number of stages in the chain. There is, in principle, no limit to the length of the chain. However the silicon area makes it much too costly to associate one charge pump with each floating gate (Cataldo and Palumbo [23] have developed a dynamic model of the charge pump to allow optimisation of chip area or charge time). Therefore charge pumping is an adjunct but not a replacement for on-chip high voltage addressing.

There is a charge-up time associated with the charge-pump (this is the time delay observed whilst waiting for camera flash units to power up), and also a recovery period if the output is discharged during application of the high voltage. Figure C.2 shows the charging behaviour of a 5-diode Dickson charge pump. The clocks had a 5V magnitude, a 2μs period and 50% duty cycle. Parasitic capacitance at each node was 20fF.

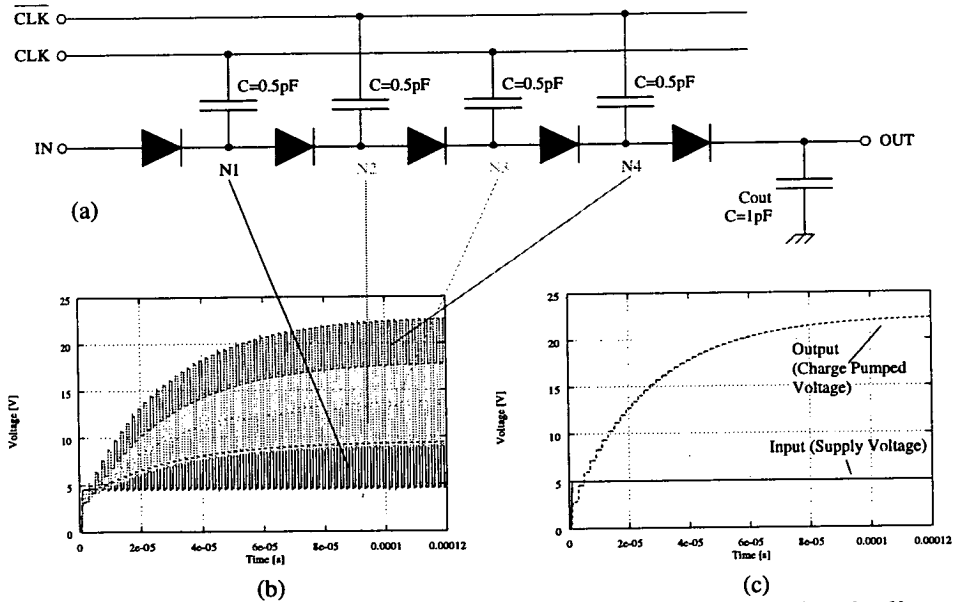


Figure C.2: `Hspice` simulation of Dickson charge pump: (a) Circuit diagram, (b) Internal nodes and (c) Input-Output behaviour.

Appendix D

Common CMOS Failure Modes

D.1 Introduction

This appendix contains a brief description of some of the common CMOS technology failure mechanisms which had to be considered during the design of the circuits discussed in this thesis.

D.2 Trap Up

Charge travelling through insulating oxides (injected by tunnelling, hot electron injection or UV barrier lowering) can become trapped at defects within the oxide resulting in localised distortions of the electric field. Trapping of charge on its own does not normally cause abrupt device failure but can cause long term drift in transistor thresholds and window closure in EEPROM cells therefore it is apt to describe this behaviour as a failure mechanism.

Trap sites are considered to be 'dangling bonds' in the chemical structure of the MOS device. These traps may exist at the surface of the gate, at the gate-insulator interface and also within the insulator itself. The actual distribution of trapped charge between these different layers has so far not been definitively resolved.

Three types of *built-in* traps are believed to exist [146] as a result of the MOS fabrication:

1. **Intrinsic oxide traps:** many types of oxide irregularities have been proposed as potential trap sites. Such irregularities are formed during the manufacture of the oxide.

2. **Intrinsic interface traps:** at the interface crystalline silicon meets amorphous silicon dioxide[1], the configurations of potential dangling bonds become even more complex.

3. **Impurity traps:** impurities introduced during processing may become substituted for silicon or oxygen atoms in the lattice, and may form traps due to the different electronic composition of their valence bands.

Further *generated* traps may be created during circuit operation, for example by

[1] In contrast to the rigid crystal lattice of silicon, amorphous material is without a regular large-scale structure

- **Rupturing of strained intrinsic bonds**[2] This may happen due to the scattering influence of energetic electrons due to the high electric field in the oxide [146].
- **Impact ionisation** [6].
- **Mobile holes** in the oxide re-combining with electrons in Si-H or SiO-H bonds leading to the liberation of hydrogen[3] or in Si-O bonds causing the loosening of the SiO_2 network [165]. Bond breaking tends to begin at the anode where electrons gain $\sim$ 3eV of energy dropping between the oxide and silicon conduction bands.

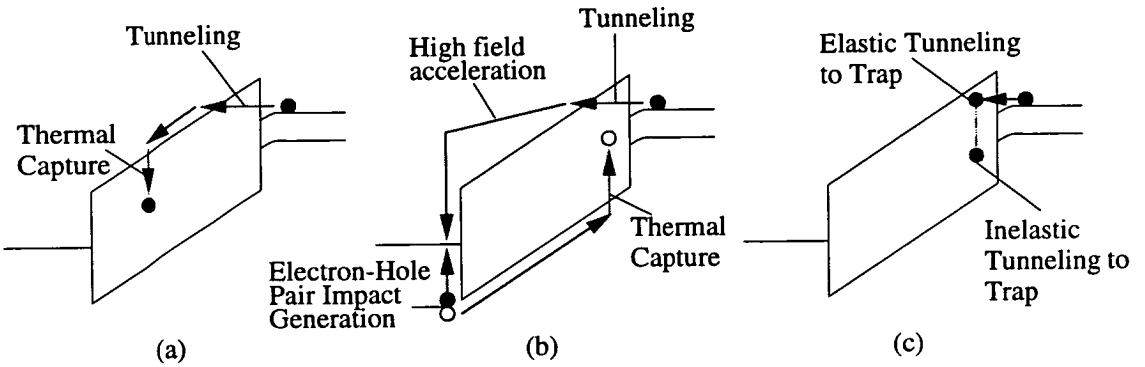


Figure D.1: A selection of trapping mechanisms: (a) thermal electron trapping, (b) trapping of impact-liberated holes (c) direct tunnelling to trap site.

When an electron or hole is trapped by a dangling bond, the trap becomes charged and distorts the local electric field. Trapped holes distort the internal field shape such as to *reduce* the required *applied* electric field whereas trapped electrons require an *increase* in the applied electric field. Some possible trapping mechanisms are:

- Thermal trapping of electrons which have tunnelled into the oxide conduction band from the cathode (figure D.1(a)).
- Thermal trapping of holes injected into the oxide valence band due to energetic impact of tunnelling electrons which have been accelerated by the high electric field between the tunnel-point and the anode (figure D.1(b)).
- Direct tunnelling of electrons between the cathode and the trap site (figure D.1(c)).

[2]The electronic strength of bonds may be less than that determined for isolated molecules in isolation due to the influence of neighbouring atoms and contortion in the lattice: thus bonds may become 'strained' and more easily broken thus forming dangling bond trap sites.

[3]Hydrogen is introduced into the oxide as a by-product during manufacturing by water vapour (wet) oxidation techniques [52].

Thermal trapping is so called because the energy level of trap sites are below that of the conduction band. Electrons which become trapped must therefore release some energy in thermal photon emission.

De-trapping also occurs to a much lesser extent. Here trapped charge is thermally liberated from trap sides. This results in a random fluctuation in V_{fe} and I_{tunnel} although the *trend* is always strongly upwards (trapping much more prevalent than de-trapping).

D.3 Oxide Breakdown

The breakdown of insulating oxide dielectrics is a cause of abrupt failure of ICs. The 'dielectric strength', which shows a strong statistical dependency, is assessed by manufactures in two main ways using a population of test capacitors:

1. Time Zero Dielectric Breakdown (TZDB): the electric field required for dielectric breakdown under the condition of the application of a voltage ramp across the capacitor.

2. Time Dependent Dielectric Breakdown (TDDB): the time required for dielectric breakdown under the condition of either continuous voltage or continuous current (ie. tunnelling current) stressing across the capacitor.

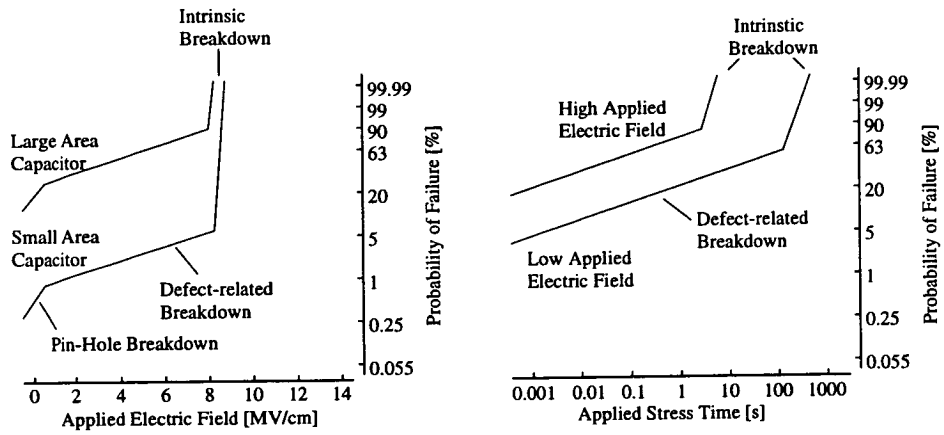


Figure D.2: (a) Extreme-value statistics sketch of TZBD behaviour. (b) Extreme-value statistics sketch of TDDB behaviour. Both sketches are based on real data plots presented in [175].

Typical extreme value statistics breakdown plots are sketched in figure D.2. Breakdown may be placed in three categories [183] which are evident from distinct regions of the plots:

1. *Pin-Hole Breakdown*: Breakdown which occurs at very low electric field due to 'pin-holes' through the oxide. This breakdown region does not appear in TDDB plots due to their almost instantaneous failure.

2. *Defect-Related Breakdown*

3. *Intrinsic Breakdown*: the gradient of the defect-related breakdown portion of the plot is indicative of the quality of the oxide (intrinsic "dielectric strength").

In fact the TZDB and TDDB plots can be related by the introduction of the concept of *charge to breakdown*, Q_{BD}:

$$Q_{BD} = \int_0^{t_{BD}} I(t)dt \tag{D.1}$$

where t_{BD} is the breakdown time and $I(t)$ is the oxide current. In both TZDB and TDDB experiments the breakdown occurs when Q_{BD} is reached [175, 176].

D.3.1 Intrinsic Oxide Breakdown

Intrinsic oxide breakdown proceeds through two distinct phases:

1. Build-Up

2. Run-Away

During build-up traps or broken bonds are formed by the trap generation mechanisms described in section D.2. Eventually these form a connecting chain of defects within 'weak' regions within the oxide which provides an ohmic path through the oxide [165]. Breakdown then enters the run-away phase leading to electrical and thermal run-away associated with catastrophic failure (although the abrupt and localised nature of this second phase has hindered investigation of its physical explanation [154]). Run-away breakdown in capacitors is generally abrupt [112]. This is because another positive feedback mechanism can take over, quickly passing the full electrostatic energy ($\frac{1}{2}CV^2$) of the capacitor through the defect resulting in sudden irreversible rupture. EEPROM floating gate capacitors tend to be very small and hence have little electrostatic energy so that each cycle can only break a few oxide bonds. In fact bonds in the damaged area are continually broken and re-form causing an erratic leakage and breakdown behaviour.

D.3.2 Defect-Related Oxide Breakdown

Early life (non-intrinsic) breakdown of oxides has been attributed to the presence of oxide or interface defects, surface asperities or metal contamination (which lowers the energy barriers) [165]. These effects can be modelled by effective oxide thinning [100]. This results in an enhanced electric field and therefore an *accelerated* t_{BD} at its weakest

spot. The value of field enhancement has a distribution related to the properties of the defect.

D.4 Electromigration

Electromigration occurs when excessive current flows in a conductor. The actual atoms of the conductor can be carried along in the current leading a thinning or eventual break in electrical conductivity [68]. The problem can be avoided by ensuring that the conductor is wide enough to handle the peak current flow. Typical maximum current densities for the Alcatel-Mietec 2.4μm CMOS process are 640μA/μm for a metal 1 track and 670μA for a single via (large currents require multiple vias).

D.5 Avalanche Breakdown

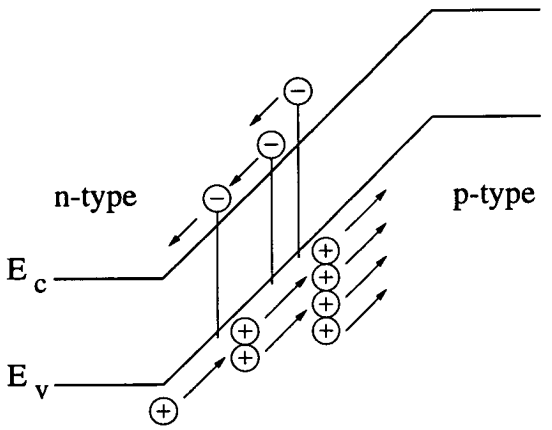


Figure D.3: Avalanche multiplication in the depletion region. A hole entering the high field depletion region impacts with the semiconductor lattice and liberates an electron-hole pair. Both the holes now moving in the field cause impact ionisation leading to four holes moving in the field. This multiplication can continue to very high levels of current.

Avalanche breakdown [184] occurs in reverse biased *pn*-junctions such as explicit diodes or those associated with MOSFET diffusion-substrate junctions. Breakdown happens when the electric field across the depletion region is high enough such that stray charge (reverse bias current - known as the *seed* current) entering the depletion region gains sufficient kinetic energy to cause *impact ionisation* upon collision with the semiconductor lattice. This ionisation releases electron-hole pairs which are also accelerated by the electric field such that they have sufficient energy to also cause impact ionisation. The depletion region current then enters avalanche *multiplication*. This is shown schematically in figure D.3. This process is haltable by reducing the

electric field but becomes breakdown if the large current thus generated causes thermal or electromigration damage.

The width of the depletion region, and consequently the electric field strength is dependent on the dopant density at the *pn* junction. Lightly doped junctions such as the well-substrate junction have higher breakdown voltages. Similarly source/drain breakdown voltages can be increased by use of 'spacers' either side the gate during self-alignment which provide a less abrupt doping profile.

D.5.1 Gate-Field Enhanced Breakdown

The threshold field for avalanche breakdown is reduced by a curvature of the electric field in the proximity of the gate terminal such that it is locally enhanced. This *gate-field enhanced breakdown* [52] means that the drain avalanche breakdown voltage, BV_{ds}, of a MOSFET is lower than that of a freestanding diode.

D.5.2 Snapback

In short n-channel MOSFETs, *snapback* [180] may occur. This is an unstable negative resistance regime sketched in figure D.4. Figure D.4(a) shows the breakdown characteristics of a long n-channel MOSFET. There is a slight positive BV_{ds} dependency such that the breakdown voltage is higher for higher gate voltages. This is because the pinch-off point retreats towards the source at higher gate voltages, broadening the depletion region and reducing the electric field.

In the short n-channel MOSFET, figure D.4(b), for moderate gate voltages, there is a *negative* BV_{ds} dependency on V_g. Moreover, the increase in drain current allows the drain voltage to decrease to a *sustain voltage*, BV_{ds}. This is because in short n-channel MOSFETs a parasitic lateral bipolar *npn*-transistor exists. The avalanching substrate current turns on this bipolar transistor such that it can sustain the avalanche seed current at a lower drain electric field. Snapback rarely occurs in p-channel MOSFETs [162] due to the lower ionisation current, lower hole mobility and lower substrate resistance. For higher gate voltages the gate-induced reduction in drain electric field starts to dominate the behaviour and so BV_{ds} begins to rise again above a minimum at about $V_d \sim 2 - 4$V [180, 177].

D.6 Punchthrough

Punchthrough occurs in very short channel transistors when the drain voltage becomes sufficiently high that the drain depletion regions extends the entire length of the channel, and the gate then loses control over the current [11]. Large currents can flow causing electromigration and thermal damage.

The punchthrough voltage shows a very strong dependency on the channel length [83, 181], and can be avoided by increasing the length by a small amount. Above a cer-

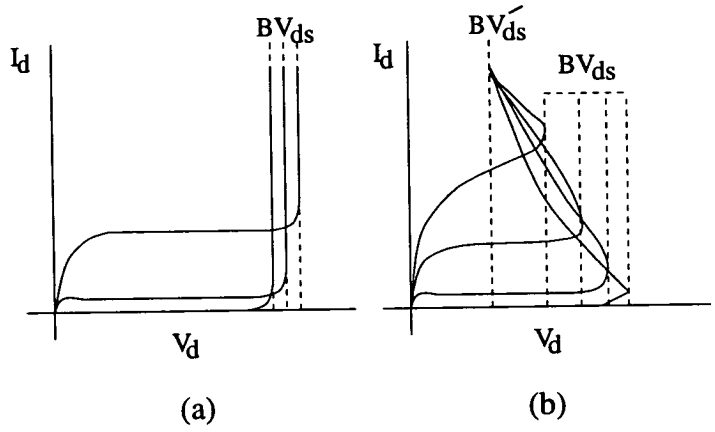


Figure D.4: (a) Breakdown characteristics of long n-channel MOSFET, (b) Breakdown characteristics of short n-channel MOSFET for low gate voltages.

tain length, avalanche breakdown takes over as the dominant drain-voltage dependent breakdown behaviour.

D.7 Latch-Up

Latch-up is caused by parasitic resistors and bipolar transistors which exist as by-products in CMOS designs.

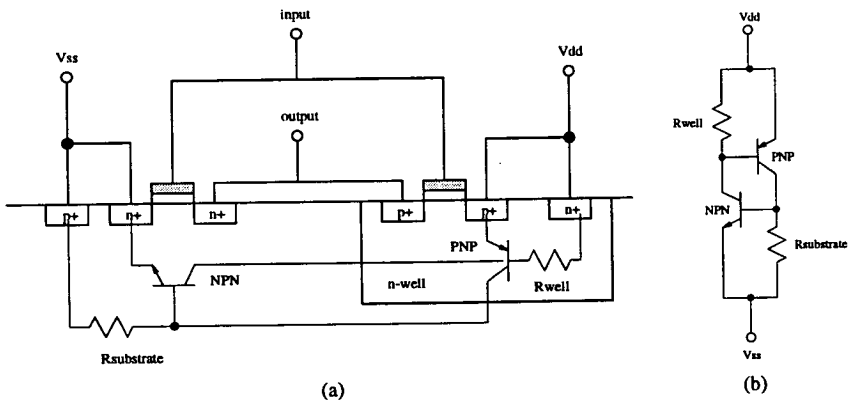


Figure D.5: (a) Cross-section of CMOS inverter with parasitic resistors and bipolar transistors, (b) Extracted parasitic latch-up circuit.

As an example, figure D.5(a) shows the cross-section of a CMOS digital inverter with the parasitic devices which contribute to latch-up. The parasitic circuit is shown extracted in figure D.5(b). Consider current entering the emitter of the PNP transistor. This will cause current to flow in $R_{substrate}$, inducing a potential difference sufficient

to turn on the NPN transistor. This causes a potential drop across R_{well} turning the PNP transistor harder on. Thus the circuit enters feedback operation and is said to *latch-up* with large currents being drawn into the substrate while the supply drops to a 'holding voltage' which maintains the latch-up. Latch-up may occasionally be 'cured' by powering down but again often destructive currents have already caused permanent damage.

Latch-up is initiated by the injection of current into the emitter of one of two parasitic bipolar transistors. To help prevent latch-up, the transistor source and drain voltages must remain within the tapped substrate and well potentials, avoiding closely packed devices and the substrate and well resistances should be minimised by use of multiple, closely-spaced taps or low resistance epitaxial start material if available [170].

D.8 Field Transistors

Field Transistors are formed when a parasitic conducting channel is formed between two unrelated n+ implants due to the gate action of an overpassing metal track. Because of the thick field oxide, the threshold of such transistors is naturally higher. Manufacturers make it higher still by the implantation of a *channel-stop* surface diffusion which raises the impurity concentration away from the active area of desired transistors [170]. Protection from substrate inversion at very high signal voltages may be further enhanced by conveying the high voltage signals in metal 2 above a grounded 'shield' of metal 1 [47]. Field transistors have found useful application in ESD protection (see section D.9).

D.9 Electrostatic Discharge Damage (ESD)

Section D.3 describes how a large oxide electric field can lead to breakdown shorting out the capacitor or transistor of which it forms the dielectric. Of course, high fields are not normally applied deliberately to ICs but can arise through *triboelectricity*; a person walking across a room can easily charge themselves to several kilovolts by friction of materials. The rapid discharge of this person into a gate capacitance will easily generate sufficient electric fields to destroy the oxide. This failure mechanism is known as electrostatic discharge (ESD) and is a serious problem with all CMOS technology. Due to the irreversible nature of this damage pad circuits are normally employed to prevent high voltages and currents from reaching the core. Discharge currents are instead diverted to power lines if appropriate or dissipated in the substrate (although care must be taken that this does not induce latch-up).

A selection of ESD protection circuits are shown in figure D.6:

- Back-to-back diodes limit the input to $\pm$ a diode threshold of the power-rails as they forward-bias outwith these limits.

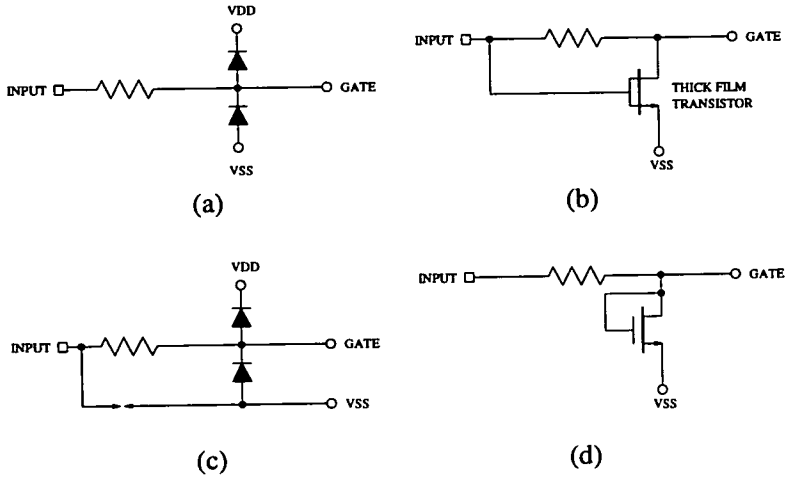


Figure D.6: ESD protection circuits: (a) Back-to-back diodes, (b) Thick film pull-down transistor, (c) Back-to-back diodes and spark gap, (d) Punchthrough transistor.

- A thick film transistor is formed by running a metal strip across active area; this transistor has a high threshold voltage (but below the gate oxide breakdown) and so can pull down high input voltages.
- A spark gap is simply a very narrow gap which provides another path to ground.
- The punchthrough transistor is a short channel MOSFET which punches through (and pulls down) above a certain input voltage.

All the designs incorporate a resistor to limit the peak current flow with may otherwise be several amps. Using anti-static mats and grounding straps when touching ICs can also help reduce the extent of the problem.

Appendix E

Bake Retention Testing

E.1 Introduction

This appendix briefly describes the rationale and practice of bake retention testing which is routinely conducted on digital EEPROM designs to test their ability to retain programmed data.

E.2 Thermionic Emission Theory

Since floating gate insulating oxides tend to be thick enough to make any leakage tunnelling currents insignificant at low operating voltages[1], the most significant mechanism is due to thermal emission (Richardson emission) of electrons over the top of the Si-SiO_2 potential barrier.

E.2.1 Boltzmann Distribution of Electrons

Electrons in a semiconductor exist in a number of energy states related to the atomic structure and temperature. The Fermi-Dirac distribution describes the probability distribution of electrons in these energy states [184]. For energy levels above $\sim 3kT$, this may be approximated by the Maxwell-Boltzmann distribution law. This defines a system in which n_1 particles with energy E_1 and n_2 particles with energy E_2 may be described by the relationship [186]

$$n_1 = n_2 \exp\left[-\left(E_1 - E_2\right)/kT\right] \tag{E.1}$$

n_E may then be defined as the number of electrons at energy state E as

$$n_E = n_0 \exp\left(-E/kT\right) \tag{E.2}$$

where n_0 is the number of electrons at the zero energy state. It is then possible to define the distribution function, $f(E)$, as

$$f(E) = \frac{n_E}{n_{total}} = \frac{n_0 \exp\left(-E/kT\right)}{n_0 \int_0^\infty \exp\left(-E/kT\right) dE} = \exp\left(-E/kT\right)/kT \tag{E.3}$$

[1]Fowler-Nordheim tunnelling does show a weak temperature dependency and is so is slightly stronger if the floating gate is operated at elevated temperatures [132]

E.2.2 Thermionic Charge Loss

Nozawa and Kohyama [127] use this distribution to explain charge retention in SAMOS floating gate structures. Electrons with energy $> \phi_B$ (greater than the interface barrier between floating gate and insulating oxide) are emitted over the barrier and so are swept away and lost from the floating gate (see figure E.1). Distribution equilibrium is restored by a relaxation mechanism.

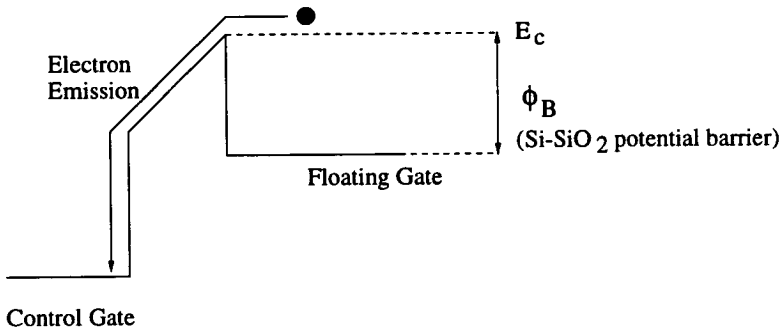


Figure E.1: Schematic energy diagram of charge loss in floating gate system with negative ϕ_{fg} (ie. negative bias on the floating gate with respect to the control gate). Note that provided the field is such to sweep charge away from the floating gate, the actual value of ϕ_{fg} and other biases is irrelevant.

Therefore, if the number of electrons at energy state E is $n(t).f(E)$ where $n(t)$ is the total number of electrons on the floating gate, then

$$-\frac{dn(t)}{dt} = \nu n(t) \int_{\phi_B}^{\infty} f(E)dE = \nu n(t) \exp\left(-\phi_B/kT\right) \tag{E.4}$$

where ν is the electron-lattice collision frequency. The solution of equation E.4 is then

$$n(t) = n(0) \exp\left\{-\nu t \exp\left(-\phi_B/kT\right)\right\} \tag{E.5}$$

which defines the number of electrons on the floating gate at a given time.

E.2.3 Arrhenius Model

Equation E.5 is sufficient to apply the thermionic emission model to charge loss estimation using experimental data to fit ν and ϕ_B. If the specific time-to-failure (TTF) – eg. the time until charge loss is sufficient to flip an EEPROM bit – then this relationship may be abstracted by the Arrhenius equation

$$R = R_0 \exp\left(-E_a/kT\right) \tag{E.6}$$

This equation was developed to describe the temperature dependency of the rate, R, of chemical reactions. It has since been adapted to many electronics situations which also exhibit a temperature dependency [156].

The thermionic emission expression in equation E.4 maps into the Arrhenius model with reaction rate $R = \frac{-dn(t)}{dt}$, $R_0 = \nu n(0)$ and the process activation energy (usually expressed in electron-volts), $E_a = \phi_B$. Thus the Arrhenius model is commonly used to measure the intrinsic charge retention properties of commercial floating gate devices such as EEPROMs.

Since the TTF, t_F, is inversely proportional to the reaction rate, R. Then for two different temperatures, T_1 and T_2 the times to failure, t_1 and t_2 relate according to

$$\begin{aligned} \frac{t_1}{t2} = \frac{R_2}{R_1} &= \frac{R_0 \exp\left(-E_a/kT_2\right)}{R_0 \exp\left(-E_a/kT_1\right)} \\ &= \exp\left[\frac{E_a}{k}\left(\frac{1}{T_1} - \frac{1}{T_2}\right)\right] \end{aligned} \tag{E.7}$$

Therefore the TTF at one temperature may be used to predict the time to failure at another:

$$\begin{aligned} t_F &= t_2 \exp\left[\frac{E_a}{k}\left(\frac{1}{T_F} - \frac{1}{T_2}\right)\right] \\ &= t_2 \exp\left(\frac{-E_a}{kT_2}\right) \exp\left(\frac{E_a}{kT_F}\right) \end{aligned} \tag{E.8}$$

Therefore a linear plot in logarithmic time may be developed

$$\ln\left(t_F\right) = const_1 + \frac{const_2}{T} \tag{E.9}$$

Thus it is not necessary to know the electron-lattice collision frequency, the number of electrons on the floating gate or the Si-SiO_2 barrier height, since these constants can be fitted to a straight-line (Arrhenius plot).

Activation energies determined experimentally have been found to be anything between 1.24eV [127] and 1.8eV [128] with the apparent disparity with the work function, ϕ_B, of the Si-SiO_2 interface (3.2eV) due to a complex behaviour of polarisation arising out of electron cloud shift [128].

E.3 Commercial EEPROM Testing

Commercial digital EEPROMs are commonly tested for retention with the Arrhenius model using "Bake Retention". This involves programming a topological checkerboard pattern onto a chip, monitoring a continuous READ cycle at elevated temperatures (about 125°C is typical) and recording the time until the first bit flips. The Arrhenius model is then used to extrapolate the time-to-failure (Failure Units - FITs -

Process	E_a
Oxide Breakdown	0.30eV
Metal Electromigration	0.55eV
Oxide Defect	0.60eV
Silicon Electromigration	0.90eV
Ionic Contamination	1.00eV

Table E.1: Typical failure mechanisms and corresponding activation energies [182]

are often used; these refer to number of cell failures projected for 10^9 device-hours).

Defect free chips will have thermionic storage times of many years. Earlier chip failures generally tend to be due to individual cells (the weakest bit) failing due to isolated defects which have lower activation energies than ϕ_B; a selection of these are shown in table E.1[2]. In practice EEPROM retention tends to be limited by defect densities rather than intrinsic oxide characteristics, with problems manifest as poor retention (leaks) rather than unprogrammable cells [112].

A second issue is *endurance* which measures the number of programming cycles an EEPROM cell can endure before it either becomes unprogrammable (due to aging) or has a significant loss of retention capabilities due to aggravation of latent defects or TDDB. Bake retention tests are therefore usually repeated on cycled EEPROMs. Cycled EEPROMs (particularly textured polysilicon EEPROMs) will contain trapped oxide charge (electrons or holes) which may become thermally liberated during bake retention and either add to or neutralise floating gate charge [164] thus weakening the result. Since the trapped charge is finite in extent this behaviour is short-term and can be measured during prolonged bakes.

[2]Whilst the Fermi-Dirac distribution provides a sound physical basis for the application of the Arrhenius model to thermionic emission across the Si-SiO_2 barrier, it is unclear how well the behaviour of these other failure mechanisms correspond to this model.

Appendix F

Publications and Articles

F.1 Publications

The following articles were published externally during the course of this work:

1. L. W. Buchan, A. F. Murray and H. M. Reekie, "Floating Gate Memories for Pulse-Stream Neural Networks", *Electronics Letters*, vol. 33 no. 5 (1997), pp. 397 – 399.

2. L. William Buchan, Alan F. Murray and H. Martin Reekie, "Standard CMOS Floating Gate Memories for Non-Volatile Weight Storage in Analogue VLSI Neural Networks", In *Proceedings of the 15th IMACS World Congress*,vol. 6,pp. 511-516, Berlin,1997.

F.2 Other Articles

The following articles have not been published externally:

1. L. W. Buchan, A. F. Murray and H. M. Reekie, "Floating Gate Memories for Pulse-Stream Neural Networks", *PhDEE* (Edinburgh University Department of Electrical Engineering Internal Postgraduate Journal), issue 3, April 1997

2. L. W. Buchan, A. F. Murray and H. M. Reekie, "Pulse-Stream Neural Sensor Data Processors With Non-Volatile Weight Storage", Accepted for publication in *Proceedings of IEEE Workshop on Neural Networks for Signal Processing 1997* but withdrawn due to attendance difficulties

3. Alan F. Murray and Bill Buchan, "A User's Guide to Non-Volatile, On-Chip Analogue Memory", Submitted for consideration for possible future external publication in the Electronics and Communications Engineering Journal.

References

[1] Å. Abusland and T. S. Lande. Local Generation and Storage of Reference Voltages in CMOS Technology. In *Proceedings of ECCTD '93, 11th European Conference on Circuit Theory and Design*, volume 1, pages 281 – 286, 1993.

[2] Y. Aihara, M. Kodama, K. Nakahara, H. Okise, and K. Murata. Characteristics of a Thin Film Lithium-Ion Battery using Platicized Solid Polymer Electrolyte. *Journal of Power Sources*, 65(1 - 2):143 – 147, 1997.

[3] Phillip E. Allen and Douglas R. Holberg. *CMOS Analog Circuit Design*. Holt, Rinehart and Winston, Inc., New York (USA), 1987.

[4] U. Apel, E. Habekotté, and B. Höfflinger. High-Voltage Transistors in a Low-Voltage CMOS Process. In *Proceedings of ISSSE '89, International Symposium on Signals, Systems and Electronics*, Erlangen (Germany), September 1989.

[5] Yutaka Arima, Koichiro Mashiko, Keisuke Okada, Tsuyoshi Yamada, Atsushi Maeda, Harufusa Kondoh, and Shimpei Kayono. A Self-Learning Neural Network Chip with 125 Neurons and 10K Self-Organisation Synapses. *IEEE Journal of Solid-State Circuits*, 26(4):607 – 611, April 1991.

[6] D. Arnold, E. Cartier, and D. J. DiMaria. Theory of High-Field Electron-Transport and Impact Ionization in Silicon Dioxide. *Physical Review B*, 49(15):10278 – 10297, April 1994.

[7] K. Asanovic and N. Morgan. Experimental Determination of Precision Requirements for Backpopagation Training of Artificial Neural Networks. In *Proceedings of Microneuro '92, International Conference on Microelectronics for Neural Networks*, pages 9 – 15, 1992.

[8] M. Av-Ron, T. H. DiStefano M. Shatzkes, and R. A. Gdula. Electron-Tunneling at Al-SiO_2 interfaces. *Journal of Applied Physics*, 52, 1981.

[9] B. Jayant Baliga. An Overview of Smart Power Technology. *IEEE Transactions on Electron Devices*, 38(7):1568 – 1575, July 1991.

[10] H. Ballan, M. Declercq, and F. Krummenacher. Design and Optimization of High Voltage Analog and Digital Circuits Built in a Standard 5V CMOS Technology. In *Proceedings of CICC '94, IEEE Custom Integrated Circuits Conference*, pages 574 – 577, San Diego (USA), May 1994.

[11] John J. Barnes, Katsuhiro Shimohigashi, and Robert W. Dutton. Short-Channel MOSFET's in the Punchthrough Current Mode. *IEEE Transactions on Electron Devices*, ED-26(4):446 – 453, April 1979.

[12] F. H. Behrens, S. Finco, and M. I. Castro Simas. Medium-Voltage Lateral NMOS Power Devices in Standard CMOS Technology. In *Proceedings of EPE '93, 5th European Conference on Power Electronics and Applications*, Brighton (UK), September 1993.

[13] Y. Berg, R. L. Sigvartsen, T. S. Lande, and Å. Abusland. An Analog Feedforward Neural-Network with On-Chip Learning. *Analog Integrated Circuits and Signal Processing*, 9(1):65 – 75, January 1996.

[14] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford (UK), 1995.

[15] D. R. Brown, S. Collins, and G. F. Marshall. Carrier Trapping in Inter-Polysilicon Charge Injectors. *Electronics Letters*, 31(1), January 1995.

[16] Paul B. Brown, Ronald Millecchia, and Michael Stinely. Analog Memory for Continuous-Voltage Discrete-Time Implementation of Neural Networks. In *Proceedings of ICNN '87, International Conference on Neural Networks*, volume 3, pages 523 – 530, 1987.

[17] G. Cairns. *Learning with Analogue VLSI Multi-Layer Perceptrons*. PhD thesis, Oxford University, 1995.

[18] A. Caiti, G. Canepa, D. De Rossi, F. Germagnoli, G. Magenes, and T. Parisini. Towards the Realization of an Artificial Tactile System: Fine-Form Discrimination by a Tensorial Tactile Sensor Array and Neural Inversion Algorithms. *IEEE Transactions on Systems, Man and Cybernetics*, 25(6), June 1995.

[19] Stephen A. Campbell. *The Science and Engineering of Microelectronic Fabrication*, chapter 11. Oxford University Press, Inc., New York (USA), 1996.

[20] H. C. Card, C. R. Schneider, and R. S. Schneider. Learning Capacitive Weights in Analog CMOS Neural Networks. *Journal of VLSI Signal Processing*, 8(3):209 – 225, 1994.

[21] L. Richard Carley. Trimming Analog Circuits Using Floating-Gate Analog MOS Memory. *IEEE Journal of Solid-State Circuits*, 24(6):1569 – 1575, December 1989.

[22] R. Castello, D. D. Caviglia, M. Franciotta, and F. Montecchi. Selfrefreshing Analogue Memory Cell for Variable Synaptic Weights. *Electronics Letters*, 27(20):1871 – 1872, September 1991.

[23] Giuseppe Di Cataldo and Gaetano Palumbo. Double and Triple Charge Pump for Power IC: Dynamic Models Which Take Parasitic Effects into Account. *IEEE Transactions on Circuits and Systems - I: Fundamental Theory and Applications*, 40(2):92 – 101, February 1993.

[24] Gert Cauwenberghs. An Analog VLSI Recurrent Neural Network Learning a Continuous-Time Trajectory. *IEEE Transactions on Neural Networks*, 7(2):346 – 361, March 1996.

[25] Gert Cauwenberghs and Amnon Yariv. Fault-Tolerant Dynamic Multilevel Storage in Analog VLSI. *IEEE Transactions on Circuits and Systems - II: Analog and Digital Signal Processing*, 41(12), December 1994.

[26] Y.-Y. Chai and L. G. Johnson. Floating Gate MOSFET with Reduced Programming Voltage. *Electronics Letters*, 30(18), September 1994.

[27] Y.-Y. Chai and L. G. Johnson. A 2 × 2 Analog Memory Implemented with a Special Layout Injector. *IEEE Journal of Solid-State Circuits*, 31(6):856 – 859, June 1996.

[28] S. Chen, C. F. N. Cowan, and P. M. Grant. Orthogonal Least Squares Algorithm for Radial Basis Function Networks. *IEEE Transactions on Neural Networks*, 2(2):302 – 309, March 1991.

[29] Sheng Chen, Bernard Mulgrew, and Peter M. Grant. A Clustering Technique for Digital Communication Channel Equalization Using Radial Basis Function Networks. *IEEE Transactions on Neural Networks*, 4(4):570 – 579, July 1993.

[30] Jeff Child. EPROM Stretches to 10M Erase/Writes. *Computer Design*, 34(3):132, March 1995.

[31] Jeff Child. Flash to Kill EPROM Market by 2000. *Computer Design*, 35(3):73 – 75, February 1996.

[32] Jeff Child. Flash-Write Voltages Head Down Two Paths. *Computer Design*, 35(3):76, February 1996.

[33] S. Churcher, D. J. Baxter, A. Hamilton, A. F. Murray, and H. M. Reekie. Generic Analog Neural Computation - The EPSILON Chip. In *Proceedings of NIPS '93, Neural Information Processing Systems Conference*, pages 773 – 780. Morgan Kaufmann, 1993.

[34] S. Churcher, D. J. Baxter, A. Hamilton, A. F. Murray, and H. M. Reekie. The EPSILON Chip - An Analogue VLSI Neural Net Building Block. In *Proceedings of Microneuro '93, the International Conference on Microelectronics for Neural Networks*, pages 217–225, Edinburgh (UK), 1993.

[35] S. Churcher, A. F. Murray, and H. M. Reekie. Programmable Analogue VLSI for Radial Basis Function Networks. *Electronics Letters*, 29(18):1603 – 1604, September 1993.

[36] Stephen Churcher. *VLSI Neural Networks for Computer Vision*. PhD thesis, The University of Edinburgh, 1993.

[37] Lawrence T. Clark, Robert O. Grondin, and Sandwip K. Dey. Integrated Circuit Neural Networks using Ferroelectric Analog Memory. In *Proceedings of IPCCC '92, 11th International Phoenix Conference on Computers and Communications*, pages 736 – 742, 1992.

[38] J. D. Cockcroft and E. T. Walton. Production of High Velocity Positive Ions. *Proceedings of the Royal Society of London - Series A*, 136:619 – 630, 1932.

[39] S. Collins, G. F. Marshall, and D. R. Brown. An Analogue Radial Basis Function Circuit using a Compact Euclidean Distance Calculator. In *Proceedings of ISCAS '94, IEEE International Symposium on Circuits and Systems*, volume 6, pages 233 – 236. IEEE, 1994.

[40] Dallas Semiconductor Corp. High-Density, High-Performance NV SRAM Features Large 2M x 8 Memory Array. WWW page: http://www.dalsemi.com/News_Center/Press_Releases/1997/pr1270.html, August 1997. Dallas Semiconductor Corp., Dallas (USA).

[41] Simtek Corp. Common nvSRAM Questions. WWW page: http://www.simtek.com/simtek/docs/TechNotes.html, June 1997. Simtek Corp., Colorado Springs (USA).

[42] Intel Corporation. 80170NX Electrically Trainable Analog Neural Network, June 1991. Order Number: 290408-002.

[43] Intel Corporation. WWW page: http://developer.intel.com/design/news/strata.htm, September 1997.

[44] Intel Corporation. WWW page: http://www.intel.com/pressroom/archive/releases/fl091797.htm, September 1997.

[45] K. W. Current. CMOS Quaternary Latch. *Electronics Letters*, 25(13):856 – 858, June 1989.

[46] C. Das. MIETEC 2μm CMOS MPC - Electrical Parameters. Technical Report MIE/F/02 Revision 3, EUROCHIP Service Organisation, December 1994.

[47] M. Declercq, F. Clement, M. Schubert, A. Harb, and M. Dutoit. Design and Optimization of High-Voltage CMOS Devices Compatible With A Standard 5V CMOS Technology. In *Proceedings of CICC '93, IEEE Custom Integrated Circuits Conference*, San Diego (USA), May 1993.

[48] Michel Declercq, Hussein Ballane, André Tuor, Bertrand Hochet, and François Clément. High-Voltage CMOS Devices and Circuits Compatible with a Standard CMOS Technology, June 1994. Research Abstract.

[49] Michel J. Declercq, Martin Schubert, and François Clement. 5V-to-75V CMOS Output Interface Circuits. In *Proceedings of ISSCC '93, IEEE International Solid-State Circuits Conference*, pages 162 – 163, & 283, San Fransisco (USA), February 1993.

[50] F. Devos, M. Zhang, Y. Ni, and J.-F. Pône. Trimming CMOS Smart Imager with Tunnel-Effect Nonvolatile Analogue Memory. *Electronics Letters*, 29(20):1766 – 1767, September 1993.

[51] John F. Dickson. On-Chip High-Voltage Generation in MNOS Integrated Circuits Using an Improved Voltage Multiplier Technique. *IEEE Journal of Solid-State Circuits*, SC-11(3):374 – 378, June 1976.

[52] Thomas E. Dillinger. *VLSI Engineering*. Prentice-Hall, Englewood Cliffs (USA), 1988.

[53] C. Diorio, P. Hasler, B. A. Minch, and C. A. Mead. A Single-Transistor Silicon Synapse. *IEEE Transactions on Electron Devices*, 43(11):1972 – 1980, 1996.

[54] C. Diorio, S. Mahajan, P. Hasler, B. Minch, and C. Mead. A High-Resolution Nonvolatile Analog Memory Cell. In *ISCAS '95, IEEE International Symposium on Circuits and Systems*, pages 2233–2236, 1995), Seattle, Washington (USA), May 1995.

[55] Chris Diorio, Paul Hasler, Bradley A. Minch, and Carver Mead. A complementary pair of four-terminal silicon synapses. *Analog Integrated Circuits and Signal Processing*, 13(1 – 2):153 – 166, 1997.

[56] James Donald and Lex A. Akers. A Neural Processing Node with On-Chip Learning. In *Proceedings of ISCAS '93, International Symposium on Circuits and Systems*, volume 4, pages 2748 – 2751, Chicago (USA), 1993.

[57] D. A. Durfee and F. S. Shoucair. Comparison of Floating Gate Neural Network Memory Cells in Standard VLSI CMOS Technology. *IEEE Transactions on Neural Networks*, 3(3):347 – 353, May 1992.

[58] D. A. Durfee and F. S. Shoucair. Low Programming Voltage Floating Gate Analogue Memory Cells in Standard VLSI Technology. *Electronics Letters*, 28(10), May 1992.

[59] Silvio P. Eberhardt, Raoul Tawel, Timothy X. Brown, Taher Daud, and Anilkumar P. Thakoor. Analog VLSI Neural Networks: Implementation Issues and Examples in Optimization and Supervised Learning. *IEEE Transactions on Industrial Electronics*, 39(6):552 – 564, December 1992.

[60] Peter J. Edwards and Alan F. Murray. *Analogue Imprecision in MLP Training*. World Scientific Publishing Co. Pte. Ltd, Singapore, 1996.

[61] R. H. Fowler and L. Nordheim. Electron Emission in Intense Electric Fields. *Proceedings of the Royal Society of London - Series A*, 119:173 – 181, June 1928.

[62] D. A. Freitas and K. W. Current. CMOS Current Comparator Circuit. *Electronics Letters*, 19(17):695 – 697, August 1983.

[63] D. Frohman-Bentchkowsky. A Fully Decoded 2048-Bit Electrically Programmable FAMOS Read-Only Memory. *IEEE Journal of Solid-State Circuits*, 6(5):301 – 306, 1971.

[64] Robert C. Frye, Edward A. Rietman, and Chee C. Wong. Back-Propagation Learning and Nonidealities in Analog Neural Network Hardware. *IEEE Transactions on Neural Networks*, 2(1):110 – 117, January 1991.

[65] O. Fujita, Y. Amemiya, and A. Iwata. Characteristics of Floating Gate Device as Analogue Memory for Neural Networks. *Electronics Letters*, 27(11):924 – 926, May 1991.

[66] Osamu Fujita and Yoshihito Amemiya. A Floating-Gate Analog Memory Device for Neural Networks. *IEEE Transactions on Electron Devices*, 40(11):2029 – 2035, November 1993.

[67] Weinan Gao and W. Martin Snelgrove. The Floating Gate MOS Device as an Analogue Trimming Element. *Microelectronics Journal*, 25(5):353 – 361, 1994.

[68] Randall L. Geiger, Philip E. Allen, and Noel R. Strader. *VLSI Design Techniques for Analog and Digital Circuits*. McGraw-Hill Book Co., 1990.

[69] Hiroshi Gotou. New Operation Mode for Stacked-Gate Flash Memory Cell. *IEEE Electron Device Letters*, 16(3):121 – 123, March 1995.

[70] J. Haas, K. Au, L. C. Martin, T. L. Portlock, and T. Sakurai. High Voltage CMOS LCD Driver Using Low Voltage CMOS Process. In *Proceedings of CICC '89, IEEE Custom Integrated Circuits Conference*, San Diego (USA), May 1989.

[71] A. Hamilton, S. Churcher, A. F. Murray, G. B. Jackson, P. J. Edwards, and H. M. Reekie. Pulse Stream VLSI Circuits and Systems: The EPSILON Neural Network Chipset. *International Journal of Neural Systems*, 4(4):395 – 406, 1994.

[72] P. Hasler, C. Diorio, B. A. Minch, and C. Mead. Single Transistor Learning Synapse with Long-Term Storage. In *Proceedings of ISCAS'95, IEEE International Symposium on Circuits and Systems*, Seattle (USA), May 1995.

[73] Simon Haykin. *Neural Networks A Comprehensive Foundation*. Macmillan, Englewood Cliffs (USA), 1994.

[74] M. Herrman and A. Schenk. Field and High-Temperature Dependence of Long Term Charge Loss in Erasable Programmable Read Only Memories: Measurements and Modeling. *Journal of Applied Physics*, 77(9):4522 – 4540, May 1995.

[75] B. Hivert, M. Hoummady, J. M. Henrioud, and D. Hauden. Feasibility of Surface-Acoustic-Wave (SAW) Sensor Array-Processing with Formal Neural Networks. *Sensors and Actuators B-Chemical*, 19(1-3):645 – 648, 1994.

[76] B. Hochet. Multivalued MOS Memory for Variable-Synapse Neural Networks. *Electronics Letters*, 25(10):669 – 670, May 1989.

[77] Mark Holler, Simon Tam, Hernan Castro, and Ronald Benson. An Electrically Trainable Artificial Neural Network (ETANN) with 10240 "Floating Gate" Synapses. In *Proceedings of IJCNN '89, International Joint Conference on Neural Networks*, volume 2, pages 191 – 196, 1989.

[78] P. W. Hollis, J. S. Harper, and J. J. Paulos. The Effects of Precision Constraints in a Backpropagation Learning Network. *Neural Computation*, (2):363 – 373, 1990.

[79] P. W. Hollis and J. J. Paulos. Artificial Neural Networks Using MOS Analog Multipliers. *IEEE Journal of Solid State Circuits*, 25(3):849 – 855, June 1990.

[80] Andrew J. Holmes. *The Use of Non-Volatile a-Si:H Memory Devices for Synaptic Weight Storage in Artificial Neural Networks*. PhD thesis, The University of Edinburgh, 1995.

[81] Terence B. Hook and T.-P. Ma. Perimeter Related Current in High Field Tunneling into SiO_2. *Applied Physics Letters*, 47(4):417 – 419, August 1985.

[82] Y. Horio, M. Yamamoto, and S. Nakamura. Active Analog Memories for Neuro-Computing. In *Proceedings of ISCAS'90, International Symposium on Circuits and Systems*, volume 4, pages 2986 – 2989, 1990.

[83] Fu-Chieh Hsu, Richard S. Muller, Chenming Hu, and Ping-Keung Ko. A Simple Punchthrough Model for Short-Channel MOSFET's. *IEEE Transactions on Electron Devices*, ED-30(10):1354 – 1359, October 1983.

[84] Information Storage Devices Inc. WWW page: http://www.isd.com, November 1996. ISD, San Jose (USA).

[85] M. Jabri, S. Pichard, P. Leong, and Y. Xie. Algorithmic and Implementations Issues in Analog Low Power Learning Neural Network Chips. *Journal of VLSI Signal Processing*, 6(1):67 – 76, 1993.

[86] G. B. Jackson. *Hardware Neural Systems for Applications: a Pulsed Analog Approach.* PhD thesis, University of Edinburgh, February 1996.

[87] G. B. Jackson, A. Hamilton, and A. F. Murray. The EPSILON Processor Card: A Framework for Analog Neural Computation. In *Proceedings of Microneuro '94, International Conference on Microelectronics for Neural Networks*, 1994.

[88] William S. Johnson, George Perlegos, Alan Renniger, Greg Kuhn, and T. R. Ranaganath. A 16Kb Electrically Eraseable Non-Volatile Memory. *IEEE International Solid-State Circuits Conference Technical Papers*, 271:152–153, 1980.

[89] Tae-Sung Jung, Young-Joon Choi, Kang-Deog Suh, Byung-Hoon Suh, Jin-Ki Kim, Young-Ho Lim, Yong-Nam Koh, Jong-Wook Park, Ki-Jong Lee, Jung-Hoon Park, Kee-Tae Park, Jhang-Rae Kim, Jeong-Hyong Yi, and Hyung-Kyu Lim. A 117-mm^2 3.3-V Only 128-Mb Multilevel NAND Flash Memory for Mass Storage Applications. *IEEE Journal of Solid-State Circuits*, 31(11):1575 – 1583, November 1996.

[90] D. Kahng and S. Sze. A Floating Gate and its Application to Memory Devices. *Bell Systems Technical Journal*, 46:1288–1295, 1967.

[91] D. B. Kirk. *Accurate and Precise Computation using Analog VLSI, with Applications to Computer Graphics and Neural Networks.* PhD thesis, California Institute of Technology, 1993.

[92] Avinoam Kolodny, Sidney T. K. Nieh, Boaz Eitan, and Joseph Shappir. Analysis and Modeling of Floating-Gate EEPROM Cells. *IEEE Transactions on Electron Devices*, 33(6):835 – 844, June 1986.

[93] Hideo Kosaka, Tadashi Shibata, Hiroshi Ishii, and Tadahiro Ohmi. An Excellent Weight-Updating-Linearity EEPROM Synapse Memory Cell for Self-Learning Neuron-MOS Neural Networks. *IEEE Transactions on Electron Devices*, 42(1):135 – 143, January 1995.

[94] C. Kuo, J. R. Yeargain, W. J. Downey, K. A. Ilgenstein, J. R. Jorvig, S. L. Smith, and A. R. Bormann. An 80ns 32K EEPROM using the FETMOS Cell. *IEEE Journal of Solid-State Circuits*, 17(5):821 – 827, October 1982.

[95] S. K. Lai, V. K. Dham, and D. Guterman. Comparison and Trends in Today's Dominant EE Technologies. *IEDM Technical Digest*, pages 580 – 583, 1986.

[96] Tor Sverre Lande, Hassan Ranjbar, Mohammed Ismail, and Yngvar Berg. An Analog Floating-Gate Memory in a Standard Digital Technology. In *Proceedings of Microneuro '96, Fifth International Conference on Microelectronics for Neural Networks and Fuzzy Systems*, Lausanne (Switzerland), February 1996. IEEE Computer Society Press.

[97] M. Lanzoni, L. Briozzo, and B. Riccò. A Novel Approach to Controlled Programming of Tunnel-Based Floating-Gate MOSFET's. *IEEE Journal of Solid-State Circuits*, 29(2):147 – 150, February 1994.

[98] John Lazzaro, John Wawrzynek, and Alan Kramer. Systems technologies for silicon auditory models. *IEEE Micro*, 14(3), March 1994.

[99] Bang W. Lee, Bing J. Sheu, and Han Yang. Analog Floating-Gate Synapses for General-Purpose VLSI Neural Computation. *IEEE Transactions on Circuits and Systems*, 38(6):654 – 658, 1991.

[100] Jack C. Lee, Ih-Chin Chen, and Chenming Hu. Modeling and Characterization of Gate Oxide Reliability. *IEEE Transactions on Electron Devices*, 35(12):2268 – 2278, December 1988.

[101] Torsten Lehmann. Implementation Issues of Self-learning Pulsed Integrated Neural Systems. In *Proceedings of the 13th NORCHIP Conference*, pages 145 – 152, Copenhagen (Denmark), 1995.

[102] Torsten Lehmann. Teaching Pulsed Integrated Neural Systems: a Psychobiological Approach. In *Proceedings of Microneuro '96, Fifth International Conference on Microelectronics for Neural Networks and Fuzzy Systems*, Lausanne (Switzerland), February 1996. IEEE Computer Society Press.

[103] J. A. Leonard, M. A. Kramer, and L. H. Ungar. A Neural Network Architecture that Computes its Own Reliability. *Computers and Chemical Engineering*, 16(9):819 – 835, 1992.

[104] Bernabé Linares-Barranco, Edgar Sánchez-Sinencio, Angel Rodriguez-Vázquez, and Josè L. Huertas. A CMOS Analog Adaptive BAM with On-Chip Learning and Weight Refreshing. *IEEE Transactions on Neural Networks*, 4(3), May 1993.

[105] W. Maly. *Atlas of IC Technologies*. The Bejamin/Cummings Publishing Company, Inc., Menlo Park (USA), 1987.

[106] J. R. Mann. Floating Gate Circuits in MOSIS. Technical Report 824, Massachusetts Institute of Technology Lincoln Laboratory, November 1990.

[107] G. Marshall and S. Collins. An Analog Radial Basis Function Circuit Incorporating Floating-Gate Devices. *Analog Integrated Circuits and Signal Processing*, 11(1):21–34, 1996.

[108] G. F. Marshall and S. Collins. Fuzzy Logic Architecture using Subthreshold Analogue Floating-Gate Devices. *IEEE Transactions on Fuzzy Systems*, 5(1):32 – 43, 1997.

[109] T. Matsushita, T. Mihara, H. Ikeda, M. Hirota, and Y. Hirota. A Surge-Free Intelligent Power Device Specific to Automotive High Side Switches. *IEEE Transactions on Electron Devices*, 38(7):1576 – 1579, July 1991.

[110] D. J. Mayes. *Implementing Radial Basis Function Neural Networks in Pulsed Analogue VLSI*. PhD thesis, University of Edinburgh, January 1997.

[111] D. J. Mayes, A. Hamilton, and J. E. Louvet. A VSLI Current-Mode Synapse Chip. In *From Natural to Artificial Neural Computation, Proceedings of the IWANN '95, International Workshop on Artificial Neural Networks (Lecture Notes in Computer Science No. 930)*, pages 815 – 821, Malaga-Torremolinos (Spain), June 1995.

[112] Neal Mielke, Albert Fazio, and Ho-Chun Liou. Reliability Comparison of FLOTOX and Textured-Polysilicon E^2PROMs. In *Proceedings of IRPS '87, 25th IEEE International Reliability Physics Symposium*, pages 85 – 92, San Diego (USA), April 1987.

[113] Antonio J. Montalvo, Ronald S. Gyurcsik, and John J. Paulos. Toward a General-Purpose Analog VLSI Neural Network with On-Chip Learning. *IEEE Transactions on Neural Networks*, 8(2):413 – 423, March 1997.

[114] Antonio J. Montalvo and John J. Paulos. Improved Floating-Gate Devices Using Standard CMOS Technology. *IEEE Electron Device Letters*, 14(8), 1993.

[115] D. Montanari, J. Van Houdt, G. Groeseneken, and H. E. Maes. Novel Level-Identifying Circuit for Flash Multi-Level Memories. In *Proceedings of ESSCIRC '97, 23rd European Solid-State Circuits Conference*, pages 184 – 187, September 1997.

[116] John Moody and Christian Darken. Learning with Localized Receptive Fields. In *Proceedings of the Connectionist Models Summer School*, San Mateo (USA), 1988. Morgan Kaufman.

[117] Takashi Morie, Osamu Fujita, and Kuniharu Uchimura. Self-Learning Analog Neural Network LSI with High-Resolution Non-Volatile Analog Memory and a Partially-Serial Weight-Update Architecture. *IEICE Transactions on Electronics*, E80-C(7):990 – 995, July 1997.

[118] A. F. Murray and A. V. W. Smith. Asynchronous VLSI neural networks using pulse-stream arithmetic. *IEEE Journal of Solid-State Circuits*, 23(3):688–697, June 1988.

[119] Alan F. Murray and Bill Buchan. A User's Guide to Non-Volatile, On-Chip Analogue Memory. Paper submitted to IEE Electronics and Communications Engineering Journal.

[120] Alan F. Murray, Stephen Churcher, Alister Hamilton, Andrew J. Holmes, Geoff B. Jackson, H. Martin Reekie, and Robin J. Woodburn. Pulse stream VLSI neural networks. *IEEE Micro*, 14(3):29 – 39, June 1994.

[121] Hirotaka Muto, Hiroyoshi Kitabayashi, Koichiro Nakanishi, Setsuo Wake, and Moriyoshi Nakajima. Numerical Analysis of Tunneling Current due to Electric Field Concentration at Gate Edge of Polysilicon/SiO_2/Silicon Structures. *Japanese Journal of Applied Physics Part I*, 33(1B):623 – 629, 1994.

[122] Koji Nakajima, Shigeo Sato, Tomoyasu Kitaura, Junichi Murota, and Yasuji Sawada. Hardware Implementation of New Analog Memory for Neural Networks. *IEICE Transactions on Electronics*, E78-C(1), January 1995.

[123] E. M. S. Narayanan, G. A. J. Amaratunga, W. I. Milne, J. I. Humphrey, and Q. Huang. Analysis of CMOS-Compatible Lateral Insulated Base Transistors. *IEEE Transactions on Electron Devices*, 38(7):1624 – 1632, July 1991.

[124] Ron Neale. Will Multi-Level Flash Memory be the Way to the Future. *Electronic Engineering*, 68(835):33 – 36, 1996.

[125] A. Nedungadi and T. R. Viswanathan. Design of Linear CMOS Transconductance Elements. *IEEE Transactions on Circuits and Systems*, 31(10):891 – 894, 1984.

[126] T. H. Ning, C. M. Osburn, and H. N. Yu. Emission Probability of Hot Electrons from Silicon into Silicon Dioxide. *Journal of Applied Physics*, 48:286 – 293, 1977.

[127] Hiroshi Nozawa and Susumu Kohyama. A Thermionic Electron Emission Model for Charge Retention in SAMOS Structures. *Japanese Journal of Applied Physics*, 21(2):L111 – L112, February 1982.

[128] Hiroshi Nozawa and Keikichi Tamaru. Low Activation Energy by Polarization in a Floating Gate Structure. *Japanese Journal of Applied Physics*, 32(Part 2, No. 10B):L1506 – L1508, October 1993.

[129] Masayoshi Ohkawa, Hiroshi Sugawara, Naoaki Sudo, Masaru Tsukiji, Ken ichiro Nakagawa, Masato Kawata, Ken ichi Oyama, Toshio Takeshima, and Shuichi Ohya. A 98mm^2 Die Size 3.3-V 64-Mb Flash Memory with FN-NOR Type Four-Level Cell. *IEEE Journal of Solid-State Circuits*, 31(11):1584 – 1589, November 1996.

[130] T.-C. Ong, P. K. Ko, and C. Hu. The EEPROM as an Analog Memory Device. *IEEE Transactions on Electron Devices*, 36(9):1840 – 1841, September 1989.

[131] Giuseppe Palmisano and Gaetano Palumbo. High Performance CMOS Current Comparator Design. *IEEE Transactions on Circuits and Systems - II: Analog and Digital Signal Processing*, 43(12):785 – 790, December 1996.

[132] Constantin Papadas, George Pananakakis, Gérard Ghibaudo, Carlo Riva, Federico Pio, and Paolo Ghezzi. Modeling of the Intrinsic Retention Characteristics of FLOTOX EEPROM Cells Under Elevated Temperature Conditions. *IEEE Transactions on Electron Devices*, 42(4), April 1995.

[133] J. Park and I. W. Sandberg. Universal Approximation Using Radial Basis Function Networks. *Neural Computation*, 3:246 – 257, 1991.

[134] Zahir Parpia. Intel Announced 64Mbit Multilevel Flash Chip. *Zahir's Electronic Newsletter*, 3(15), September 1997.

[135] Zahir Parpia, C. Andre T. Salama, and Robert A. Hadaway. Modeling and Characterization of CMOS-Compatible High-Voltage Device Structures. *IEEE Transactions on Electron Devices*, 34(11):2335 – 2343, November 1987.

[136] Richard D. Pashley. Flash Memories: The Best of Two Worlds. *IEEE Spectrum*, 26(12):30 – 33, December 1989.

[137] Corey Petersen and Allen R. Barlow. High Voltage Circuits in Standard CMOS Processes. In *Proceedings of CICC '82, IEEE Custom Integrated Circuits Conference*, pages 287 – 291. IEEE, May 1982.

[138] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipies in C.* Cambridge University Press, Cambridge (UK), 2nd edition, 1992.

[139] Riko Radojcic. Some Aspects of Hot-Electron Aging in MOSFET's. *IEEE Transactions on Electron Devices*, ED-31(10):1381 – 1385, 1984.

[140] J. Ramírez-Angulo, S. C. Choi, and G. González-Altamirano. Low-Voltage Circuits Building Blocks Using Multiple-Input Floating-Gate Transistors. *IEEE Transactions on Circuits and Systems - I: Fundamental Theory and Applications*, 42(11):971 – 974, November 1995.

[141] D. Rinaldi, S. Santini, and M. Vanzi. Electron Tunnelling from Rough Surfaces: An Application to TPFG EEPROM Cells. *Semiconductor Science and Technology*, (9):1414 – 1425, 1994.

[142] Peter E. Cottrell Ronald R. Troutman, Thomas V. Harroun and Satya N. Chakravarti. Design Considerations for RAM Chips. *IEEE Transactions on Electron Devices*, ED-27(8):1629 – 1639, 1980.

[143] Anirban Roy, Frank R. Libsch, and Marvin H. White. Electron Tunneling from Polysilicon Asperities into Poly-Oxides. *Solid-State Electronics*, 32(8):655 – 659, 1989.

[144] RS. *Catalogue*, volume 3, Electronic Components and Power Conversion. July 1997.

[145] Eduard Säckinger and Walter Guggenbühl. An Analog Trimming Circuit Based on a Floating-Gate Device. *IEEE Journal of Solid-State Circuits*, 23(6):1437 – 1440, December 1988.

[146] C. T. Sah. Models and Experiments on Degradation of Oxidized Silicon. *Solid-State Electronics*, 33(2):147 – 167, 1990.

[147] T. Sawaji, T. Sakai, H. Nagai, T. Kunishima, and T. Matsumoto. Implementing Resistive Fuse with Floating Gate MOS Transistors. In *Proceedings of ICNN '97, International Conference on Neural Networks*, volume 2, pages 894 – 896, Houston (USA), June 1997.

[148] Christian Schneider and Howard Card. Analog CMOS Synaptic Learning Circuits Adapted from Invertebrate Biology. *IEEE Transactions on Circuits and Systems*, 38(12):1430 – 1438, 1991.

[149] M. Schubert. 70V-to-5V Differential CMOS Input Interface. *Electronics Letters*, 30(4):296 – 297, February 1994.

[150] Tadashi Shibata, Hideo Kosaka, Hiroshi Ishii, and Tadahiro Ohmi. A Neuron-MOS Neural Network Using Self-Learning-Compatible Synapse Circuits. *IEEE Journal of Solid-State Circuits*, 30(8):913 – 922, August 1995.

[151] Takeshi Shima and Stephanie Rinnert. Multiple-Valued Memory Using Floating Gate Devices. *IEICE Transactions on Electronics*, E76-C(3), March 1993.

[152] Chi-Kai Sin, Alan Kramer, V. Hu, Robert R. Chu, and Ping K. Ko. EEPROM as an Analog Storage Device, with Particular Applications in Neural Networks. *IEEE Transactions on Electron Devices*, 39(6):1410 – 1419, June 1992.

[153] P. I. Suciu, B. P. Cox, D. D. Rinerson, and S. F. Cagnina. Cell Model for EEPROM Floating Gate Memories. *IEEE IEDM*, pages 737 – 740, 1982.

[154] J. Suñé, M. Nafría, and X. Aymerich. Reversible Dielectric-Breakdown of Thin Gate Oxides in MOS Devices. *Microelectronics and Reliability*, 33(7):1031 – 1039, 1993.

[155] S. M. Sze. *Semiconductor Devices: Physics and Technology*. Wiley, Chichester, New York (USA), 1985.

[156] S. M. Sze. *VLSI Technology*. McGraw-Hill Book Company, 2nd edition, 1988.

[157] S. M. Sze and G. Gibbons. Effect of Junction Curvature on Breakdown Voltage in Semiconductors. *Solid State Electronics*, (9):831 – 845, September 1966.

[158] R. Tawel. Learning in Analog Neural Network Hardware. *Computers and Electrical Engineering*, 19(6):453 – 467, 1993.

[159] Theodore L. Tewksbury and Hae-Seung Lee. Characterization, Modeling, and Minimization of Transient Threshold Voltage Shifts in MOSFET's. *IEEE Journal of Solid-State Circuits*, 29(3), March 1994.

[160] A. Thomsen and M. A. Brooke. Low Control Voltage Programming of Floating Gate MOSFETs and Applications. *IEEE Transactions on Circuits and Systems - I: Fundamental Theory and Applications*, 41(6):443 – 452, June 1994.

[161] Axel Thomsen and Martin A. Brooke. A Floating-Gate MOSFET with Tunneling Injector Fabricated Using a Standard Double-Polysilicon CMOS Process. *IEEE Electron Device Letters*, 12(3):111 – 113, 1991.

[162] Toru Toyabe, Ken Yamaguchi, Shojiro Asai, and Michael S. Mock. A Numerical Model of Avalanche Breakdown in MOSFET's. *IEEE Transactions on Electron Devices*, ED-25(7), July 1978.

[163] Hieu Van Tran, Trevor Blyth, David Sowards, Larry Engh, B. S. Nataraj, Tony Dunne, Hai Wang, Vishal Sarin, Tin Lam, Hagop Nazarian, and Genda Hu. A 2.5V 256-Level Non-Volatile Analog Storage Device Using EEPROM Technology. In *Proceedings of ISSCC '96, IEEE International Solid-State Circuits Conference (Digest of Technical Papers)*, number FP 16.6, pages 270 – 271, 1996.

[164] Gautam Verma and Neal Mielke. Reliability Performance of ETOX Based Flash Memories. In *Proceedings of IRPS '88, IEEE International Reliability Physics Symposium*, pages 158 – 166, 1988.

[165] J. F. Verweij and J. H. Klootwijk. Dielectric Breakdown I: A Review of Oxide Breakdown. *Microelectronics Journal*, 27(7):611 – 622, 1996.

[166] E. Vittoz, H. Oguey, M. A. Maher, O. Nys, E. Dijkstra, and M. Chevroulet. Analog Storage of Adjustable Synaptic Weights. In U. Ramacher and U. Rückert, editors, *VLSI Design of Neural Networks*, pages 47 – 63. Kluwer, 1991.

[167] Eric A. Vittoz. MOS Transistors Operated in the Lateral Bipolar Mode and Their Application in CMOS Technology. *IEEE Journal of Solid-State Circuits*, 18(3):273 – 279, June 1983.

[168] Steven S. Watkins, Paul M. Chau, Raoul Tawel, and Bjorn Lambrigsten. Execution of a Remote Sensing Application on a Custom Neurocomputer. *IEEE Transactions on Neural Networks*, 6(6), November 1995.

[169] H. A. Richard Wegener. Endurance Model for Textured-Poly Floating Gate Memories. *IEEE-IEDM Digest*, pages 480 – 483, 1984.

[170] Neil H. E. Weste and Kamran Eshraghian. *Principles of CMOS VLSI Design.* Addison-Wesley Publishing Company, 2nd edition, 1993.

[171] Marvin H. White, Yang (Larry) Yang, Ansha Purwar, and Margaret L. French. A Low Voltage SONOS Nonvolatile Semiconductor Memory Technology. *IEEE Transactions on Components, Packaging and Manufacturing Technology - Part A*, 20(2):190 – 195, June 1997.

[172] B. Widrow and M. A. Lehr. 30 Years of Adaptive Neural Networks: Perceptron, Madaline and Backpropagation. *Proceedings of the IEEE*, 78:1415 – 1442, 1990.

[173] Kurt Wolf. The Differences Between NAND and NOR Flash Memories. *Electronic Design*, 42(10):80, May 1994.

[174] R. F. Wolffenbuttel, editor. *Silicon Sensors and Circuits: On-Chip Compatability.* Chapman & Hall, London (UK), 1996.

[175] D. R. Wolters and J. J. van der Schoot. Dielectric-Breakdown in MOS Devices 1. Defect-Related and Intrinsic Breakdown. *Philips Journal of Research*, 40(3):115 – 136, 1985.

[176] D. R. Wolters and J. J. van der Schoot. Dielectric-Breakdown in MOS Devices 2. Conditions for the Instrinic Breakdown. *Philips Journal of Research*, 40(3):137 – 163, 1985.

[177] Hei Wong. A Physically-Based MOS Transistor Avalanche Breakdown Model. *IEEE Transactions on Electron Devices*, 42(12), December 1995.

[178] Robin Woodburn. *An Investigation of Practical Issues in Translating Algorithms Based on Back-Propagation into Analogue, VLSI Circuits.* PhD thesis, University of Edinburgh, March 1996.

[179] A. Wright. Analog Data Storage - Speaking of the Future. *Electronics World and Wireless World*, 98(1671):110 – 113, 1992.

[180] C. M. Wu and K. W. Yeh. Safety Drain Voltage Avoiding Avalanche Breakdown in MOSFET. *IEEE Electron Device Letters*, EDL-3(9), September 1982.

[181] Ching-Yuan Wu, Wei-Zang Hsiao, and Hsing-Hai Chen. A Simple Punch-through Voltage Model for Short-Channel MOSFET's with Single Channel Implantation in VLSI. *IEEE Transactions on Electron Devices*, ED-32(9):1704 – 1707, September 1985.

[182] Xicor. *Nonvolatile Solutions Data Book.* XICOR, Inc., Milpitas (USA), 1992.

[183] Kikuo Yamabe and Kenji Taniguchi. Time-Dependent Dielectric-Breakdown of Thin Thermally Grown SiO_2 Films. *IEEE Journal of Solid-State Circuits*, 20(1):343 – 348, February 1985.

[184] Edward S. Yang. *Microelectronic Devices.* McGraw-Hill Book Co., Singapore, 1988.

[185] Giora Yaron, S. Jayashima Prasad, Mark S. Ebel, and Bruce M. K. Leong. A 16K E^2PROM Employing New Array Architecture and Designed-In Reliability Features. *IEEE Journal of Solid-State Circuits*, SC-17(5), 1982.

[186] Carol Young. *The New Penguin Dictionary of Electronics.* Penguin Books, London (UK), 1979.