Applications of Nanoelectronic Technology to Image Processors, Velocity-Tuned Filters and Crossbar Memories

by

Woo Hyung Lee

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical Engineering)
in The University of Michigan
2009

Doctoral Committee:

Professor Pinaki Mazumder, Co-Chair
Assistant Professor Wei Lu, Co-Chair
Professor Marios C. Papaefthymiou
Associate Professor Anna C. Gilbert
Assistant Professor Michel M. Maharbiz

© Woo Hyung Lee
2009

To my parents

Acknowledgements

Warmest thanks to my advisors, Professor Pinaki Mazumder and Professor Wei Lu, for giving me the opportunity to pursue the Ph. D program at the University of Michigan. While working for them, I have experienced a big picture of my research area and appreciated their invaluable advices and comments. I also would like to express my appreciation to Professor Marios Papaefthymiou for his kindness to my questions about anything related with my research area. I am also appreciating my other committee members for their accepting my proposal. I also would like to thank Manoj Rajagopalan, Sing-Rong Li, Sanjay Pant, Joohee Kim, Yoonna Oh and Eunjung Cho.

I cannot describe my appreciation on my parents, Yoonkeun Lee and Youngsoon Lee, who are in heaven now. I really love and miss them very much. Whenever I am in trouble with the world, I believe they give me hope for the future. I also love and appreciate my wife, Youri Bae from my heart. Without her, I cannot imagine how I live. I really love my new born bary, Sophia Lee, very much and could be refreshed by her smile. I also appreciate my parents in law, Joongnam Bae and Seolhee Kim from my heart.

TABLE OF CONTENTS

DEDICATION ii

ACKNOWLEDGEMENTS iii

LIST OF FIGURES viii

LIST OF TABLES xii

ABSTRACT xiv

CHAPTERS

I. Introduction ... 1
1.1. Motivation ... 1
1.2. Backgrounds ... 6
  1.2.1. RTD fabrication ... 6
  1.2.2. Cellular neural network ... 6
  1.2.3. Image processing using an array of RTD nanostructure ... 9
  1.2.4. Velocity-tuned filter ... 10
  1.2.5. Crossbar memory ... 11
1.3. Thesis Overview ... 19
  1.3.1. Image Processing using an array of RTD nanostructure ... 19
  1.3.2. Velocity-tuned filter ... 19
  1.3.3. Crossbar memory architecture ... 20
II. 2-D Array of Multi-Peak Resonant Tunneling Diodes Based Color Image Processing ... 21
2.1. Introduction ... 21
2.2. Multi-Peak RTDs Based Color Image Processor ... 23
2.3. Color Representation Method ... 29
2.4. Color Quantization ... 30
  2.4.1. Implementation and results ... 30
  2.4.2. Settling time analysis ... 35
  2.4.3. Power consumption analysis ... 37
2.5. Smoothing Function ... 39
  2.5.1. Implementation and results ... 39

2.5.2. Settling time 43
2.5.3. Power consumption analysis 44
2.6. Color Extraction 47
2.7. Comparison with DSP Chips 55
2.8. Stability 58
2.9. Conclusion 60
III. Design of a Velocity-Tuned Filter Using a Matrix of Resonant Tunneling Diodes .61
3.1. Introduction 61
3.2. Array of RTDs Based Velocity-Tuned Filters 62
3.2.1. Conventional velocity-tuned filter 62
3.2.2. Resonant tunneling diode 66
3.2.3. Proposed velocity-tuned filter 69
3.3. System Analysis 75
3.3.1. Delay analysis of velocity-tuned filter 75
3.3.2. Power consumption analysis of velocity-tuned filter 79
3.3.3. Stability of velocity-tuned filter 83
3.4. Conclusion 87
IV. Crossbar Memory Simulation And Performance Evaluation 89
4.1. Introduction 89
4.1.1. Motivation 89
4.1.2. Contrast with competing technologies 91
4.1.3. Amorphous Si crossbar memory cell 92
4.2. Structure 96
4.2.1. Crossbar modeling 100
4.3. Writing Strategy and Circuit Implementation 106
4.4. Reading Strategy and Circuit Implementation 109
4.5. Memory Architecture 116
4.6. Power Dissipation 125
4.6.1. Power estimation 125
4.6.2. Analytical modeling on static power 131
4.7. Noise Analysis 137
4.8. Area Overhead 140
4.9. Bank-Based System Design 153
4.10. Technology Comparison 157
4.11. Conclusion 158
V. Summary and Future Work 160

5.1. Summary 160
5.2. Future Work 162

BIBLIOGRAPHY 163

List of Figures

Figure

1.1: *I-V* curve of a resonant tunneling diode………………………………………… 2
1.2: CNN architecture consisting of M by N array of cells…………………………. 7
1.3: Cell realization of a standard CNN cell C(i,j)…………………………………... 8
1.4: Conceptual diagram and circuit model of an array of RTD image processor…... 9
1.5: Memory cells of DRAM (a), SRAM (b) and MRAM (c)………………………. 13
1.6: A simple crossbar memory architecture for nanoscale molecular-switch crossbar circuits ……………………………………………………………….. 16
1.7: A crossbar memory architecture based on nanowire transistor………………... 18
2.1: A multi-peak RTD based image processor…………………………………….. 24
2.2: Simplified piecewise-linear model of *I-V* characteristics for an eight-peak RTDs………………………………………………………………………… 25
2.3: A color representation method with metallic islands on an eight-peak RTD….. 30
2.4: Transient representation of 4 x 4 color image quantization from 0 ps to 200 ps 31
2.5: HSPICE simulation results divided into rows with time from 0ns to 1ns……... 32
2.6: 50X50 pixel color image quantization from 0ns to 1ns……………………….. 35
2.7: Equivalent circuit model of the multi-peak RTDs for color quantization……... 37
2.8: Transient representation of a 4 x 4 color image smoothing function………….. 40
2.9: HSPICE results divided into rows with time from 0 to 2ns…………………… 41
2.10: 50 x 50 color image smoothing from 0ns to 3ns……………………………... 42
2.11: Equivalent circuit model of an array of multi-peak RTD structure for smoothing………………………………………………………………………… 44
2.12: Circuit configurations for color extraction……………………………………. 47
2.13: White color extraction………………………………………………………… 48

2.14: A circuit configuration for any color extraction………………………………. 51
2.15: Red color extraction of 4x4 pixel image………………………………………. 52
2.16: HSPICE simulation results divided into rows………………………………… 52
2.17: The color extractions of various colors with time from 0ns to 1.5ns…………. 54
2.18: DSP block diagram of TI TMSC55XX……………………………………….. 56
3.1: Conventional velocity-tuned filter……………………………………………... 64
3.2: Modeling of resonant tunneling diode based on experimental result…………. 66
3.3: Proposed velocity-tuned filter with metallic islands on the RTD substrate…… 69
3.4: Circuit model of a basic cell of the proposed velocity-tuned filter…………….. 71
3.5: HSPICE simulation result for velocity-tuned filter with velocity 0…………… 74
3.6: HSPICE simulation result for velocity-tuned filter with velocity 1……………. 74
3.7: Experimental result of velocity-tuned filter for a velocity of 0 pixel/second….. 77
3.8: Experimental result of velocity-tuned filter for a velocity of 1 pixel/second….. 78
3.9: RC tree representation of the proposed velocity-tuned filter…………………… 82
3.10: Equivalent circuit model of an unit cell of velocity-tuned filters…………….. 83
4.1: A crosscut view of an amorphous Silicon crossbar memory cell……………… 94
4.2: Floorplan of the amorphous crossbar memory architecture…………………… 96
4.3: A schematic of the row decoder of a 4x4 array with highlighted write “1” operation………………………………………………………………………… 99
4.4: A schematic of the row decoder of a 4x4 array with highlighted read operation 100
4.5: A cross cut of the crossbar resistive memory device…………………………... 101
4.6: A simple RC model of the crossbar resistive memory device…………………. 102
4.7: A diagram of a cell of the crossbar resistive memory device………………….. 104
4.8: A simple RC model of the crossbar resistive memory device…………………. 104
4.9: A simple 2-D RC model of the crossbar resistive memory device…………….. 105
4.10: Row decoder and row power assignment for writing “1”…………………….. 107
4.11: Column decoder and column power assignment for writing “1”……………... 107

4.12: Row decoder and row power assignment for writing “0”…………………….. 108
4.13: Column decoder and column power assignment for writing “0”……………… 108
4.14: Sense amplifier using the single-ended current-mirror amplifier…………….. 110
4.15: Sense amplifier using cross-coupled amplifier…………………………….. 111
4.16: Sense amplifier using a cross-coupled CMOS inverter latch…………………. 112
4.17: Suggested sense amplifier for the amorphous crossbar memory……………… 113
4.18: HSPICE simulation results of the read circuit………………………………… 114
4.19: Schematic of a 1 Kb memory design with a single sense amplifier………….. 117
4.20: Schematic of a 1 Kb memory design with sense amplifiers before the column decoder……………………………………………………………………… 118
4.21: HSPICE simulation results of the memory with sense amplifiers before the column decoder for reading “high”………………………………………………. 120
4.22: HSPICE simulation results of the memory with sense amplifiers before the column decoder for reading “low”……………………………………………….. 121
4.23: HSPICE simulation results of the 1Kb crossbar memory design with sense amplifiers before the column decoder for reading……………………………….. 123
4.24: Static power dissipation of the SA part in the 1Kb crossbar memory design with a SA after decoders from 0 ns to 7 ns……………………………………….. 127
4.25: Power dissipation of the SA part in 1Kb crossbar memories with a SA after decoders from 0 ns to 7 ns……………………………………………………... 129
4.26: A 2x1 MUX for power selection which is connected to row decoders……….. 133
4.27: A simplified 2-D RC model of crossbar memory cells……………………….. 138
4.28: Interconnect coupling model………………………………………………….. 139
4.29: A simplified sense amplifier part to investigate the feasibility of usage of MUX logics………………………………………………………………………….. 141
4.30: Simulation results of the simplified sense amplifier with MUX logics……….. 142
4.31: Schematic of the 1 Kb crossbar memory design with using 2 by 1 MUX

logics to reduce the numbers of sense amplifiers by one half………………………. 143
4.32: Simulation results on the 1 Kb crossbar memory design with 2 by 1 MUX logics to reduce the numbers of sense amplifiers…………………………………… 144
4.33: Schematic of the 1 Kb crossbar memory design with using 4 by 1 MUX logics to reduce the numbers of sense amplifiers by one quarter…………………… 145
4.34: Simulation results on the 1 Kb crossbar memory design with 4 by 1 MUX logics to reduce the numbers of sense amplifiers…………………………………… 146
4.35: Schematic of the 1Kb crossbar memory design with using 8 by 1 MUX logics to reduce the numbers of sense amplifiers by one quarter………………………….. 147
4.36: Simulation results on the 1Kb crossbar memory design with 8 by 1 MUX logics to reduce the numbers of sense amplifiers…………………………………… 149
4.37: Simulation results with the changes of the MUX selection bits and addresses on the 1Kb crossbar memory design with 8 by 1 MUX logics……………………… 151
4.38: Schematic view of 4 Kb crossbar memories with 4 banks……………………. 153

List of Tables

Table

2.1: Comparison of stabilization time among image processing functions…………. 36

2.2: Comparison of Properties of Image Functions…………………………………. 55

2.3: Operational Specifications for different processors……………………………. 57

2.4: Performance comparison among different processors for smoothing function… 58

3.1: Performance comparison of VTFs with 20 x 20 pixel images…………………. 82

4.1: Area composition of the 1 Kb crossbar memory design with a SA attached to all the bit lines………………………………………………………………………. 122

4.2: Area composition of the 1 Kb crossbar memory design with a SA at the final stage……………………………………………………………………………. 122

4.3: Comparison of date read time between the 1Kb memories ……………………. 124

4.4: Comparison of power dissipation between the two types of 1 Kb crossbar memory cell designs……………………………………………………………… 131

4.5: Area comparison among the 1Kb crossbar memory designs…………………… 151

4.6: Area comparison among the 1Mb crossbar memory designs…………………... 152

4.7: Read time comparison among the 1Mb crossbar memory designs with various sizes of MUX…………………………………………………………………... 152

4.8: Comparison of total area of 1 Mb crossbar memory designs…………………... 153

4.9: Read time comparison between 1 Kb and 4 Kb with 4 banks………………….. 154

4.10: Read time and area overhead comparison for the optimal size of a unit array to make 1 Gb crossbar memories with a cell size of 200 nm x 200 nm……………………………………………………………………………… 156

4.11: Estimated read time for 4 Gb and 32 Gb crossbar memory designs…………………………………………………………………………….. 156

4.12: Estimation of memory capacity to be realized in a 1 cm x 1 cm die………….. 156
4.13: Technology comparison between the amorphous crossbar memories………… 157

ABSTRACT

It is widely expected new devices and architectures will have to be developed to sustain the relentless performance scaling trend the CMOS industry has enjoyed in the last 40 years. In this thesis, we demonstrate several nanoelectronics implementations that offer much higher information throughput than their CMOS counterparts. Chapter 1 provides the motivation and background information for the implementations including image processors, velocity-tuned filters and crossbar memories. In Chapter 2, color image processors based on multi-peak resonant tunneling diodes are discussed. The multi-peak resonant tunneling diodes can be configured into a two-dimensional array of regular computing elements, locally connected by programmable passive and active elements with a view to realize a wide gamut of color image processing functions such as quantization, color extraction, image smoothing, edge detection and line detection. In order to process color information of the input images, two different methods for color representation schemes were discussed: one using color mapping, and the other using direct RGB representation. Finally, we demonstrate image functions through HSPICE simulation using these methods on the proposed nano-architecture.

In chapter 3, we propose a nanoscale velocity-tuned filter that employs resonant tunneling diodes to perform temporal filtering to track moving and stationary objects. The new velocity-tuned filter is not only amenable for nanocomputing, but also superior to other approaches in terms of area, power and speed. We show that the proposed nanoarchitecture for velocity-tuned filter is asymptotically stable in the specific region.

In chapter 4, we electrically modeled a crossbar memory cell and designed specific peripheral circuitry, including the column and row decoders, a sensing circuitry for detecting the difference in the resistance of the cell, and the control circuit for reading and writing. An analytical model on static power dissipation was conducted to suggest

optimal design for power dissipation. To improve the area overhead, we introduced MUX logics to reduce the numbers of sense amplifiers attached to all the bit lines. In addition, scaling issues of the crossbar memory design due to the size mismatch between crossbar memory arrays and CMOS peripheral circuitry is discussed.

CHAPTER I

Introduction

1.1. Motivation

Even though CMOS technology has been playing a prevailing role in commercial electronic products during the past three decades, it is fraught with myriad problems due to continual aggressive scaling of device structures. The scaling problem has resulted in short channel effects, and, in the near future, the scaling may hit a red brick wall because of physical limitations of CMOS device structures. Further, it also exacerbates the thermal management of heat dissipated inside a chip and also the noise problem becomes quite severe as the feature size of CMOS devices aggressively decreases towards 25 nano meter [1.28][1.29].

To overcome these problems with a conventional CMOS technology, two kinds of approaches have been investigated. One way is to replace conventional devices with new types of devices or modified CMOS devices [1.11][1.12][1.13][1.14]. The other approach is to find other architectures, which can enhance the performance significantly [1.15][1.16].

Resonant Tunneling Diodes (RTDs) have been a strong candidate for replacing the conventional CMOS in specific applications since it retains fast switching speed. In the device characteristics, the RTDs have three regions in the *I-V* characteristic curve. It has two positive differential resistance regions and one negative differential resistance region between them as shown in Figure 1.1. This negative differential resistance characteristic can be used in carefully designed circuits to improve the device performance and consume less power [1.10].

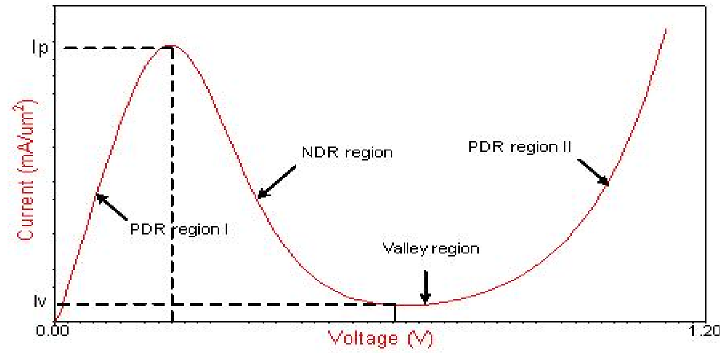


Figure 1.1: *I-V* characteristic of a resonant tunneling diode.

To reduce the area of RTDs based chips, an RTD array was constructed consisting of a 2D array of metallic islands on top of an RTD substrate. This device structure demonstrates better area usage as well as better performance than conventional RTD based chips in power and speed. The better area usage comes from the size of the small cell and the power is also saved through the small area of current path even though the cell is a two terminal device that typically leads to high power dissipation. Speed is enhanced by using the fast switching due to the negative differential resistance characteristic [1.1]. However, this approach reveals that the functionality is vulnerable from the size variation during fabrication process. Also, the interconnection between cells becomes complicated when many functions are implemented. These problems will be reduced as the process technology is advanced. The current lithography technology can define 45 nm with less than 5 % errors [1.52], and the defined length will be reduced in the future. The interconnection problems can also be reduced further in the future.

The second approach is to find other types of architectures than conventional CMOS based processor architecture, which fit to the specific functionalities offered by the nanoscale devices so the performance can be enhanced significantly [1.16][1.17].

Cellular Neural Network (CNN) is an analog dynamic processor array architecture which reflects the processing elements that interact directly within a finite local neighborhood. The CNN is a large size nonlinear circuit consisting of repeatedly spaced circuit clones. The connectivity between clones within neighboring cells forms a CNN template. By changing this template, various functions can be implemented. In CNNs, each cell pertaining to its own function and the interaction with neighboring cells decide the overall function of CNNs. Theoretically, each cell pertains to any function mathematically. However, it is not easy to implement all the complicated functions in reality. The physical implementation of CNNs depends on whether the cell function or the interaction function can be found in physical devices. For image processing, the commonly used cell function of CNNs is a bi-stable state function. Since the bi-stable state function can be implemented with RTDs, they have been typically used for the physical implementation of the CNNs [1.16][2.6].

Boolean logic can be implemented using RTDs as well. The Boolean logic is realized using the bi-stable state due to its negative differential resistance characteristic of RTDs since the Boolean logic requires two states for functionality. With this structure, basic Boolean functions, AND, OR, and AND·OR gates, can be implemented [1.56]. To enhance the performance, this type of architecture can incorporate other types of devices, for instance, CMOS, HEMT, HBT or single electron transistors. However, the incorporation has been challenging and requires more efforts to overcome fabrication obstacles in the future.

Image processing, which can be improved by parallel processing, needs new types of architectures to improve performance. As each frame of an image cannot be stored within a word bit, it will be more efficient to use multiple processors to hold the data of each frame of an image. After holding this data, the multiple processors need to communicate with each other to process the image. However, if we use sequential processing, the performance will be severely degraded as the image size is increased. As CNNs are parallel processing structures and each repeated circuit cell communicates with each other through the CNN template, image processing with CNN has been widely studied [1.1][1.4].

Image processing with CNN incorporated with an array of RTD structures is a fascinating technique as it uses less area and takes less processing time which will be discussed in Chapter 2. Image processing using an array of RTD structures with CNNs uses the bi-stable state with the negative differential resistance and templates of CNNs. The image processor using CNNs can implement functions such as edge, line detection, smoothing and shift [1.1][2.6]. These functions are implemented by controlling the templates of the CNNs. The templates can be changed in hardware implementation by controlling the device parameters or replacing the devices between cells, which are the smallest unit of CNNs.

Another application of the RTD array structure is velocity tuned filters which are used in motion detectors. Motion detectors are the main function of real time vision machines, which require complex, intensive, and costly computations. The reason is that the real time vision machines are required to monitor images and calculate the correlation between images within a very short time to react to the changes in the environment [1.20], [1.21].

The velocity-tuned filter can be combined with pattern recognizers, optical flow sensors, and noise removers to create real-time vision machines. Even though the velocity tuned filters using spatiotemporal derivatives and Reichardt correlation detectors have been studied, they cannot provide sufficient area compactness, low power consumption and/or, speed [1.3], [1.4]. These problems can be attributed to limitations of conventional analog circuits that cannot achieve high performance for the real-time vision machine. Among nanoelectronic devices, an array of RTD structures show excellent performance in terms of area, power consumption, and speed [1.19], [1.20].

However, using RTD arrays to implement the image processor or the velocity-tuned filter may encounter difficulties in fabrication to achieve uniform cell size. Since the cell size affects the *I-V* curve, its variation will affect the overall functionality as well as the performance of the processors. The methods to overcome this problem and the process variation margin to guarantee the functionality of the processors are discussed in Chapter 2.

Memory architectures are also confronted with scaling problems and require new nanoelectronic devices and corresponding system architectures. Crossbar memory devices [1.30]-[1.33] in which an active material is sandwiched between two sets of conducting nanowires crossing each other, show properties to address high performance and small area occupation. The performance, area and cost of the crossbar memory architecture are compared with those of conventional volatile and non-volatile memory architecture in Chapter 4.

The crossbar structure offers many advantages as memory devices because of its simple two-terminal design. With its simple design, terabit-scale memories can be realized with the width of the metal lines of the array less than 5 nm. The crossbar structure can be non-volatile based on the active material between the two sets of conducting nanowires. This non-volatile characteristic provides instant restart, longer standby operation resulting in elongated battery life.

1.2. Background

1.2.1. RTD fabrication

A double barrier structure is constructed by the formation of a single quantum well structure sandwiched by two very thin tunnel barriers. In the quantum well, carriers have discrete energy levels. This structure can be fabricated by using AlAs/InGaAs or InAlAs/InGaAs with molecular beam heteroepitaxy technology.

A typical material to create the double potential barrier in the conduction band or valence band is III-V compound material. Heterojunctions can be made of various III-V compound semiconductors for the double potential barrier.

Si/SiGe materials can be used to form the double potential barrier. However, the peak to valley current ratio of RTDs using Si/SiGe is poor due to the limited conduction band and valence band discontinuities between Si and SiGe. Since the valence band discontinuity in Si/SiGe is larger than the conduction band discontinuity, the heterojunctions using resonant tunneling of holes were implemented at first. However, the NDR characteristic was only observed at low temperature. The heterojunctions using resonant tunneling of electrons were realized later and show 2 - 3 peak to valley current ratio at room temperature.

1.2.2. Cellular neural network

A CNN was introduced by Leon O. Chua and Lin Yang in Berkeley in 1988 [1.3]. CNNs are a parallel computing paradigm and can be defined as array of cells. An individual cell can be a processor which makes a CNN architecture and also a device which has some function to represent the device characteristics.

Let's assume a CNN which consists of an M x N retangular array of cells C(i,j) with Cartesian coordinates, I = 1,2,…,M, j = 1, 2, …, N as shown in Figure 1.2. Then, the area

of influence, A(i,j), of cell C(i,j) is defined to be the set of all the surrounding cells with radius r, as given in Eqn. (1.1)

$$A(i,j) = \{C(k.l) \mid \underset{1 \le k \le M, 1 \le l \le N}{MAX} \{|k-i|, |l-j|\} \le r \tag{1.1}$$

Then, a class M x N standard CNN is defined by an M x N rectangular array of cells C(i,j) located at a site (i,j). The individual cell can be represented as

$$x'_{ij} = -f(x_{ij}) + \sum_{C(k,l) \in A(i,j)} A(i,j;k,l) y_{kl} + \sum_{C(k,l) \in A(i,j)} B(i,j;k,l) u_{kl} + z_{ij} \tag{1.2}$$

where x', $f(x_{ij})$, y_{kl}, u_{kl}, and z_{ij} are differential function, intrinsic function, output, input, and threshold of a cell C(i,j) respectively [1.44]. In Eqn. (1.2), B(i,j;k,l) represents a feedforward template and A(i,j;k,l), a feedback template. These templates can be chosen to be any function to define the current state.

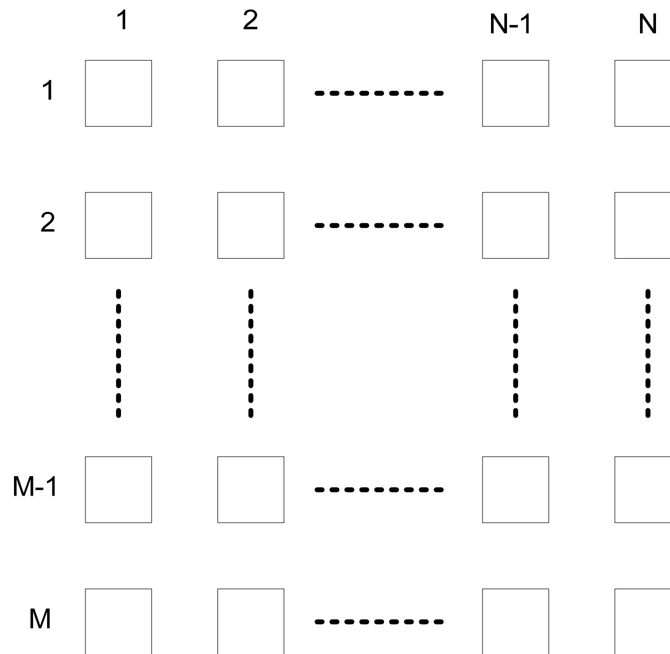


Figure 1.2: A CNN architecture consisting of an M by N array of cells.

Synaptic signal flow graph representation is a good example to show how CNNs work with the feedback template and the feedforward template as shown in Figure 1.3. These two templates demonstrate the directions of signal flow from neighboring cells and their synaptic weights a_{kl} and b_{kl}, respectively. In Figure 1.3, the feedforward template represents the synaptic current sources controlled by the inputs of surrounding cells and the feedback template controlled by the outputs of surrounding cells. The state of the cell changes based on the feedback and the feedforward current, input, current state, and threshold current of the cell.

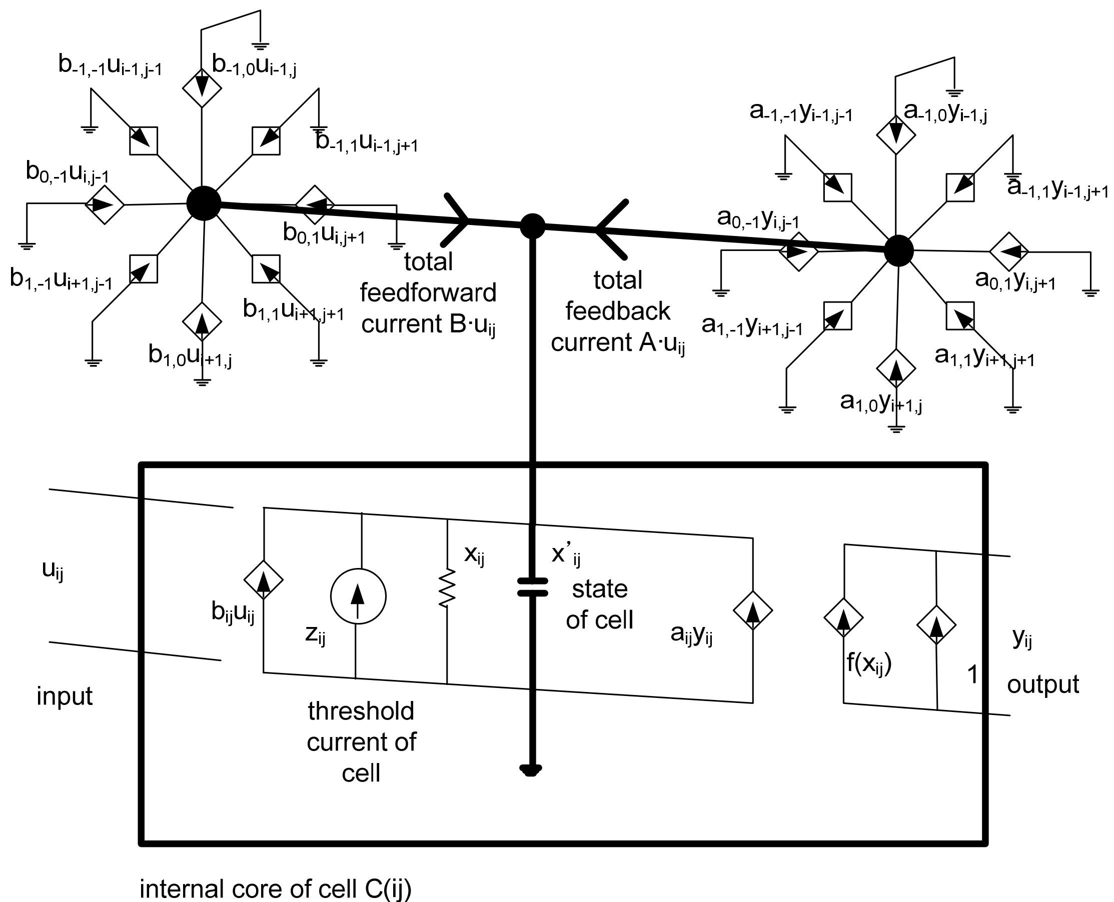


Figure 1.3: Cell realization of a standard CNN cell C(i,j) [1.44].

1.2.3. Image processing using an array of RTD nanostructure

Based on the basic logic gate application using an array of RTD nanostructure, an image processing technique using the nanostructure has been suggested intensively in the last decade [1.1][1.16][1.23]. In this study, more analytical methods, for instance, device characterization and system equations were suggested. Edge detection, smoothing, and noise reduction capabilities of the structure were implemented using the analytical methods. Reference [1.1] represented a new image processor using an array of cells on a resonant tunneling diode substrate as shown in Figure 1.4. Basic image processing function was realized with resistive connections between each cell. The resistive connections work as averaging the values of neighboring cells in the array. This structure can be regarded as CNNs since each metal island works as a cell in CNNs and the resistive network operates as a CNN template. However, reference [1.1] does not provide versatile image processing functions using the structure.

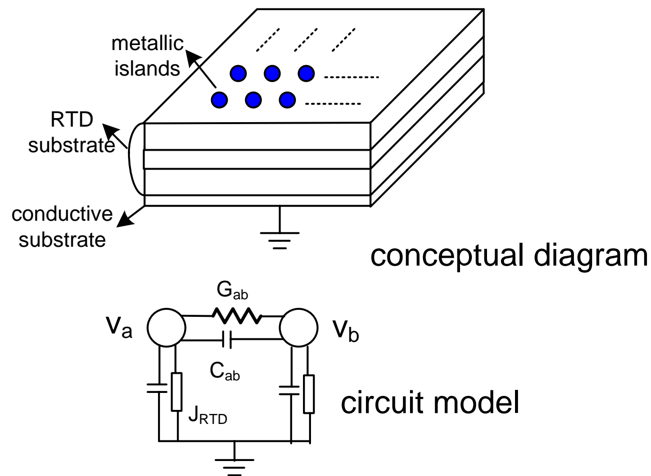


Figure 1.4: A conceptual diagram and circuit model of an array of RTD image processor [1.1][1.16].

Reference [1.16] demonstrates the feasibility of CNNs using an array of nanoscale quantum dots which mimic RTDs to image processing. In this work, image smoothing, edge detection and noise reduction capabilities of the nanostructure using controlling the conductance and capacitance between cells as shown in Figure 1.4. However, this work can not process complicated images which are various colors. Also, this work does not provide any performance measurement or reasonable modeling to calculate the performance. Reference [1.23] proposes and analyzes a cellular nonlinear network based on nanostructure. Each cell in the structure is composed of two RTDs in series through a metallic dot. They demonstrated image processing capabilities as examples of erosion, dilation, and edge detection with a 400 x 269 cell templates. This image processing was implemented with controlling cell templates. In this work, RTDs are used for interconnection between each cell and the metallic islands are used for the connection between RTDs. However, this work demonstrates image processing on black or white images. Also, the structure of this image processor is much complicated compared with other image processors based on an array of cells. This increases the cost to fabricate and makes this processor unrealistic. In addition, the functionality is limited to the function of RTDs since they are the only interconnection device in the processor.

1.2.4. Velocity-tuned filter

Studies of velocity tuned filters for real time vision machines have been conducted in the last decade vigorously [1.20][1.21]. Since the real time vision machines need to implement a large set of filters, they require large arrays of filters. The velocity tuned filters are a main part of motion estimation. The velocity tuned filters are simple, regular architectures based on analog circuits. Two methods are utilized for motion estimation. Gabor type filters are used in the one approach [1.24][1.25]. In the space-time domain, image velocity can be considered as an orientation, which makes Gabor filters model cortical cells' velocity and direction of movement [1.26]. Even though Gabor filters outperform the previous approaches, they require intensive computations [1.27]. To overcome this problem, a new type of Gabor filters was suggested, which uses CNNs to

implement the filters. However, they require a second order differential amplifier for each cell which is inefficient in the area, power, and processing time. Like the energy estimation based velocity tuned filter, Gabor filters uses the structure of CNNs which requires area efficient structure. Therefore Gabor filters which use a second order differential amplifier for each cell is not suitable to the real time vision machine.

The other method uses energy based estimation, which uses wide band velocity tuned filters. Energy based estimation requires calculations of energy from the frequency elements of filtered outputs. Additionally, it requires shunting inhibition circuitry to linearize output differences from filters as suggested by Torralba [1.21][3.5]. Therefore, this velocity tuned filter requires complicated circuitry and shows poor performance in terms of area, power and speed. Especially, the shunting inhibition circuitry requires a complicated analog circuitry which can not be fast enough for the real time vision machine. The area also affects the number of functions for the velocity tuned filter. The smaller the area, more functions can be implemented in the limited die area. Since the velocity tuned filter uses CNNs structure, power dissipation is also hot issue. Basically, CNNs can be regarded as a parallel processing processor if we consider each cell as a processor of parallel processing processors. The power dissipation will be increased as the number of cells is increased. Hence, the structure which uses CNNs should be tiny as much as possible to have reasonable power dissipation to be used realistically. For this reason, a new velocity tuned filter which uses small area, low power dissipation and fast processing time is suggested in Chapter 3.

1.2.5. Crossbar memory

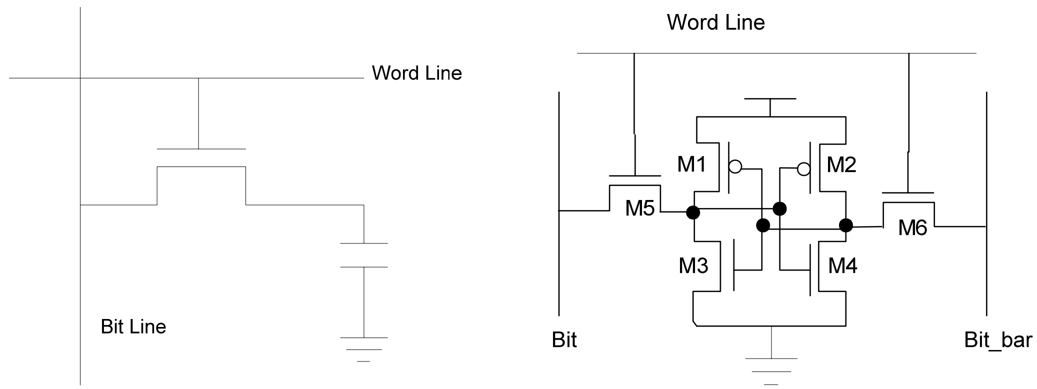
There are several ways to store data on a memory chip. DRAM and SRAM store data using charge on a capacitor which is inherent in a transistor. MRAM stores data using a magnetic storage element which consist of two ferro-magnetic plates. In DRAM and SRAM, charge is stored when current flows through the transistor which is connected to

a word line. However, MRAM writes data when a magnetic field which was induced from current through the write word line, affects ferro-magnetic plates.

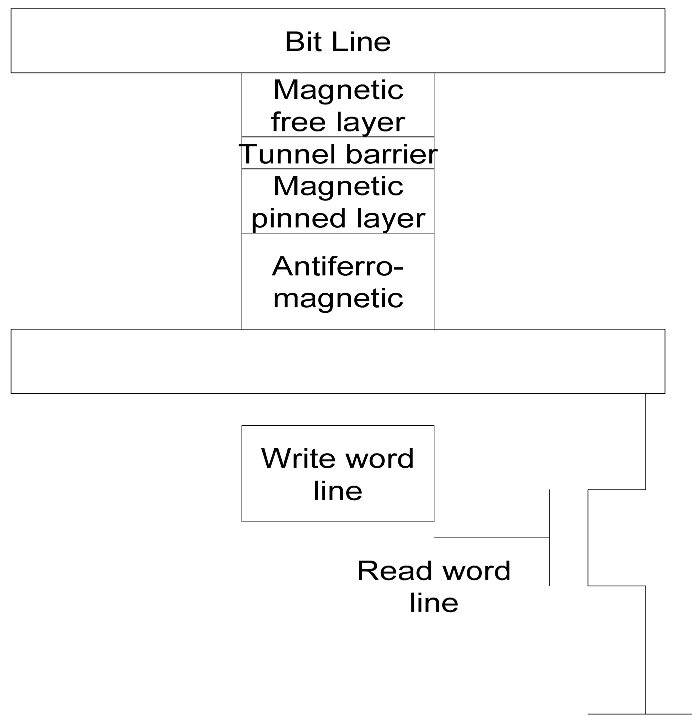
As shown in Figure 1.5 (a), a word line is enabled and then bit line data affect the charge in the capacitor when DRAM writes data. When it reads data, the bit line is pre-charged and then the word line is enabled. The bit line is affected by the charge on the capacitor. Finally, a sense amplifier magnifies the change of the value of the bit line and decides it is “1” or “0” [1.46].

In SRAM, the writing and reading are conducted through careful sizing of the transistor. In the reading mode, if we assume bit line is “1” and the drain node of M1 and M2 is “0”. The issue is to prevent the bit line data from affecting the drain node reversely. To avoid this case, we need to make M3 stronger than M5. In the writing mode, if we assume bit_bar is “1” and the drain node of M2 and M4 is “0”. We have to prevent the drain node value from affecting the bit_bar’s value. To avoid this case, we should make M6 stronger than M2. Therefore, M3 and M4 should be strongest and M1 and M2 weakest and M5 and M6 middle respectively [1.46].

In MRAM, writing is conducted by current flow through write word line and the bit line. This current flow generates magnetic field and affects the magnetic fields of the two ferro-magnetic plates. In reading, we use the resistance change of the cell from the magnetic field. By generating a sense amplifier circuit detects the resistance change for the reading. [1.45].



(a) (b)



(c)

Figure 1.5: Memory cells of DRAM (a), SRAM (b) and MRAM (c).

Crossbar memories are reconfigurable architectures where an active material is sandwiched between two sets of conducting nanowires crossing each other. Using the hysteric characteristic of the active material, crossbar memory architectures were implemented [1.48][1.30]. The crossbar memories write data by applying voltages high or –high across the crossbar memory cell. It is not easy to implement general logic operations with the crossbar structure by itself since the structure has difficulty in achieving enough gain and inversion. To surmount this difficulty, hybrid CMOS/crossbar structures have been proposed [1.34][1.35]. With these structures, the ultra-high device density is maintained while the flexible control on the crossbar arrays was provided by the CMOS circuitry. Here, the constraint in attaching the CMOS circuit to crossbar array is how to match the decoder length to the width/length of the array.

Rigorous investigation on the feasibility of volatile and nonvolatile switching devices on Si substrate has been conducted [1.36]-[1.40] because such devices can be fully fabricated with CMOS processing technology. These devices are not costly to fabricate and have a simple structure to operate. So, there have been intensive researches on crossbar application design using crossbar arrays [1.48]-[1.51].

Figure 1.6 shows a simple 4 x 4 memory design using nanoscale molecular-switch crossbar circuits. This memory accesses a cell through a column decoder and a row decoder. The column and row decoders are programmed by changing the resistance of the specific crossbar point which is shown with black color in Figure 1.6. The address bits makes one column and row have low voltage, while other columns and rows have high voltage. This makes the external power goes through the selected column and row to write and read data in the array.

However, the simple crossbar memories in Figure 1.6 overuse power sources since it uses 3 power sources to apply + high and –high. This will increase power dissipation more than when we use 2 power sources. Also, this circuit uses AC voltage for reading data

which is not compatible to normal digital circuits and requires additional circuits to convert it to digital values.

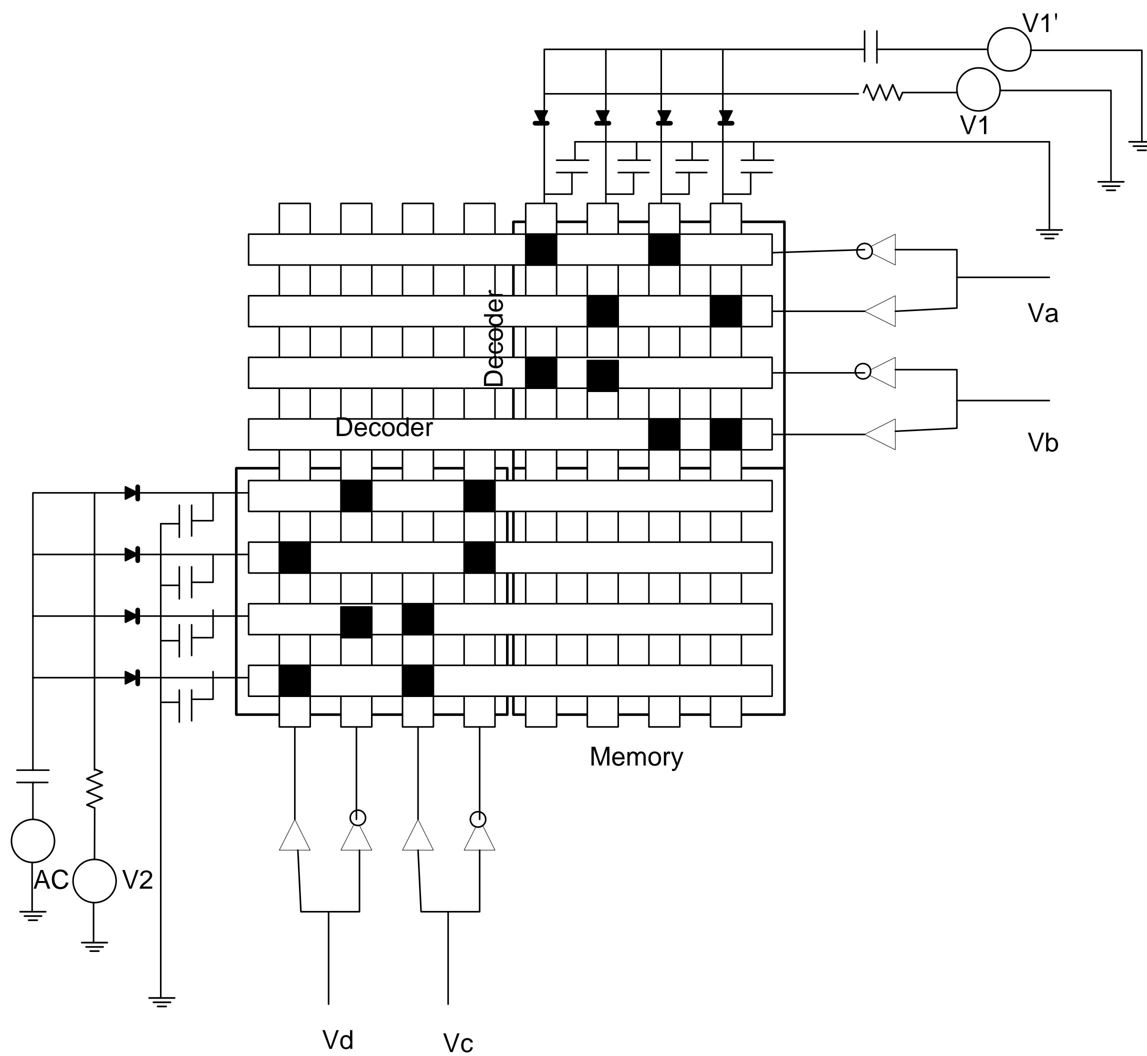


Figure 1.6: A simple crossbar memory architecture for nanoscale molecular-switch crossbar circuits [1.51].

Figure 1.7 shows another crossbar memory which was suggested from DeHon *et. al* [1.48]. This circuit combines the conventional CMOS circuits incorporated with a crossbar memory array. Reading and writing are operated according to the column and row address bits. For reading, all the rows are precharged and make the selected row drive the current through the path which is assigned by the row and column addresses.

However, this memory design uses also 3 power sources for writing which is inefficient in power management. Since this design uses pass gates for decoders, there is problem regarding charge sharing issue in the floating node. In addition, applied voltages will be dropped by the threshold voltage when it arrives at the memory array. Since it also precharges all the rows when the read operation works, leakage current will be large as the size of array increases. Also, the modeling of memory cell is not correctly designed to represent a real device since interconnect and internal capacitance elements should be considered.

The crossbar memory design which will be discussed in Chapter 4 demonstrates a unique design which solves all the problems mentioned above. The new design models the crossbar memory cell based on physical parameters and provides new decoders which do not have the problems revealed in previous crossbar memory designs. Power control logics which are not introduced in previous designs are described in Chapter 4. New system designs with a 1Kb crossbar memory system including all the parasitic components are introduced. New sense amplifiers are demonstrated which overcomes the problems shown in the previous crossbar memories. Noise analysis and power dissipation modeling is announced newly introduced in Chapter 4.

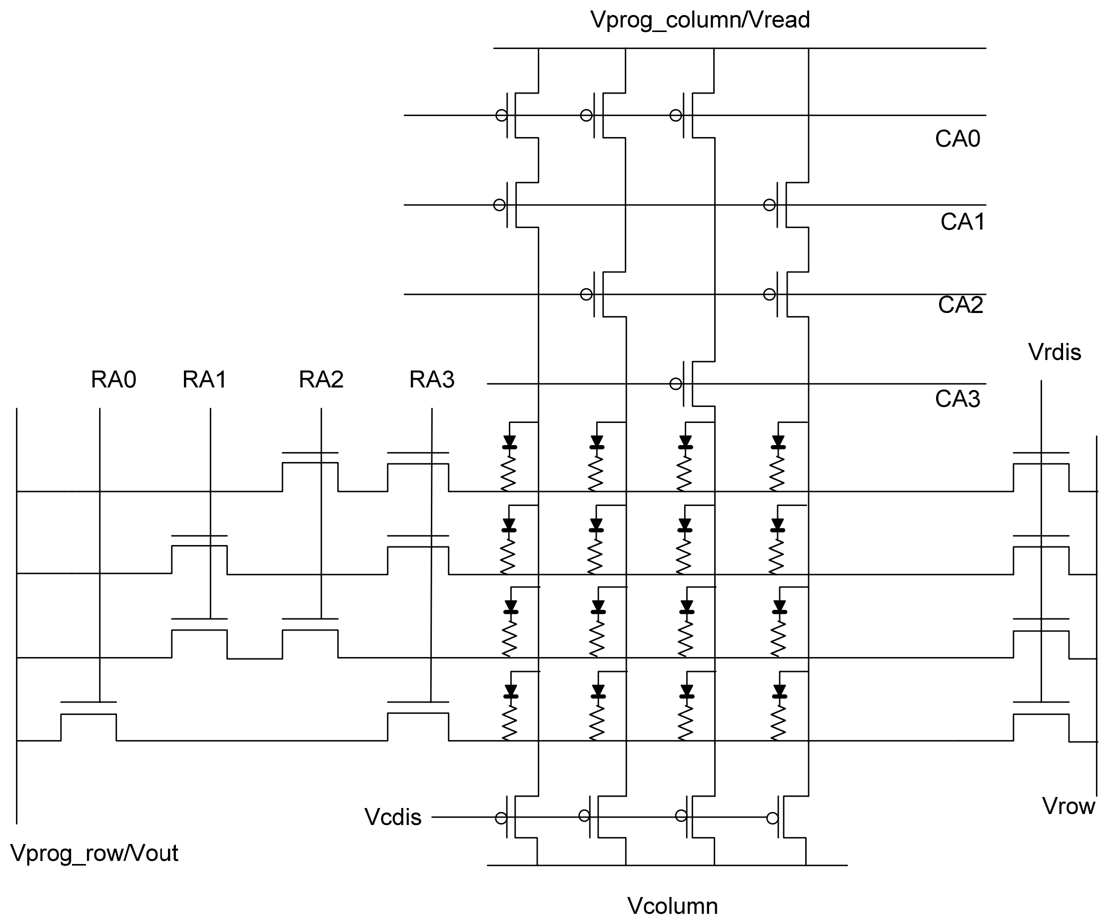


Figure 1.7: A crossbar memory architecture based on nanowire transistor [1.48].

1.3. Thesis Overview

1.3.1. Image processing using an array of RTD nanostructure

An arrary of RTD nanostructures are fast and compact, and form the basis of a promising new approach to image processing. To date, binary-output image processors, which are based on resonant tunneling diodes, have been thoroughly studied. However, it is impossible to process color images using a conventional cell network and research has been focused on gray scale image processing with conventional CMOS circuit design along with CNNs [1.51]. We suggest a new cell network that implements color image processing. Our new cell structure is formed by a metallic island on the surface of a multi-peak resonant tunneling diode. We suggest two methods, color mapping and RGB value, to represent color values. Using both methods, we realize desired effects including quantization, smoothing, and color extraction. In addition, we analyze the changes in voltage and current of the cells that hold pixel information in the device. In addition, the effect of *I-V* curve deviation due to the process variation on the color image processing is discussed in Chapter 2. A preliminary version of this work has appeared in [1.54].

1.3.2. Velocity-tuned filter

We propose a new velocity tuned filter used for unmanned vehicles that uses real time motion estimation. Since this application requires tiny, energy efficient and speedy architecture, we demonstrated a new velocity tuned filter based on nanoelectronic architecture. In the nanoelectronic architecture, a velocity tuned filter was implemented on a CNN based cell structure on a resonant tunneling diode. The simulation results show that the new velocity tuned filter is 10, 100, and 1000 times better than conventional velocity tuned filters in area, power consumption, and speed respectively. Since the nanoelectronic architecture embraces nonlinear circuits, a stability analysis was

performed. From the analysis, we concluded that the new velocity tuned filter is asymptotically stable where the derivative of the *I-V* curve of the resonant tunneling diode with respect to *I* is greater than or equal to 0. Comparison of our analysis approach with other analyses is discussed. The effect of *I-V* curve deviation due to the process variation on the velocity tune filter and is discussed in Chapter 3 [1.53][1.55].

1.3.3. Crossbar memory architecture

Two terminal resistive switching devices based on the M/a-Si/p-Si device structure have been found to be a strong candidate for the crossbar architecture [1.40][1.41]. These devices propose ultra-high density and intrinsic defect tolerable capability. Also, the M/a-Si/p-Si devices depict comparable scalability and significantly better performance compared with the crossbar structure devices based on molecules [1.42] which suffer from poor electrical properties with extremely low yield, small on/off ratio, poor thermal stability and slow switching speed. The poor thermal stability of molecules also results in reliability problem [1.42].

Since the density of crossbar memory is determined by the size of the top and bottom electrodes, we study the new structure, the M/a-Si/p-Si crossbar devices and the feasibility to apply the structure to ultra-high density memory architecture. To apply it to memory system, electrical modeling of the devices is conducted and the hybrid CMOS/crossbar structure is used to implement the peripheral circuitry in the memory system.

In implementation of peripheral circuitry, manipulating sense amplifier position and constitution and sizing the number of column decoders to achieve maximum performance are discussed. In addition, scaling issue on the crossbar memory design due to the size mismatch between crossbar memory arrays and our suggested sense amplifiers is discussed in Chapter 4.

CHAPTER II

2-D Array of Multi-Peak Resonant Tunneling Diodes Based Color Image Processing

2.1. Introduction

Since 1987 when the Cellular Neural Networks (CNN) was invented by Chua and Yang [2.1], it has been used as a powerful analog array processor. Image processing, motion detection, pattern recognition, and real time processing are the main applications of CNN paradigm of local computation over a sphere of activity, also known as a neighborhood of a processing element. To process such functions in real time, massively parallel computation of spatial data over a two-dimensional area is required. This requirement is satisfied by CNNs due to their array of regularly spaced and locally connected cells that function as a nonlinear dynamic system.

Resonant Tunneling Diode (RTD) provides folded-back or negative differential resistance (NDR) *I-V* characteristic owing to quantum tunneling through its mesoscale double-barrier quantum well when a bias voltage is applied across its two terminals. This characteristic reduces the device count in a cell. In addition, RTDs have pico-second switching speed due to the small intrinsic capacitance. RTDs have been deployed in the past to design high-speed and high-density integrated circuits [2.2]-[2.5]. RTDs have been utilized to build CNNs with improved speed and integration density [2.2].

Research works on an array of RTDs have been conducted to provide efficient area utilization of CNNs and RTDs [2.6][1.1]. This structure is known as a compact structure and has fast processing time due to the small intrinsic capacitance and NDR characteristic. This is why the structure has been used for image processing and Boolean logic circuitry.

However, the research on image processing which uses such a structure with the CNNs on the RTDs has revealed its limitation in image processing. Image processing using this structure has been confined to black or white images due to a bi-stable state of the RTDs.

Therefore, it is inappropriate and impossible to process color images which need more than one bit per pixel. With black or white image processor, it is impossible to remove noise from color images and to extract colors that we want in color images. Hence, we suggest a new type of a array of multi-peak RTDs which can be fabricated by vertically stacking double-barrier (one-peak) RTDs. Color image processing with the multi-peak RTDs is now possible due to this array structure. Since we assume the radius of the metallic islands more than 20 nm, a continuous model can be applied to the metal dots [1.1][2.10]. Hence, *I-V* characteristic of an array of metallic islands depends directly on the multi-peak RTDs.

In this chapter, we suggest a new type of an array of multi-peak RTDs which can be fabricated by vertically stacking double-barrier (one-peak) RTDs for color image processing. In this chapter, we use a model of an eight-peak RTD which was experimentally confirmed [2.7]. Nine stable states exist in this multi-peak RTD. To exploit the nine stable states for the color processing, we adopt two methods of color representation. One uses a color map to index the color value, while the other method encoded as RGB (Red, Green, Blue) uses three cells to represent one pixel color. Using color depiction methods and characteristics of multi-peak RTD's, we describe diverse color image processing methods to achieve quantization, smoothing, and color extraction with various image sizes.

2.2. Multi-Peak RTDs Based Color Image Processor

Conventional image processors using an array of RTDs have utilized a bi-stable state of resonant tunneling diode [1.16][1.1]. The bi-stable state decides the output image and processing methods. Using parallel processing and small intrinsic capacitance, fast image processing is possible with this structure.

The disadvantage is that it is impossible to represent various functions, to generate color images, and to process those color images because three stable states are required to display color information.

Multiple stable states are essential to represent the color information due to the various color values. Hence, a device which has those functionalities should be considered. In this sense, the multi-peak state resonant tunneling diode is considered to be a proper candidate for such a device. The multi-peak resonant tunneling diode has 9 stable states in our simulation. This device can be fabricated by stacking serially eight pseudomorphic AlAs/In0.53Ga0.47As/InAs RTDs as shown in Figure 2.1 [1.1]. Each RTD can be modeled as R, C and voltage controlled current source in the circuit simulation.

In Figure 2.1, the interconnection between two multi-peak RTDs is implemented with RC models. This RC models changes based on the image function that the processor realizes. In the CNN perspective, the changes of RC model parameters correspond to the change of the templates in CNNs.

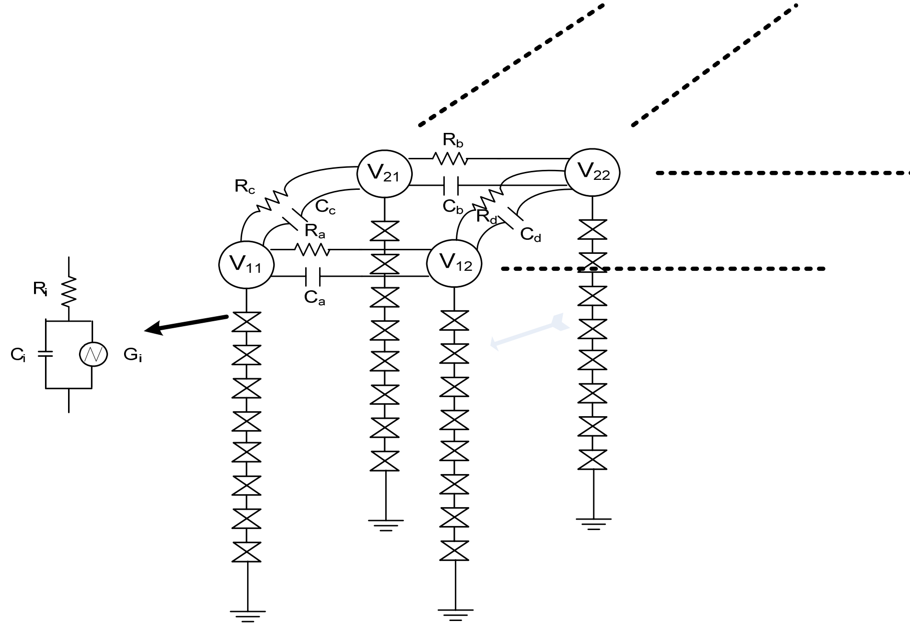
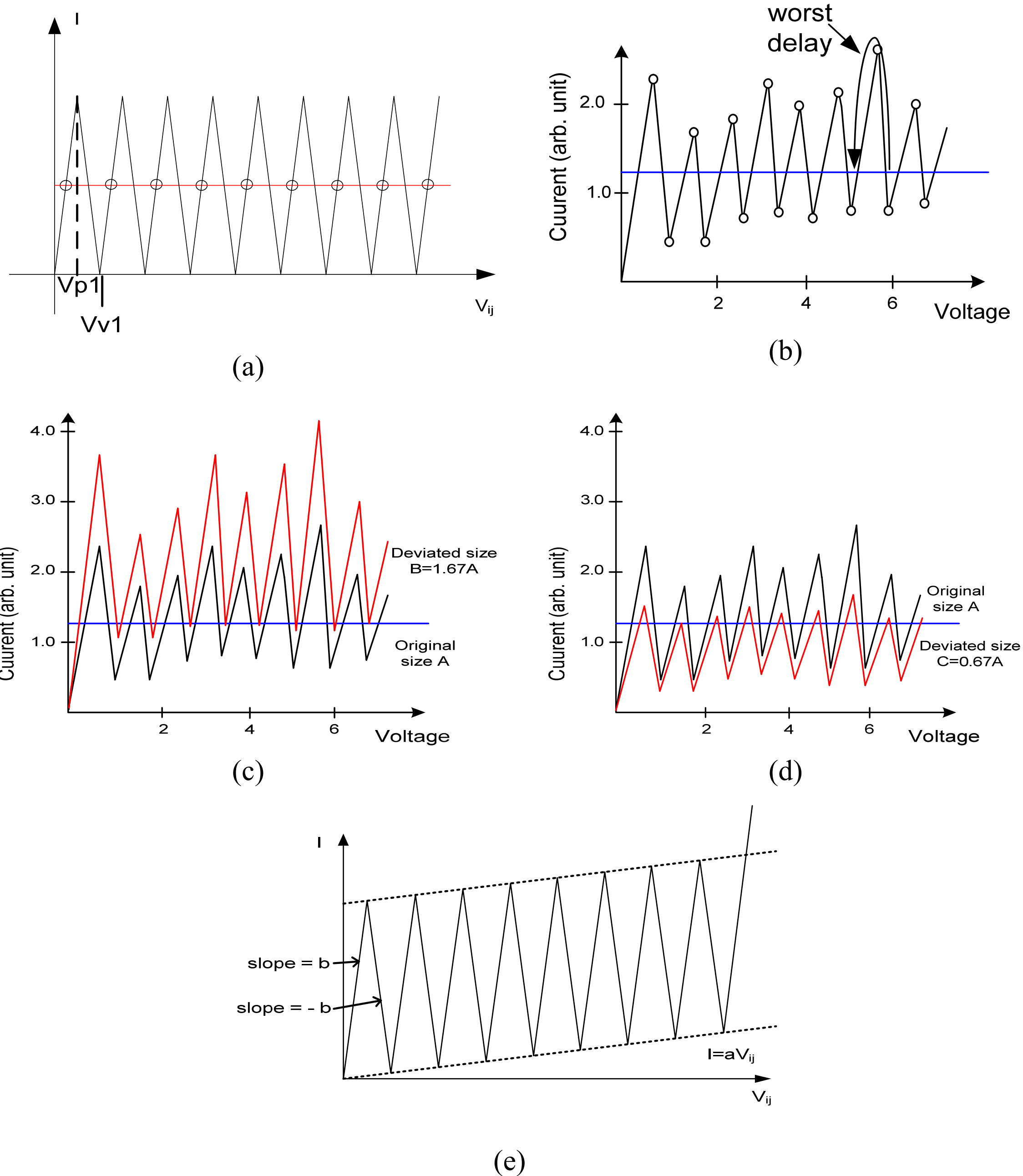


Figure 2.1: A multi-peak RTD based image processor.

The *I-V* characteristics of a multi-peak RTD in the vertical direction can be simply piecewise-linearly modeled as shown in Figure 2.2(a). Since the eight-peak RTD is used, nine stable states exist when using constant current source that delivers current in the range between the lowest peak current and the highest valley current. However, the *I-V* curve of experimental data shows a DC shift as shown in Figure 2.2(b). To be valid in functionality, the lowest peak should be higher than the highest valley. These peak or valley values also change based on the size of the metallic islands. Since the size of RTDs varies depending on the process variation, we need to consider a shift of *I-V* curve to up and down in the direction. Based on the experimental result from [2.7], if the size deviation of the metallic islands is 67% to bigger size or 33% to smaller size, the functionality is guaranteed. The process variation also affects the power dissipation and speed. However, the overall speed and power dissipation is not much changed if we assume the randomly distributed inputs. The worst prixel speed is degraded around 30 ps from Eqn. (2.9) and the power dissipation of the worst pixel is increased around 0.2 nW from Eqn. (2.14). The overall speed and power dissipation is more dependent on the initial value of output and templates as shown in Figure 2.5, Figure 2.9 and Figure 2.16.



(e)

Figure 2.2: Simplified piecewise-linear model of *I-V* characteristics for an eight-peak RTD in a vertical direction (a), piecewise-linear model based on experimental data (b), maximum size deviation for valid functionality (c), minimum size deviation for valid functionality (d) and piecewise-linear model with a linear DC shift (e).

The standard CNN state equation can be written as follows.

$$\frac{dx_{ij}}{dt} = -f(x_{ij}) + \sum_{k,l \in N_{i,j}} (a_{k-i,l-j} f(x_{kl}) + b_{k-i,l-j} u_{kl}) \tag{2.1}$$
$$+ I_{ij}.$$

where $X_{ij} \in R$, $f(X_{ij}) \in R$, $u_{kl} \in R$, and $I_{ij} \in R$ denote the state, output, input, and threshold. *a(i,j;k,l)* and *b(i,j;k,l)* are called the feedback and the feedforward operators or templates, respectively. Considering the network connection between cells and neglecting the feedforward effect, Eqn. (2.1) can be transformed into Eqn. (2.2) below which will be used for quantization, smoothing, and color extraction. The left term shows the current change on a capacitor and the right term represents the current from neighboring cells, external current source, I_{ij}, and internal current dissipation. The color image processing is realized by the current voltage characteristic which is represented by $J(v_{ij})$. Since it has nine stable states in the current voltage characteristic, we use these nine stable states as color values.

$$\frac{C dv_{ij}}{dt} = -J(v_{ij}) + \sum_{k,l \in N_{i,j}} (a_{k-i,l-j} v_{kl}) + I_{ij}. \tag{2.2}$$

where I_{ij} is the driving current, and C is the contact capacitance of each group of RTDs and the feedback template

$$a = \begin{bmatrix} 0 & q & 0 \\ q & -4q & q \\ 0 & q & 0 \end{bmatrix}.$$

In the cases of quantization and smoothing function, the template for feedback is represented as a_{ij} in Eqn. (2.2). The template for the quantization function and smoothing function is decided by the conduction from the neighboring cells to the target cell. In the above example, there is a target cell which has connections with four neighboring cells

with q conduction value. The center cell has -4q conduction value which means that it reversely dissipates the incoming current. By the value of q in the template, the quantization function or smoothing function is decided. The quantization and smoothing function are realized with resistive network connection with the defined neighboring cells in the template. So, the current flows from the four neighboring cells to the center cell or reverse direction. The center cell circulates the received currents through itself to satisfy the KCL. Also, this current flow depends on the outputs of the neighboring cells. Hence, the template a_{ij} is represented as a matrix which has q in left, right, up and down current flow and -4q for center cell. The value of q is a conductance value in real number which changes depending on image functions.

The *I-V* curves of Figure 2.2 (a), (b) and (e) can be represented as Eqn. (2.3), Eqn. (2.4a) and Eqn. (2.4b), respectively. Even if Eqn. (2.4a) demonstrates more realistic curve form based on experimental data, Eqn. (2.3) is chosen for a model for the eight-peak RTD since the functionality is not affected with simple implementation and the overall processing time changes less than 30 ps (5%) from Eqn. (2.9) and Figure 2.5. This is because the initial voltage and the templates of each cell more affects the processing time than the eight-peak RTD models.

$$J(v_{ij}) = \begin{cases} \alpha v_{ij}, & 0 < v_{ij} \le V_{p1} \\ \alpha(|v_{ij} - V_{v1}|), & V_{p1} < v_{ij} \le 3V_{p1} \\ \alpha(|v_{ij} - 2V_{v1}|), & 3V_{p1} < v_{ij} \le 5V_{p1} \\ \alpha(|v_{ij} - 3V_{v1}|), & 5V_{p1} < v_{ij} \le 7V_{p1} \\ \alpha(|v_{ij} - 4V_{v1}|), & 7V_{p1} < v_{ij} \le 9V_{p1} \\ \alpha(|v_{ij} - 5V_{v1}|), & 9V_{p1} < v_{ij} \le 11V_{p1} \\ \alpha(|v_{ij} - 6V_{v1}|), & 11V_{p1} < v_{ij} \le 13V_{p1} \\ \alpha(|v_{ij} - 7V_{v1}|), & 13V_{p1} < v_{ij} \le 15V_{p1} \\ \alpha(|v_{ij} - 8V_{v1}|), & 15V_{p1} < v_{ij} \le 17V_{p1} \end{cases}, \qquad (2.3)$$

$$J(v_{ij}) = \begin{cases} 3v_{ij}, & 0 < v_{ij} \le 0.8 \\ 0.4 - 10(v_{ij} - 1.0), & 0.8 < v_{ij} \le 1.0 \\ 1.6 + 2.4(v_{ij} - 1.5), & 1.0 < v_{ij} \le 1.5 \\ 0.4 - 4(v_{ij} - 1.8), & 1.5 < v_{ij} \le 1.8 \\ 1.7 + 2.6(v_{ij} - 2.3), & 1.8 < v_{ij} \le 2.3 \\ 0.5 - 4(v_{ij} - 2.6), & 2.3 < v_{ij} \le 2.6 \\ 2.0 + 2.1(v_{ij} - 3.3), & 2.6 < v_{ij} \le 3.3 \\ 0.7 - 6.5(v_{ij} - 3.5), & 3.3 < v_{ij} \le 3.5 \\ 1.7 + 2(v_{ij} - 4), & 3.5 < v_{ij} \le 4.0 \\ 0.7 - 5(v_{ij} - 4.2), & 4.0 < v_{ij} \le 4.2 \\ 1.9 + 1.5(v_{ij} - 5), & 4.2 < v_{ij} \le 5.0 \\ 0.7 - 6(v_{ij} - 5.2), & 5.0 < v_{ij} \le 5.2 \\ 2.3 + 2.7(v_{ij} - 5.8), & 5.2 < v_{ij} \le 5.8 \\ 0.7 - 8(v_{ij} - 6), & 5.8 < v_{ij} \le 6.0 \\ 1.8 + 2.2(v_{ij} - 6.5), & 6.0 < v_{ij} \le 6.5 \\ 0.8 - 3.3(v_{ij} - 6.8), & 6.5 < v_{ij} \le 6.8 \\ 1.9 - 2.2(v_{ij} - 7.3), & 6.8 < v_{ij} \end{cases}, \qquad (2.4a)$$

$$J(v_{ij}) = \begin{cases} bv_{ij}, & 0 < v_{ij} \le c_1/2b \\ -bv_{ij} + c_1, & c_1/2b < v_{ij} \le c_1/(a+b) \\ bv_{ij} - c_2, & c_1/(a+b) < v_{ij} \le (c_3 - c_2)/2b \\ -bv_{ij} + c_3, & (c_3 - c_2)/2b < v_{ij} \le c_3/(a+b) \\ bv_{ij} - c_4, & c_3/(a+b) < v_{ij} \le (c_5 - c_4)/2b \\ -bv_{ij} - c_5, & (c_5 - c_4)/2b < v_{ij} \le c_5/(a+b) \\ bv_{ij} - c_6, & c_5/(a+b) < v_{ij} \le (c_7 - c_6)/2b \\ -bv_{ij} - c_7, & (c_7 - c_6)/2b < v_{ij} \le c_7/(a+b) \\ bv_{ij} - c_8, & c_7/(a+b) < v_{ij} \le (c_9 - c_8)/2b \\ -bv_{ij} - c_9, & (c_9 - c_8)/2b < v_{ij} \le c_9/(a+b) \\ bv_{ij} - c_{10}, & c_9/(a+b) < v_{ij} \le (c_{11} - c_{10})/2b \\ -bv_{ij} - c_{11}, & (c_{11} - c_{10})/2b < v_{ij} \le c_{11}/(a+b) \\ bv_{ij} - c_{12}, & c_{11}/(a+b) < v_{ij} \le (c_{13} - c_{12})/2b \\ -bv_{ij} - c_{13}, & (c_{13} - c_{12})/2b < v_{ij} \le c_{13}/(a+b) \\ bv_{ij} - c_{14}, & c_{13}/(a+b) < v_{ij} \le (c_{15} - c_{14})/2b \\ -bv_{ij} - c_{15}, & (c_{15} - c_{14})/2b < v_{ij} \le c_{15}/(a+b) \\ bv_{ij} - c_{16}, & c_{15}/(a+b) < v_{ij} \end{cases}. \qquad (2.4b)$$

2.3. Color Representation Method

As shown in Figure 2.2 (a) and (b) and Eqn. (2.3), an eight-peak resonant tunneling diode has nine stable states. The nine states can be used to indicate color values. That can be used for processing the color images.

Two methods are suggested to store color images using an array of eight-peak RTDs. One method is to match the voltages of an array of eight-peak RTDs to color index values. In this method, each index value designates each color in a color map as shown in Figure 2.3 (b). This method, called the color map method, is used for color images which have color maps.

However, there is a limitation of color number in this method. The maximum number of colors to represent is 9 because an eight-peak RTD has nine stable states. Despite this drawback, the color map method is easier to perform in fabrication than the other method. The second method uses three groups of eight-peak RTDs to represent one pixel, and each eight-peak RTD corresponds to the red, green, and blue (RGB) colors. The voltage value of each eight-peak RTD indicates the intensity of the RGB color. Since each eight-peak RTD can express 9 different intensities, a total of 729 colors are represented, and the number of colors can be extended by increasing the number of eight-peak RTDs per pixel. This method is called RGB color method, and is used to process bitmap images.

The multiple quantization levels can also be used to provide gray scale to images. If we assume three groups of eight-peak RTDs represent one gray scale, a total of 729 gray levels can be represented for images. A fine resolution can be achievable in the gray images due to the fact above. This will be useful when we implement a smooth function which requires polishing the edges of gray levels in the neighboring pixels.

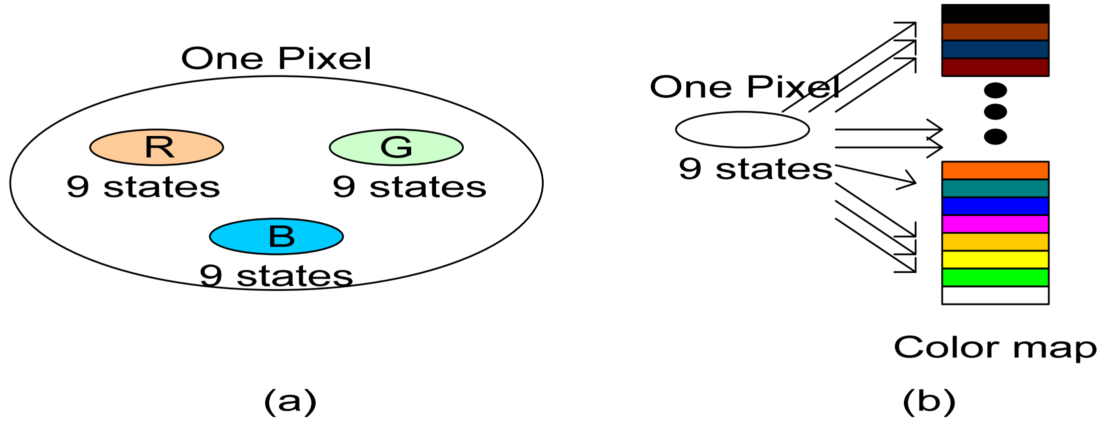


Figure 2.3: A color representation method with eight-peak RTDs. (a) RGB value method and (b) color map method.

2.4. Color Quantization

2.4.1. Implementation and results

Using discrete stable states of multi-peak RTD's, color quantization is implemented. Figure 2.4 represents the changes of colors of a 4 x 4 pixel image. This change of color image arises from *I-V* characteristics of multi-peak RTD's. From the initial color value, the quantized value is obtained by minimum Euclidean distance. From Eqn. (2.2), the quantized color value can be obtained as a function of I_{ij}.

$$J(v_{ij}) = I_{ij} \tag{2.5}$$

$$v_{ijk} = J^{-1}(I_{ij}) \tag{2.6}$$

where v_{ijk} is the solution of Eqn. (2.3) and k=1,2,⋯,9, where 1 corresponds to the lowest value of solution and 9 corresponds to the highest value of solution in the Figure 2.2. If we assume that the input image value is u_{ij}, the minimum Euclidean distance can be derived as follows.

$$Dist_{\min} = \min_{k=1,..,9}(|u_{ij} - v_{ijk}|) \tag{2.7}$$

The solution of Eqn. (2.3) which has the minimum Euclidean distance is the quantization value. The change from unstable state to stable state is depicted in Figure 2.5. In Figure 2.5, the time to get to the final stable state is confirmed by monitoring the tracks of colors of the images. As shown in the HSPICE simulation results, the color values are stabilized after 200 ps. The images representing the HSPICE simulation results show the same characteristics. Based on the simulation, the minimum connection resistance between metallic islands is 250 MΩ to implement quantization.

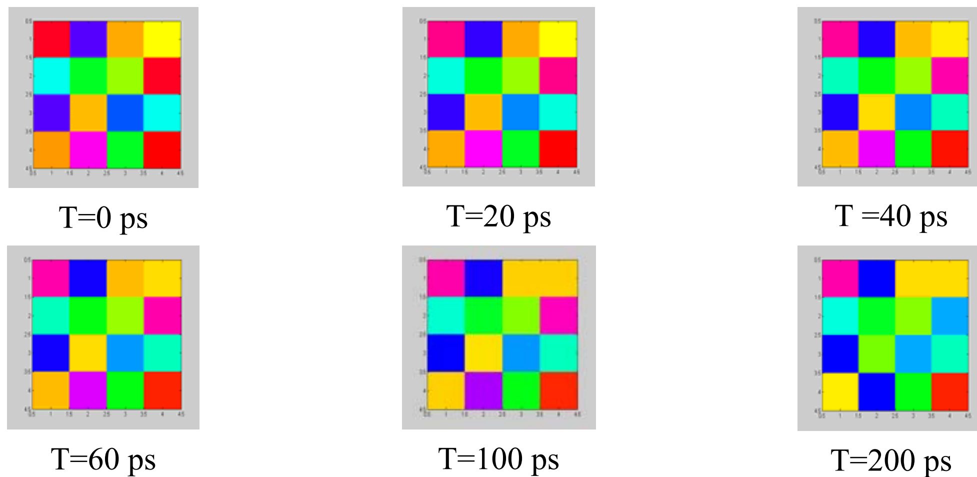
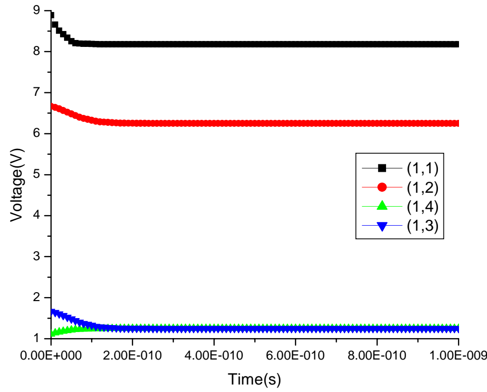
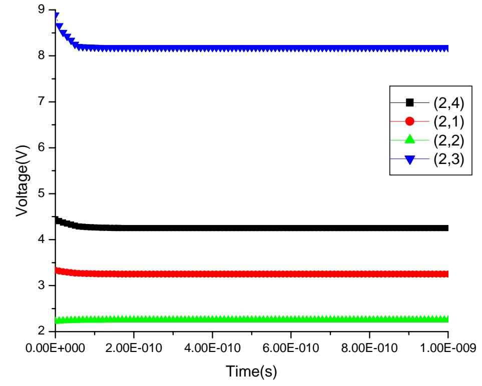


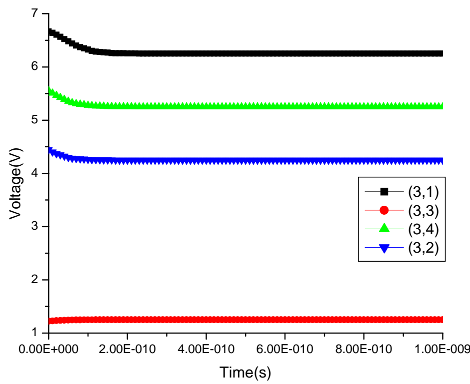
Figure 2.4: Transient representation of 4 x 4 color image quantization from 0 ps to 200 ps.



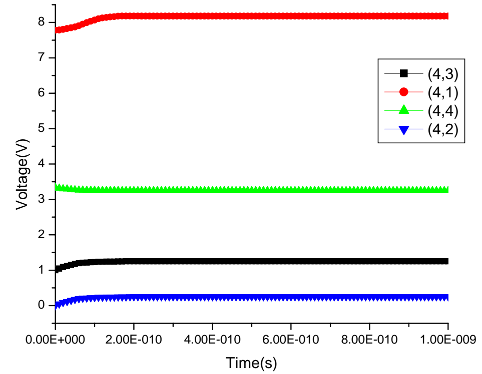
(a) First row



(b) Second row



(c) Third row



(d) Fourth row

Figure 2.5: HSPICE simulation results divided into rows with time from 0ns to 1ns.

The feedback template and feedforward template for the quantization function are represented as

$$a = \begin{pmatrix} 0 & k & 0 \\ k & -4k & k \\ 0 & k & 0 \end{pmatrix}, \quad b = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

where k is conductance between the neighboring cells and the center cell. The typical value for k is greater than 0.01 μmho. External current value should be chosen to be between the lowest peak and the highest valley in the *I-V* curve.

The algorithm of quantization function which was implemented using multi-peak RTDs can be represented using a pseudo code as shown below.

```
PROCEDURE quantization(g_red, g_green, g_blue: VECTOR of color):
                    VECTOR of color;
 * quantize RGB value of the image *
VAR res: VECTOR of color;
BEGIN
    res.red = ROUND(g_red)/9;
    res.green = ROUND(g_green)/9;
    res.blue = ROUND(g_blue)/9;
RETURN res;
END quantization
```

In this code, the input variable and output variable are the vectors of the color. These vectorized variables are quantized in parallel since it takes the input vector and generates the output vector after quantization processing. Input variables consist of RGB color elements which are processed separately. The process is conducted by rounding the input vectors after dividing by 9. The number 9 comes from nine stable states of the multi-peak RTD. As the number of stable states changes, the divider is changed in the algorithm. The input vectors are divided by 9 since we have 9 stable discrete states as shown in Figure 2.2. After dividing by 9, the ROUND function operates as quantization which assigns the closest integer value from the input value. Therefore, each RGB elements can have 9 different values so the total number of colors after the quantization is 729. This image processing can be used for grouping the colors which have similar color values.

On the other hand, the algorithm of quantization image processing based on single conventional processor can be represented using a pseudo code as shown below. In this

code, the input variable and output variable are scalars in color representation value. In this algorithm, we have two procedures. One is for the scalar quantization and the other one is for the processing the whole array sequentially. These functions are represented as scalar quantization and quantization. Using FOR loops, the sequential processing is implemented. The number 4 comes from the pixel size of the image which has 4 by 4 pixels.

```
PROCEDURE scalar_quantization(g_red, g_green, g_blue: SCALAR of color):
                    SCALAR of color;
 * quantize RGB value of the image *
VAR res: SCALAR of color;
BEGIN
    res.red = ROUND(g_red)/9;
    res.green = ROUND(g_green)/9;
    res.blue = ROUND(g_blue)/9;
RETURN res;
END scalar_quantization
PROCEDURE quantization(img: scalar of color): VECTOR of color;
* vectorize the scalar of color *
BEGIN
    for(i =1; i<=4; i++)
    BEGIN
         for(i =1; i<=4; i++)
         BEGIN
              RETURN scalar_quantization(img.red, img.green, img.blue);
         END
     END
END quantization
```

The trace of 50 x 50 color image with time from 0ns to 1ns is depicted in Figure 2.6. Each pixel represents a color using an image value, and this color value is quantized due to the multi-peak RTD. The color change of the eyes and the mouth of the duck in the Figure 2.6 originate from the limitation of the numbers of stable states of the multi stable state resonant tunneling diode.

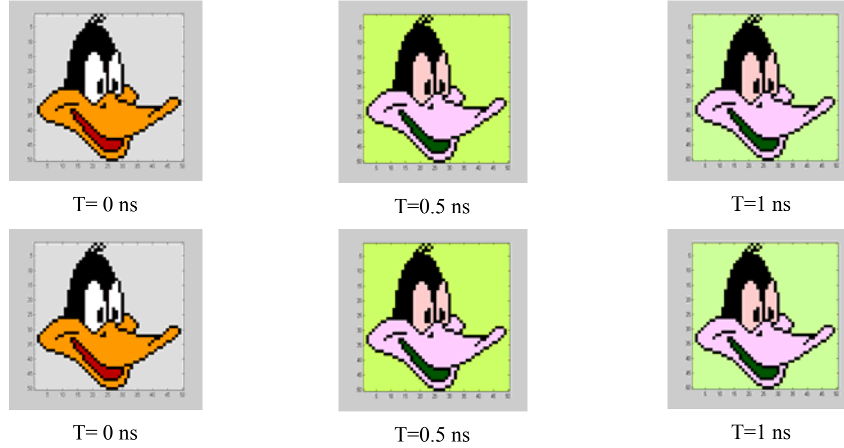


Figure 2.6: 50X50 pixel color image quantization from 0ns to 1ns.

2.4.2. Settling time analysis

The settling time of an array of multi-peak RTD structure is defined as the time needed to reach a stable state from the initial condition value. The stabilization time is determined by the slowest cell in the CNN. In Eqn. (2.2), if we assume the initial voltage is v_0, then the solution of the first order differential equation is given as

$$v(t) = \left[v_0 - \left(\frac{I_{ij}}{\alpha + \sum_{k,l \in N_{i,j}} a_{k-i,l-j}} \right) \right] e^{-t(\alpha + \sum_{k,l \in N_{i,j}} a_{k-i,l-j})/C} + \frac{I_{ij}}{\alpha + \sum_{k,l \in N_{i,j}} a_{k-i,l-j}}. \tag{2.8}$$

From Eqn. (2.8), the output of the center cell is changed when a set of neighboring cells are interconnected by any template. If the template is changed, a vector $a_{k-i,l-j}$ is changed. The changed vector affects the conductance between the neighboring cells and the center

cell from Eqn. (2.8), and the output of center cell is stabilized when there is no change in the conductance which is changed by the template and the outputs of the neighboring cells.

The current from the feedback operator in the quantization function is negligible, so Eqn. (2.8) can be approximated as

$$v(t) = (v_0 - \frac{I_{ij}}{\alpha})e^{-t\alpha / C} + \frac{I_{ij}}{\alpha}. \quad (2.9)$$

From Eqn. (2.9), the difference between the initial voltage and the quantized voltage is proportional to the stabilization time. The stabilization time is maximal in the initial voltages where the red line meets the negative slope of the *I-V* curve as in Figure 2.2.

Table 2.1 demonstrates the stabilization times according to image processing functions. The stabilization time is measured by monitoring the settling time of the output of the processed image. Color extraction function shows longer stabilization time than the other functions. This is attributed to the change of the template between image processing functions. The changed template (α) in Eqn. (2.9) affects the output v(t) leading to the deviation of stabilization time.

Table 2.1: Comparison of stabilization times among image processing functions

	Worst	Best	Avg.
Quantization	120 ps	50 ps	83 ps
Smoothing	0.7 ns	0.05 ns	0.35 ns
Color extraction	1.3 ns	0.05 ns	0.7 ns

2.4.3. Power consumption analysis

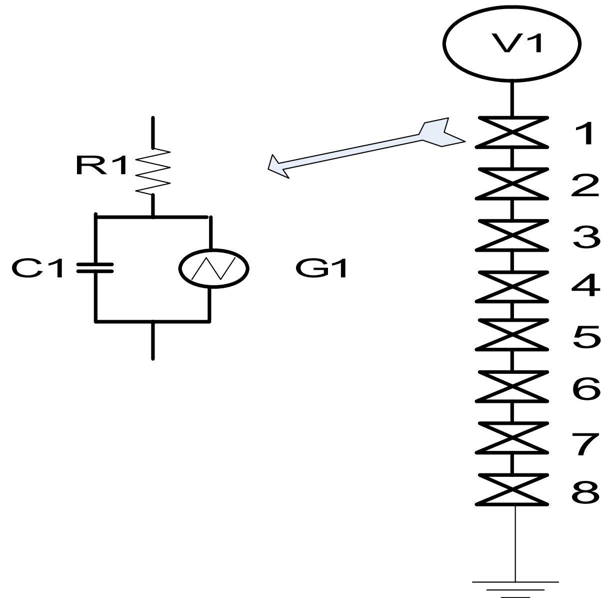


Figure 2.7: Equivalent circuit model of the multi-peak RTDs for color quantization.

Information on the power consumption of multi-peak RTDs is needed to determine whether the array of multi-peak RTDs is energy efficient. For quantization, the equivalent circuit model is shown in Figure 2.7, where we assume the current flows vertically from the metallic islands to the substrate. The IV characteristic of the G1 is defined by the zero-dimensional multi-peak resonant tunneling diode. To use the negative differential resistance characteristic of the device, we need to flow current which is between the peak value and the valley value in the IV curve of the device. We also assume the current between metallic islands for quantization is negligible. In Figure 2.7, each one-peak RTD can be modeled with a resistance serially connected to a capacitor and nonlinear voltage controlled current source with a parallel connection. The supplied input energy stored on the capacitor is expressed by the following equation

$$
\begin{aligned}
E_{v_0} &= \int_0^{\infty} i_0(t) \cdot v_0 \, dt \\
&= v_0 \int_0^{\infty} C_{total} \frac{dv_0}{dt} dt \\
&= C_{total} v_0 \int_0^{\infty} dv_0 \\
&= C_{total} v_0^2
\end{aligned}
\tag{2.10}
$$

where C_{total} is the sum of the serially-connected capacitances. Considering the external current source, the energy supplied to the metallic island equals

$$
\begin{aligned}
E_{total} &= E_{v_0} + E_{ext} \\
&= C_{total} v_0^2 + I^2 R_{total}
\end{aligned}
\tag{2.11}
$$

where R_{total} is the summation of the serially-connected resistances in the vertical direction. The energy of the capacitor after quantization is given as

$$
\begin{aligned}
E_{Voutput} &= \int_0^{\infty} i_{Vinput}(t) \cdot V_{output}(t) dt \\
&= \int_0^{\infty} C_{total} \frac{dV_{output}}{dt} \cdot V_{output}(t) dt \\
&= \frac{C_{total} V_{output}^2(t)}{2}
\end{aligned}
\tag{2.12}
$$

$$
E_{Voutput} = \frac{C_{total}}{2} \left[(v_0 - \frac{I}{\alpha}) e^{-t\alpha / C} + \frac{I}{\alpha} \right]^2 \tag{2.13}
$$

In Eqn. (2.12), the output voltage changes with time in Eqn. (2.9). From Eqn. (2.9), the energy change with time is expressed by Eqn. (2.13). Therefore, excluding input and output device, the energy consumption after quantization in the cell is given as

$$\begin{aligned} E_{diss} &= E_{total} - E_{Voutput} \\ &= C_{total} v_0^2 + I^2 R_{total} \\ &- \frac{C_{total}}{2} \left[(v_0 - \frac{I}{\alpha}) e^{-t\alpha / C} + \frac{I}{\alpha} \right]^2 \end{aligned} \tag{2.14}$$

2.5. Smoothing Function

2.5.1. Implementation and results

Figure 2.8 represents the changes in the pixel values due to the multi-peak RTD and resistance connected to each other. These color changes arise from the current change in each multi-peak RTD that is connected with resistors. In this simulation, we chose 50 MΩ for the connection resistors. As shown in the Figure 2.8, the colors in the neighboring positions change such that the color values between neighbor cells are similar. This phenomenon provides a way of detecting similarity of detected colors nearby. Color similarity can be detected when similar color pixels are merged to the same color in the long run.

Also, the traces of image color that change with time arise from the negative differential resistance which is one of the characteristics of the resonant tunneling diodes. Therefore, the simulation results reflect both the effects of resistances which connect the nearest neighbor cells and the negative differential resistance characteristic. In Figure 2.9, the change in the color images shows that the colors of the first row and the second row are quite different from those of the other rows, which are also shown in the simulation

results as the affinity of colors. In addition, the color value change by the neighboring cells can be applied to the noise removal of an image.

The feedback template and feedforward template for the smoothing function is represented as

$$a = \begin{pmatrix} 0 & l & 0 \\ l & -4l & l \\ 0 & l & 0 \end{pmatrix}, \quad b = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

where l is conductance between neighboring cells and the center cell. The typical value for l is between 0.04 and 0.02 μmho. Then, the output changes based on Eqn. (2.8).

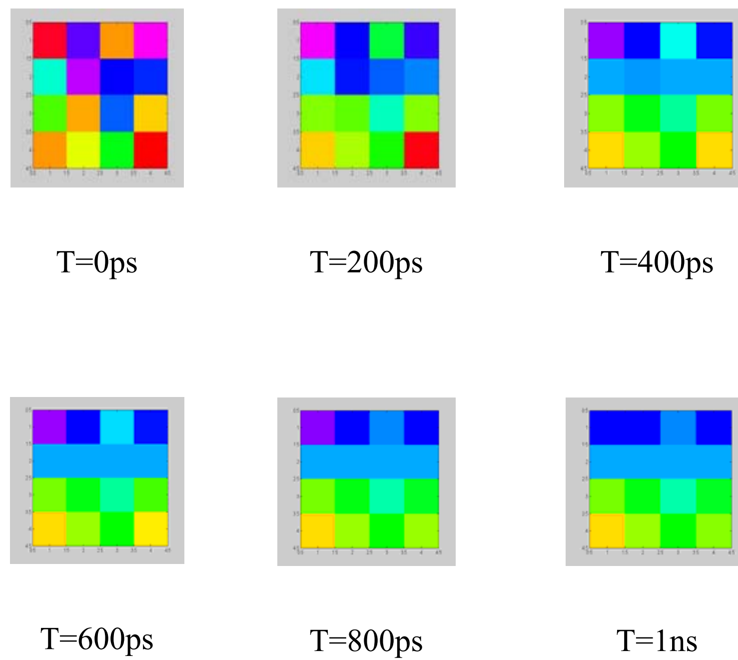
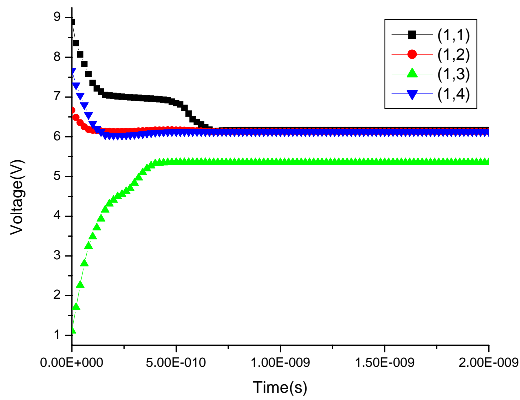
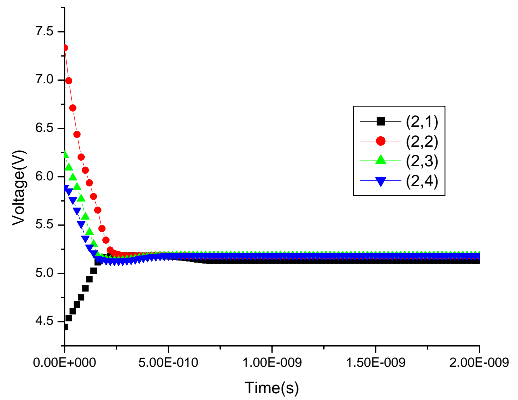


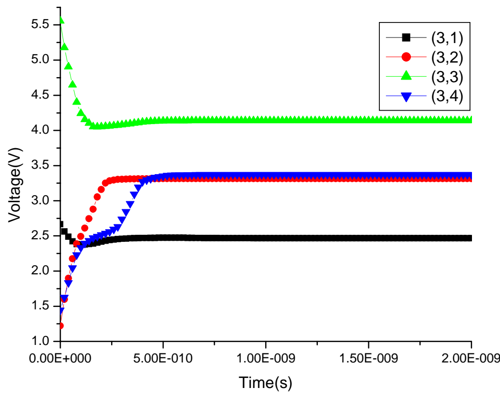
Figure 2.8: Transient representation of a 4 x 4 color image smoothing function from 0 ps to 200 ps.

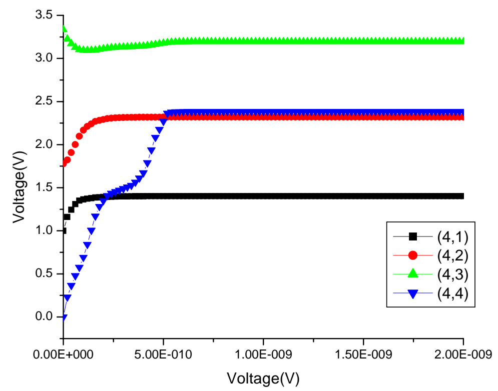


(a) First row



(b) second row



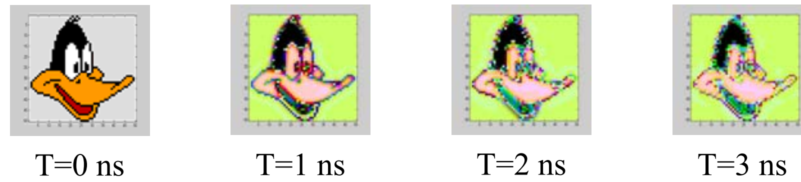


(c) Third row

(d) Fourth row

Figure 2.9: HSPICE results divided into rows with time from 0 to 2ns.

For a larger pixel-sized image, more realistic image processing is exhibited. As shown in the Figure 2.10, the smoothing effect is addressed by filling in the intermediate colors between quite different colors. This interpolation is implemented by controlling the resistance between the dots.



T=0 ns T=1 ns T=2 ns T=3 ns

Figure 2.10: 50 x 50 color image smoothing from 0ns to 3ns.

The algorithm of smoothing function which was implemented using an array of multi-peak RTDs can be represented using a pseudo code as shown below.

```
PROCEDURE smoothing(img: grid of color):
                    grid of color;
 * smoothing of the image *
VAR res: grid of color;
BEGIN
    res = img + MOVE.right(img) + MOVE.left(img); (*horizontal*)
    res = (res + MOVE.down(res) + MOVE.up(res)) DIV 5; (*vertical*)
RETURN res;
END smoothing
```

In this code, the input variable and output variable are the grids of the color. These grid variables represent the pixel values which include RGB information. The grid values are calculated with surrounding grid values. This function is the summation of the neighboring grids. After diving by 5, we can get the mean value of the grids including neighboring 4 grids and the center grid. The smoothing function can be realized with this algorithm.

However, the algorithm of smoothing image processing based on single conventional processor can be represented using a pseudo code as shown below. In this code, the input variable and output variable are scalars in color representation value. In this algorithm, we have two procedures. One is for the scalar smoothing, and the other one is for processing the whole array sequentially. These functions are represented as scalar_smoothing and smoothing function. Using FOR loops, the sequential processing is implemented with calling the scalar smoothing function in every loop.

```
PROCEDURE scalar_smoothing(img: grid of color):
                    grid of color;
 * quantize RGB value of the image *
VAR res: grid of color;
BEGIN
    res = img + MOVE.right(img) + MOVE.left(img); (*horizontal*)
    res = (res + MOVE.down(res) + MOVE.up(res)) DIV 5; (*vertical*)
RETURN res;
END scalar_smoothing
PROCEDURE smoothing(img: grid of color): grid of color;
* vectorize the grid of color *
BEGIN
    for(i =1; i<=4; i++)
    BEGIN
         for(i =1; i<=4; i++)
         BEGIN
              RETURN scalar_smoothing(img);
         END
     END
END smoothing
```

2.5.2. Settling time

As shown in Figure 2.9, the settling time of the smoothing function is longer than that of quantization function. This difference is attributed to the feedback operation in Eqns. (2.2) and (2.8). In Eqn. (2.8), the feedback operation term works as a function increasing the settling time. After the system is settled, the feedback operation term becomes zero. Hence, Eqn. (2.9) describes the smoothing function after settling.

2.5.3. Power consumption analysis

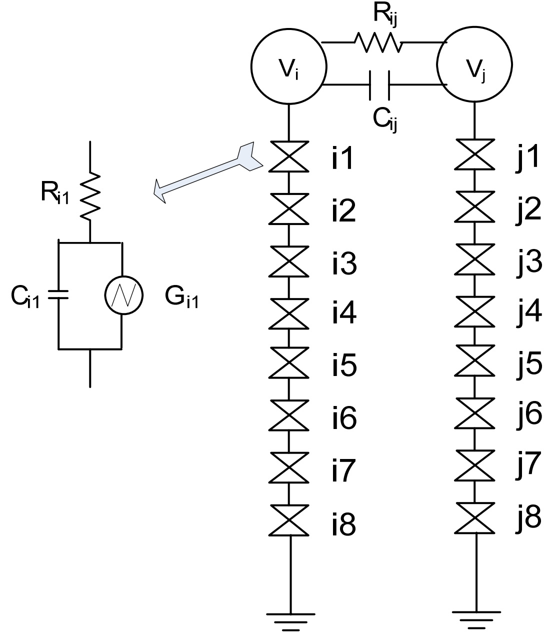


Figure 2.11: Equivalent circuit model of an array of multi-peak RTD structure for smoothing.

To analyze the smoothing function, we simplify the array of multi-peak RTD structure as two cells as shown in Figure 2.11. We then expand the two cells to an *N x N* array. If the voltages of the two cells are equal, the power consumption will be the same as that in the quantization function. When the voltages of the two cells are different and $V_{QDj} > V_{QDi}$, the supplied input energy is expressed by the following equation

$$\begin{aligned} E_{v_0} &= \int_0^{\infty} i_0(t)\cdot(v_{0i}+v_{oj})\,dt \\ &= v_{0i}\int_0^{\infty} C_t \frac{dv_{0i}}{dt}\,dt \\ &+ v_{0j}\int_0^{\infty} C_t \frac{dv_{0j}}{dt}\,dt \\ &= C_t v_{0i}\int_0^{\infty} dv_{0i} + C_t v_{0j}\int_0^{\infty} dv_{0j} \\ &= C_t(v_{0i}^2 + v_{0j}^2) \end{aligned} \tag{2.15}$$

where C_t is the summation of the capacitors surrounding cells. Including the energy from the external current source, the total energy supplied is given as

$$\begin{aligned} E_{total} &= E_{v_{0i}} + E_{v_{0j}} + 2E_{ext} \\ &= C_t v_{0i}^2 + C_t v_{0j}^2 + 2I_{ij}^2 R_{total} \end{aligned} \tag{2.16}$$

Considering current from QDi to QDj, the energy after smoothing function is expressed by the following equation

$$\begin{aligned} E_{Voutput} &= \int_0^\infty i_{QDi}(t)\cdot V_{QDi}(t)dt \\ &+ \int_0^\infty i_{QDj}(t)\cdot V_{QDj}(t)dt \\ &= \int_0^\infty C_t \frac{dV_{QDi}}{dt}\cdot V_{QDi}(t)dt \\ &+ \int_0^\infty C_t \frac{dV_{QDj}}{dt}\cdot V_{QDj}(t)dt \\ &= \frac{C_t V_{QDi}^2(t)}{2} + \frac{C_t V_{QDj}^2(t)}{2} \end{aligned} \tag{2.17}$$

$$\begin{aligned} V_{QDi}(t) &= \left[v_0 - \left(\frac{I_{ij}}{\alpha - 1/R_{ij}} \right) \right] e^{-t(\alpha - 1/R_{ij})/C} \\ &+ \frac{I_{ij}}{\alpha - \frac{1}{R_{ij}}} \end{aligned} \tag{2.18}$$

$$\begin{aligned} V_{QDj}(t) &= \left[v_0 - \left(\frac{I_{ij}}{\alpha + 1/R_{ij}} \right) \right] e^{-t(\alpha + 1/R_{ij})/C} \\ &+ \frac{I_{ij}}{\alpha + \frac{1}{R_{ij}}} \end{aligned} \tag{2.19}$$

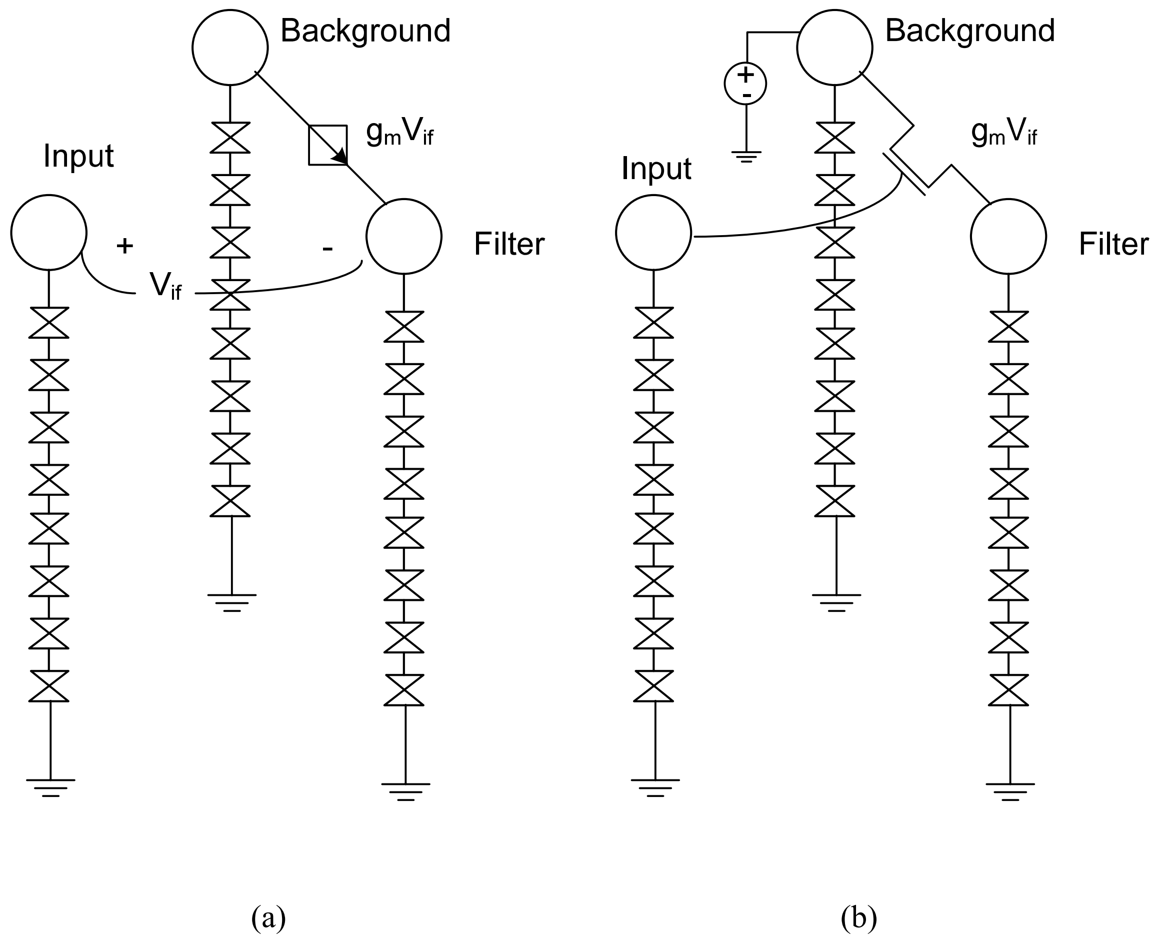
The total energy consumption used for smoothing function is given as

$$
\begin{aligned}
E_{diss} &= E_{total} - E_{Voutput} \\
&= C_t (v_{0i}^2 + v_{0j}^2) + 2 I_{ij}^2 R_{total} \\
&- \frac{C_t}{2} (V_{QD_i}^2(t) + V_{QDj}^2(t))
\end{aligned}
\quad (2.20)
$$

If we expand the two dots to an N x N array, the total energy consumption is modified as follows.

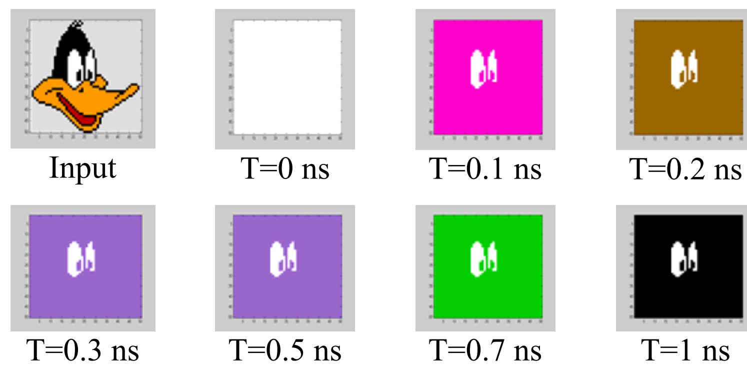
$$
\begin{aligned}
E_{diss} &= E_{total} - E_{Voutput} \\
&= C_t \sum_{n=1}^{N^2} v_{0n}^2 + 4N^2 \cdot I_{ij}^2 R_{total} - \frac{C_t}{2} \sum_{n=1}^{N^2} V_{QDn}^2(t)
\end{aligned}
\quad (2.21)
$$

2.6. Color Extraction

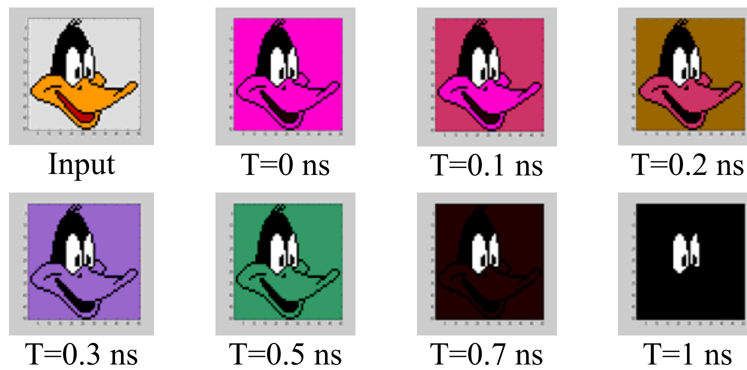


(a) (b)

Figure 2.12: Circuit configurations for color extraction. (a) using voltage controlled current source; (b) using MOSFET.



(a)



(b)

Figure 2.13: White color extraction; (a) Initialized with one color value in the filter dots; (b) Initialized with input image color value in the filter dots.

We need three cells each pixel output for color extraction function. Nine cells are required if we represent the color image with RGB method. The relationship between three cells is depicted as shown in Figure 2.12.

However, if we want to represent the relationship with template methods, the feedback and feedforward template can be represented as

$$a = \begin{pmatrix} 0 & m & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad b = \begin{pmatrix} 0 & 0 & 0 \\ n & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

The relationship between inputs to outputs is represented as n, and the relationship between background and output as m. From Figure 2.12, the feedback template a come from that the output of upper grid affect the center output and the feedback template b originates from the input of left grid affect the center output. From these templates, we expect the input image affects the conductance between the background output and the output image. The values of m, n are decided according to devices for implementation.

The circuits shown in Figure 2.12 are designed to extract color in the array of multi-peak RTD structure. Figure 2.12(a) is the circuit which uses a voltage controlled current source for color extraction. The input in Figure 2.12(a) is the RGB color value of the image to be processed. The background color after color image processing is determined by the background cell value. Also, the filter cell value is the color value of the image to be extracted.

In Figure 12(a), the change of the voltage in the filter cell can be obtained by

$$C \frac{dV_{filter}}{dt} = -J(V_{filter}) + g_m V_{if} \tag{2.22}$$

where g_m, V_{if}, and V_{filter} are conductance, voltage difference between a filter cell and an input cell, and voltage in the filter cell, respectively.

Two methods can be used to assign the initial state in Figure 2.12. One is to initialize the filter cells with one color value. The other method is to initialize the filter cells with the input image color value. These two methods can be used in the both of the circuits in Figure 2.12.

Figure 2.12(a) shows the circuit implementation with voltage controlled current source. In this circuit, the color extraction is possible using the principle that when V_{if} is greater than threshold voltage, the result is a current flow from the background cell to the filter cell. Figure 2.12(b) shows the circuit implementation with a MOSFET instead of the voltage controlled current source.

When the MOSFET is in the triode region, the change of the voltage in the filter cell can be obtained by

$$\begin{aligned} C\frac{dV_{filter}}{dt} &= -J(V_{filter}) \\ &+ \mu_n C_{ox}\frac{W}{L}[(V_{if} - V_{TH})V_{bf} - \frac{1}{2}V_{bf}^{2}] \end{aligned} \tag{2.23}$$

where $V_{if} - V_{bf} \leq V_{TH}$. In the saturation region, the change of the voltage in the filter cell is given by

$$\begin{aligned} C\frac{dV_{filter}}{dt} &= -J(V_{filter}) \\ &+ \frac{1}{2}\mu_n C_{ox}\frac{W}{L}(V_{if} - V_{TH})^2(1 + \lambda V_{DS}) \end{aligned} \tag{2.24}$$

where $V_{if} - V_{bf} > V_{TH}$ and λ is the channel-length modulation which is an empirical constant parameter.

In Figure 2.13, white color extraction from the two initialization techniques is shown with the simulation results from 0 ns to 1 ns. In Figure 2.13(a), the filter cells are initialized with white. The color which is not the same as the filter cells changes to black

which leads to the discernment of white from other colors. In Figure 2.13(b), the filter cells are initialized with the input colors. If the input color value is greater than the threshold voltage of the NMOS, the filter cells color changes to a black background color. In this simulation, we used 180 nm process libraries for the MOSFET device to realize the color extraction function.

The circuits in Figure 2.12(a), (b) are limited in terms of the number of colors that can be extracted because current controllers with the critical voltage allow only the extraction of the highest or lowest color value. Therefore, we suggest an advanced circuit which can extract any color in an image as shown in Figure 2.14. The color to be extracted is controlled by changing the four resistor values. The output voltage is given as follows.

$$
\begin{aligned}
V_{out} &= V_{in} \left(\begin{array}{l} V_{in} \cdot \dfrac{R_2}{R_1+R_2} > \dfrac{V_{DD}}{2} \; or \\ \dfrac{R_3}{R_3+R_4} V_{DD} + \dfrac{R_4}{R_3+R_4} V_{in} < \dfrac{V_{DD}}{2} \end{array} \right. \\
V_{out} &= 0 \left(\begin{array}{l} V_{in} \cdot \dfrac{R_2}{R_1+R_2} < \dfrac{V_{DD}}{2} \; or \\ \dfrac{R_3}{R_3+R_4} V_{DD} + \dfrac{R_4}{R_3+R_4} V_{in} > \dfrac{V_{DD}}{2} \end{array} \right.
\end{aligned} \tag{2.25}
$$

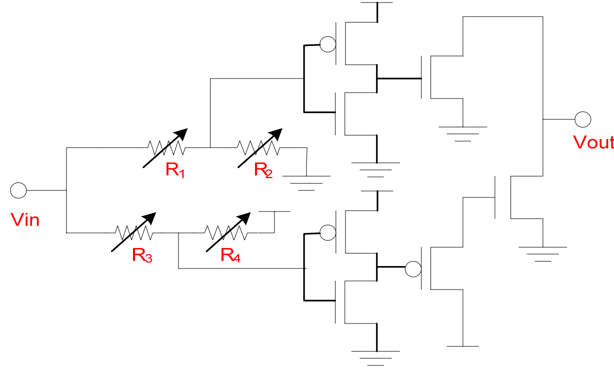


Figure 2.14: A circuit configuration for any color extraction.

Using the circuit in Figure 2.14, red color extraction of a 4x4 image is shown for time from 0 ns to 1 ns in Figure 2.15. The HSPICE simulation results for time are shown in

Figure 2.16. The HSPICE simulation result of the changes of the color values confirms the final output image in Figure 2.15.

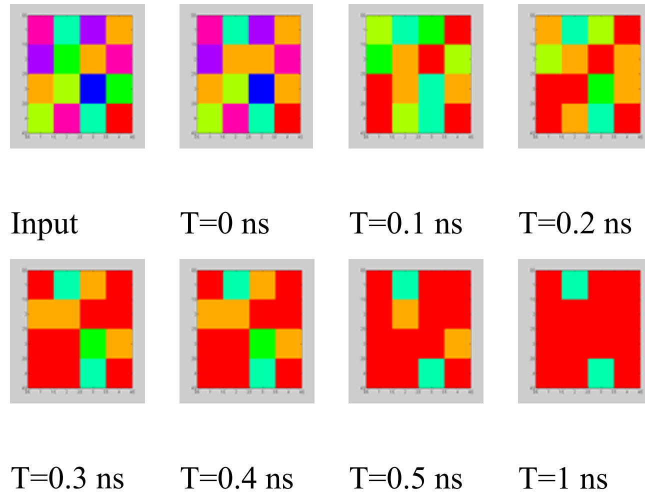


Figure 2.15: Red color extraction of 4x4 pixel image using the circuit shown in Figure 2.14.

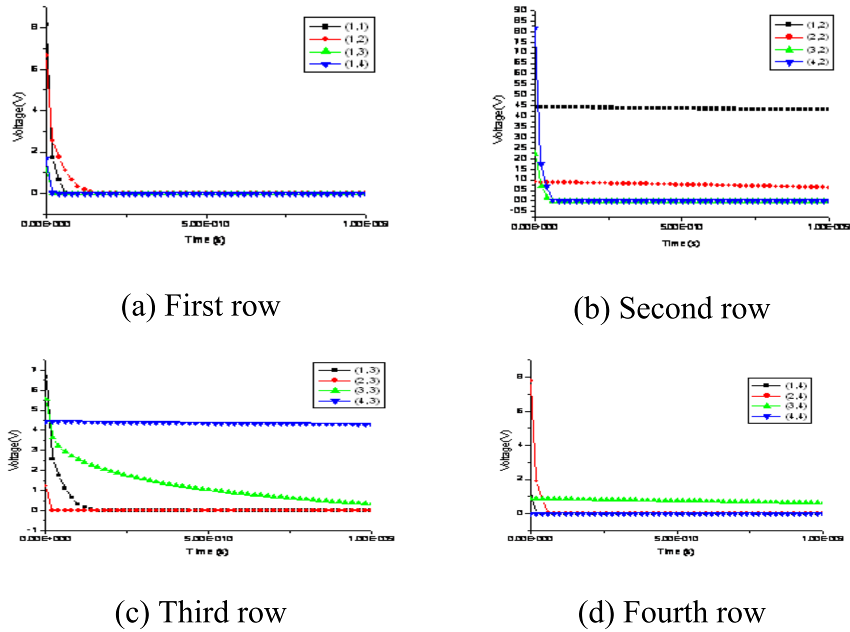


Figure 2.16: HSPICE simulation results divided into rows with time from 0ns to 1ns.

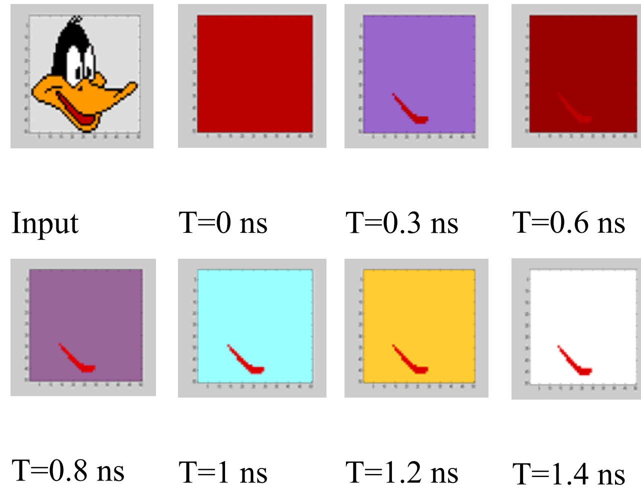
The algorithm of quantization function which was implemented using an array of multi-peak RTDs can be represented using a pseudo code as shown below.

```
PROCEDURE extraction(img: grid of color):
                    grid of color;
 * extraction of the image *
VAR res: grid of color; VAR ref_color, threshold: scalar color value;
BEGIN
    Difference = abs(res – ref_color);
    If(Difference <= threshold)
         BEGIN
             res = res;
         END
     ELSE BEGIN
          res = 0;
       END
RETURN res;
END extraction
```

In this code, the input variable and output variable are the grids of the color. These grid variables represent the pixel values which include RGB information. The grid values are compared with the reference color value which we want to extract. After comparing those two vales, if the difference is less than threshold value then, the value is reserved otherwise lowest value is assigned. The color extraction function can be realized with this algorithm.

Figure 2.17 represents the color extraction of various colors using the circuit shown in Figure 2.14. As shown in the Figure 2.17, the final images clearly represent the extracted color image.

Table 2.2 compares the performance and property of image functions. The image processor shows better performance in the functions of quantization and smoothing than color extraction. This originates from the devices for the function implementation. Since a conventional CMOS was used for the color extraction, the performance was dominated by the CMOS. However, this degradation can be overcome by using nanoelectronic devices such as SET and CNT. The nanoelectronic devices can reduce the area overhead and processing time. This will be the focus of our further study.



(a)

(b)

Figure 2.17: The color extractions of various colors with time from 0ns to 1.5ns; (a) red color extraction; (b) black color extraction.

Table 2.2: Comparison of Properties of Image Functions

Parameters	Quantization (50 x 50)	Smoothing (50 x50)	Color extraction (Figure 9(b)) (4x4)	Color extraction (Figure 12) (50 x50)
Delay (worst case)	0.13 ns	0.6 ns	0.3 ns (w/o interconnect delay)	1.3 ns (w/o interconnect delay)
Power (worst case)	10 nW	10 nW	0.038 mW	0.015 W
Power-Delay Product	$1.3\text{x}10^{-18}$ J	$6\text{x}10^{-18}$ J	$1.14\text{x}10^{-14}$ J	$1.95\text{x}10^{-11}$ J
Resistor	250 MΩ	50 MΩ	NA	NA
Area	~ 2.5 $(\mu m)^2$	~ 2.5$(\mu m)^2$	~2 $(\mu m)^2$	~30$(\mu m)^2$

2.7. Comparison with DSP Chips

A digital signal processor (DSP) is a type of microprocessor which is mainly used for communication for digital signal processing. A DSP demonstrates high speed and energy efficiency with optimized architecture for the digital signal processing [2.17].

Fast multipliers are used in a DSP for fast signal processing such as finite impulse response (FIR), Infinite impulse response (IIR), Fourier transformation and etc. Since multiplication is one of the most common functions in signal processing, a DSP requires fast multipliers for its fast operations. Modern DSP processors incorporate at least one dedicated, single-cycle multiplier or combined multiply-accumulate unit (MAC).

Multiple execution units are also incorporated in a DSP to perform high computational tasks. Arithmetic logic units and shifters and multiple MAC units are combined in parallel to generate high performance in signal processing.

Efficient memory access is also required for executing a MAC function in a single cycle. To execute in a single cycle, DSPs need to fetch a MAC instruction, a data sample and filter coefficients in a single cycle leading to two or more separate memory banks and high memory bandwidth.

Most DSPs use floating point formats for data, which require more complexity in hardware. However, floating point formats support simplified development in combination of hardware and software. DSP programmers need to scale values inside codes for fixed point formats. For fixed point formats, DSPs have 16 bit, 20 bit, 24 bit or 32 bit data words based on required accuracy.

Specialized instruction sets are used for DSPs to make maximum usage of hardware leading to increased IPC or to minimize memory space required to store DSP programs. However, DSP instructions become complicated by the specialized instruction sets. As a result, DSPs are not typically programmed in high level languages. Instead, program optimization is required for the specialized instruction sets.

Figure 2.18 represents a TMS320C55xx DSP which was developed by TI. The DSP has two instructions per cycle which are programmed for parallel execution by the assembly coder or compiler. Two MAC units are incorporated, and complex and compound instructions with mixed-width instructions from 8 to 48 bits [2.16].

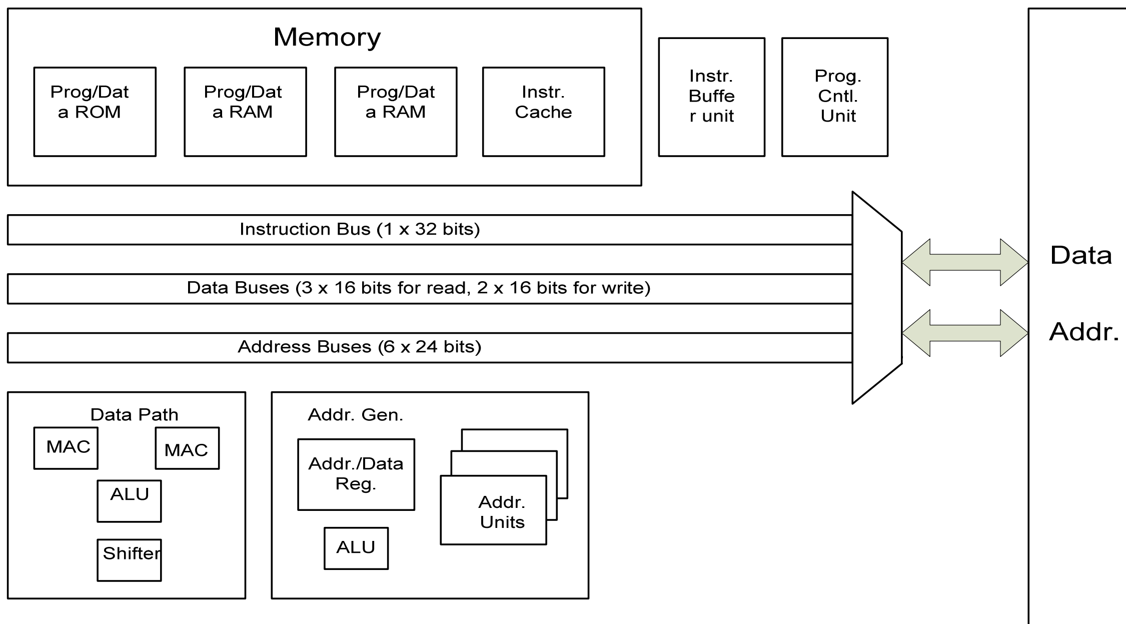


Figure 2.18: DSP block diagram of TI TMSC55XX [2.16].

Table 2.3 describes the performance comparison among commercial DSPs. The operating frequencies range from 150 to 600 MHz. In the relationship between IPC and power dissipation, power dissipation is reversely proportional to the IPC because the more processing instructions per clock cycles results in more power dissipation.

Table 2.4 demonstrates processing time and power dissipation comparison among different DSPs for smoothing function. Based on [2.17], the smoothing function requires around 20 instructions per pixel for averaging the neighboring pixel values and dividing by the number of pixels. The processing time is calculated considering CPI and frequency of DSP. The result shows that DSPs are slower than the suggested image processor by 2 to 4 orders of magnitude. However, since we did not include interconnect delay and I/O delay for the suggested image processor, the realistic speed of the suggested image processor will be degraded by several nano seconds. The DSPs dissipate more power than the suggested multi-peak RTD based processor by 4 to 6 orders of magnitude. Also, the area overhead of DSPs is bigger than the suggested processor by 5 orders of magnitude. Even if we consider the speed degradation by the interconnect delay, I/O delay and area overhead of I/O pad, the suggested image processor shows better performance than DSPs by 2 to 3 orders of magnitude.

Implementing parallel processing with the convention processor using multiple processors, we can reduce the processing time as much as our suggested processor. However, this will increase the power dissipation dramatically so that the implementation will become unrealistic. For instance, if we use 1000 processors for the parallel processing, the power dissipation will be around 1 KW. Therefore, this implementation will not be promising approach to reduce the processing time.

Table 2.3: Operational Specifications for different processors [2.11]-[2.16].

Parameters	DM 642	C641x	C6711	C55x
Frequency (MHz)	400 ~ 600	500 ~ 600	150	144 ~ 200
Power (W)	1 ~ 1.7	0.64 ~ 1.04	1.1	0.065 ~ 0.16
CPI	1.67 ~ 2.5	1.67 ~ 2	6.7	2.5 ~ 6.94

Table 2.4: Performance comparison among different processors for smoothing function (500 x 500 pixels).

Parameters	DM 642	C641x	C6711	C55x	Suggested processor
Smoothing (instructions)	20 x (1.67 ~ 2.5)	20 x (1.67 ~2)	20 x (6.7)	20 x (2.5 ~ 6.94)	1
Smoothing (ns)	55.6 ~ 125	66.8 ~ 80	893	250 ~ 964	0.6
Power (W)	1 ~ 1.7	0.64 ~ 1.04	1.1	0.065 ~ 0.16	1 x 10^{-6}
Power-delay product (nJ)	~ 212.5	~ 83.2	982.3	~ 154	6 x 10^{-7}
Area (μm^2)	~25 x 10^6	~25 x 10^6	~25 x 10^6	~25 x 10^6	25 x 10

2.8. Stability

To examine stability of the multi-peak RTD based color image processor which has nonlinear elements, Lyapunov theorem is applied. If we consider an M by N cells, the cell on the *i*th row and *j*th column is denoted by C(i,j). The output of the multi-peak RTD on the *i*th row and *j*th column is represented by V_{ij}. To apply the Lyapunov method, we define the Lyapunov function, $E(t)$, of the color image processor as given by Eqn. (2.26).

$$E(t) = -\frac{1}{2}\sum_{n,m}\sum_{k,l} A(n,m;k,l)\cdot F_{n,m}(t)\cdot F_{k,l}(t) + \frac{1}{R(V_{n,m})}\sum_{n,m}\int_0^{F_{n,m}} G^{-1}(F_{n,m})dF_{n,m}$$
$$-\sum_{n,m}\sum_{k,l} B(n,m;k,l)\cdot F_{n,m}(t)\cdot U_{k,l}(t) - \sum_{n,m} I_{n,m}F_{n,m}(t). \quad (2.26)$$

where $U_{k,l}(t)$, $F_{n,m}(t)$ *and* G is feedforward inputs and outputs from neighboring cells and relationship between the neighboring cell output and the target cell output.

1) Theorem 1: The Lyapunov function, $E(t)$, of the multi-peak RTD based color image processor is bounded by E_{max} when a supplied voltage source is bounded.

Proof:

If the supplied voltage source is bounded, $V_{n,m}$, $F_{k,l}$, $V_{k,l}$, and $U_{k,l}$ are bounded. If $V_{n,m}$ is bounded, the differential resistance which is the function of $V_{n,m}$ is also bounded. The current flow between the neighboring cells and the target cell which is defined as *A(i,j;k,l)* and *B(i,j;k,l)* is bounded as the supplied voltage source is bounded. Considering a bounded external current source, *E(t)* is bounded.

2) Theorem 2: The differential of the Lyapunov function, $E(t)$, of the multi-peak based color image processor is less than or equal to zero in the region where $dF_{n,m}/dV_{n,m} \geq 0$, that is

$$\frac{dE(t)}{dt} \leq 0 \qquad where \qquad \frac{dF_{n,m}}{dV_{n,m}} \geq 0. \tag{2.27}$$

Proof:

From Eqn. (2.26), the differential of $E(t)$ with respect to time t can be described by

$$\begin{aligned}
\frac{dE(t)}{dt} &= -\sum_{n,m}\sum_{k,l} A(n,m;k,l)\frac{dF_{n,m}}{dV_{n,m}}\cdot\frac{dV_{n.m}}{dt}\cdot F_{k,l}(t) \\
&+ \frac{1}{R(V_{n,m})}\sum_{n,m}\frac{dF_{n,m}}{dV_{n,m}}\cdot\frac{dV_{n,m}}{dt}\cdot G^{-1}(F_{n,m}) \\
&- \sum_{n,m}\sum_{k,l} B(n,m;k,l)\frac{dF_{n,m}}{dV_{n,m}}\cdot\frac{dV_{n.m}}{dt}\cdot U_{k,l}(t) \\
&- \sum_{n,m} I_{n,m}\cdot\frac{dF_{n,m}}{dV_{n,m}}\cdot\frac{dV_{n.m}}{dt} \\
&= -\sum_{n,m}\frac{dF_{n,m}}{dV_{n,m}}\cdot\frac{dV_{n.m}}{dt}\Big(\sum_{k,l} A(n,m;k,l)\cdot F_{k,l}(t) - \frac{G^{-1}(F_{n,m})}{R(V_{n,m})} \\
&+ \sum_{k,l} B(n,m;k,l)\cdot U_{k,l}(t) + I_{n,m}\Big) \\
&= -\sum_{n,m}\frac{dF_{n,m}}{dV_{n,m}}\cdot\left[\frac{dV_{n,m}}{dt}\right]^2\cdot C
\end{aligned} \tag{2.28}$$

Since we can assume C is positive in physical meaning, the polarity of $E(t)/dt$ depends on $dF_{n,m}/dV_{n,m}$. From this theorem, the multi-peak RTD based color image processor is stable in a limited region where $dF_{n,m}/dV_{n,m} \geq 0$. The differential of $F_{n,m}$ with respect to $V_{n,m}$ is positive when the differential resistance of multi-peak RTD is positive. When the functions of the color image processor are quantization or smoothing, the template A is changed and has all positive values. In the case of color extraction, the template A has positive values which are the variable with respect to the $F_{k,l}$ and template B also has positive values which are the variable with respect to the $U_{k,l}$. The changed template A and B do not affect the stability of the image processor. Based on Eqn. (2.28), the multi-peak RTD based color image processor is stable regardless of the functions of the image processor.

2.9. Conclusion

We suggest a new architecture to process color images using an array of multi-peak RTDs. We suggest various interconnect patterns for color image processing such as quantization, smoothing, and color extraction. We implemented quantization function and smoothing function through changing the conductance value between MPRTDs. In the case color extraction function, we suggested three different methods to extract the colors that we want. We demonstrate the HSPICE simulation results of those functions. Based on the simulation, we achieve 130 ps processing time for quantization, 600 ps for smoothing, and 1.3 ns for color extraction. Comparing the conventional image processor with the cutting edge technology, we obtain faster processing speed and lower power dissipation than the conventional processor.

CHAPTER III

Design of a Velocity-Tuned Filter Using a Matrix of Resonant Tunneling Diodes

3.1. Introduction

It is well known that real-time vision machine application tasks are computationally intensive and require complex and costly resources. In addition, certain specific tasks such as bio-robots and biomedical applications put additional constraints on the overall system in terms of its size, power consumption, shock resistance, and manufacturing cost. A real-time vision machine requires motion computation involving a large number of computations and many computational resources. An attractive solution to the problems is to use parallel image processing architectures [3.1][3.2].

Since velocity-tuned filters are one of the main parts of motion computation, they require a compact area and low power consumption. The velocity-tuned filter can be combined with pattern recognizers, optical flow sensors, and noise removers to create real-time vision machines. Even though velocity-tuned filters using spatiotemporal derivatives and Reichardt correlation detectors have been studied, they cannot provide sufficient area compactness, low power consumption and/or, speed [3.3][3.4]. These problems can be attributed to limitations of conventional analog circuits that cannot achieve high performance for the real-time vision machine.

Among nanoelectronic devices, an array of resonant tunneling diode structure shows good performance in terms of area, power consumption, and speed. Since being proposed by several researchers [3.1][3.2], their applications have been confined to Boolean logic or image processing.

Using the bi-stable states of the resonant tunneling diode, we can also apply it to amplify output signal difference in a system. In the case of velocity tuned filter, it is required to implement a circuit which magnifies the output difference to differentiate an object with a specific speed. Hence, we use the resonant tunneling diode to magnify the output signal difference based on speeds in the velocity tuned filter.

In this chapter, we propose a new type of velocity-tuned filter using nanoelectronic devices such as resonant tunneling diodes. The proposed velocity-tuned filter achieves performance in area, power, and speed superior to other conventional velocity-tuned filters by taking advantage of efficient parallel processing capability of an array of resonant tunneling diodes (RTDs).

The remainder of this chapter is organized as follows. Section 3.2 discusses the proposed an array of RTDs based velocity-tuned filters. In Section 3.2, the electrical characteristics of resonant tunneling diode and conventional velocity-tuned filters are addressed to help to comprehend the proposed velocity-tuned filters. Section 3.3 studies system analysis of the proposed velocity-tuned filters in speed, power, and stability. Finally, Section 3.4 presents our conclusions.

3.2. Array of RTDs Based Velocity Tuned Filters

3.2.1. Conventional velocity-tuned filter

Figure 3.1 shows a conventional velocity-tuned filter. The conventional velocity-tuned filter consists of pre-amplifiers, filters, multipliers and differential amplifiers. Pre-amplifiers consist of a pre-filtering part and an amplification part. In pre-filtering part, the input signal In_n comes from photo detectors which are not shown in Figure 3.1. After the

input signal is amplified by the pre-amplifiers, the input signal passes through the two layered filters, the receptor layer and the horizontal cell layer respectively. The main function of the receptor layer is to improve the signal to noise ratio. The second layer calculates a spatiotemporal average of the receptor output. The combined function of the two layered filter is a spatiotemporal band-pass filter. In [1.21][3.5], Torralba used CNN structure to implement the spatiotemporal band-pass filter and active and passive devices to connect the CNN cells.

After pre-filtering the signal from the optical flow, the filtered signal is amplified for providing inputs to two layered filters. The signal, then, is filtered with the two layered filters which allow the signals to have different values based on the velocities. The filtered signal is then magnified with local energy integration and the output is linearized with the difference between signals using shunting inhibition circuitry.

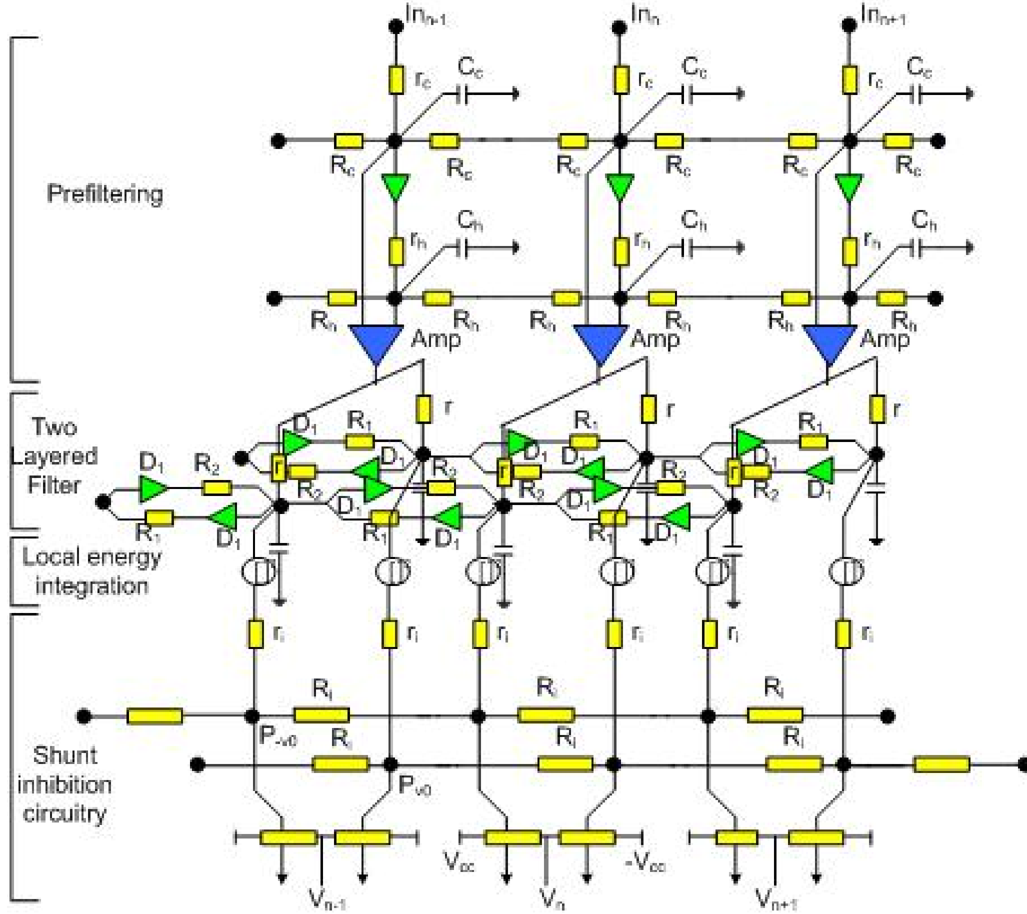


Figure 3.1: Conventional velocity-tuned filter that consists of pre-filtering, two layered filters, local energy integration, and shunting inhibition circuitry [1.21][3.5].

The local energy integration can be written in a mathematical form as

$$\begin{aligned} P(\Delta v) &= \int \Gamma_s(f_x) df_x \\ &= \int \Gamma_e(f_x) \left| G(f_x) \right|^2 df_x \\ &= \Gamma \int \left| G(f_x) \right|^2 df_x \\ &= \frac{\Gamma}{2\gamma \sqrt{1 + (v_x - v_0)^2 / \Delta v_0^2}} \end{aligned} \tag{3.1}$$

where v_x and $G(f_s)$ are input velocity and frequency response, and Γ, γ, and Δv_0 are constants [3.5]. The velocity-tuned filter needs the shunting inhibition circuitry to linearize the output. The function of the shunting inhibition is written mathematically as

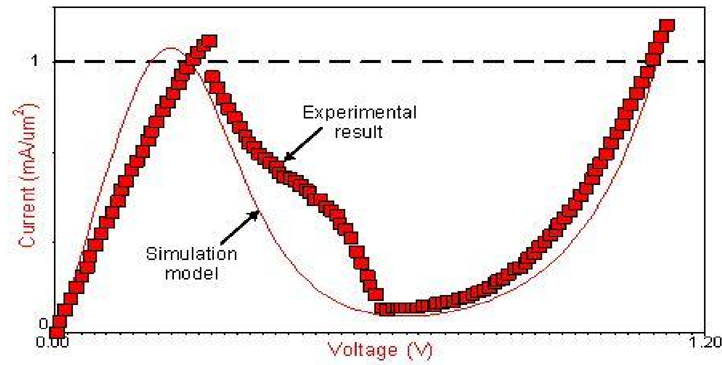
$$V_{out} = \frac{G_+ - G_-}{G_+ + G_-} V_{cc} = \frac{P_{V0} - P_{-V0}}{P_{V0} + P_{-V0}} V_{cc} \tag{3.2}$$

From (3.2), the output voltage is proportional to the difference between G_+ and G_-. The difference can be represented as the difference between P_{v0} and P_{-v0}. Therefore, an analog amplifier was implemented to magnify the difference in Torralba's work [1.21][3.5].

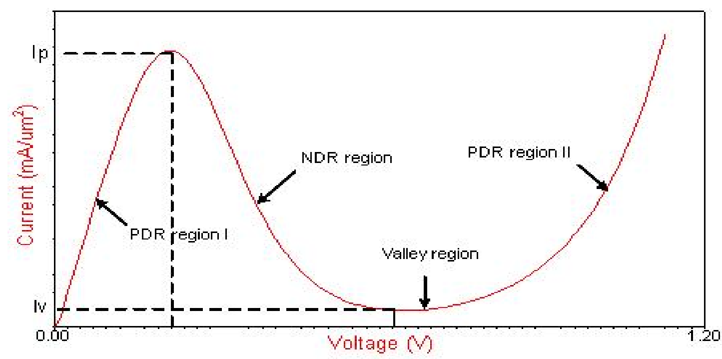
Since the conventional velocity-tuned filters use analogue circuitry to compute Eqn. (3.1) and (3.2), the conventional velocity-tuned filters are not well suited for real-time motion computation circuits for bio-robots or biomedical applications, which require high performance in area, power, and speed. Hence, the conventional velocity-tuned filters hardly meet the condition for real-time motion estimation for those applications.

To implement the real-time motion estimation, we employ an array of resonant tunneling diodes which are implemented by metallic islands on the resonant tunneling diodes. Since the incident light is pre-filtered and then amplified as an electric signal in the first part in Figure 3.1, we focus on the remaining part to improve the performance of the velocity tuned filter. The array of the resonant tunneling diodes is used to replace the filtering part, the local energy integration part, and the shunt inhibition part. Those functions can be implemented by using the negative resistance characteristic and area compactness of the array of the resonant tunneling diodes.

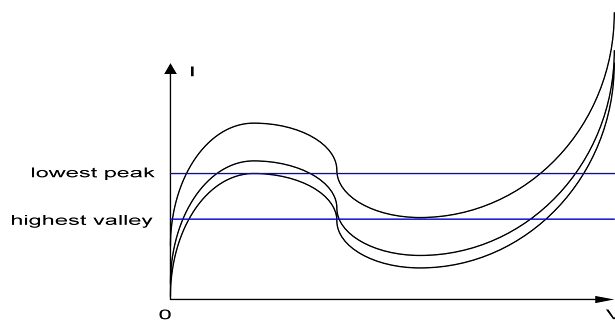
3.2.2. Resonant tunneling diode



(a) experimental result and simulation model



(b) simulation model of resonant tunneling diode with several regions



(c) *I-V* curve deviation with the size of metallic islands

Figure 3.2: Modeling of resonant tunneling diode based on experimental result.

Since the resonant tunneling diode (RTD) was introduced by Esaki *et al.* [3.6][3.7], it has been applied to various types of circuitry [3.8]. The main characteristic of the resonant tunneling diode is negative differential resistance (NDR). This characteristic originates from its heterostructure with a low-bandgap quantum well between high-bandgap materials. The thickness and width of the resonant tunneling diode are required to be fabricated in the order of several nanometers with epitaxial deposition techniques. The low-bandgap quantum well is quantized resulting in discretized energy levels in the quantum well. Figure 3.2(a) shows the modeling of the RTD *I-V* characteristic from experimental results. The fabricated RTD describes a peak-to-valley current ratio (PVCR) of 13 with a peak voltage (VP) of 0.28 *V* at room temperature. To model this *I-V* curve, factors that affect the conduction of RTD need to be examined.

However, the size of RTDs deviates with the process variation. The *I-V* curve also shifts up and down since the current flow proportional to the size of RTDs. This will affect the functionality of the system since the functions are valid when the NDR region is guaranteed. This means that if the lowest peak is higher than highest valley, the functionality is valid from the deviation of the process. From the experimental result, we have a PVCR of 13. Therefore, functionality is still valid as long as the size deviation of RTDs is less than 13.

However, it is possible to make a RTD having a PVCR of 46 with cutting edge technology [3.15]. In this case, the maximum cell size deviation from the process to guarantee the functionality of the system is 46. The situation will become even better as the process technology is developed.

Conduction of the resonant tunneling diode consists of two parts. One is conduction due to resonant tunneling and the other is conduction from diode conduction. The negative differential resistance results from the resonant tunneling conduction effect [3.9][3.10][3.11]. Using a physics-based model suggested by Schulman *et al*., the resonant tunneling current of RTD is modeled as a summation of resonant tunneling effect, J_1, and diode conduction effect, J_2:

$$J(V) = J_1(V) + J_2(V), \tag{3.3}$$

$$J_1(V) = \frac{qm \cdot kT\Gamma}{4\pi^2\hbar^3} \ln\left(\frac{1 + e^{\frac{E_F - E_r + \frac{n_1 qV}{2}}{kT}}}{1 + e^{\frac{E_F - E_r - \frac{n_1 qV}{2}}{kT}}} \right) \cdot \left(\frac{\pi}{2} + \arctan\left(\frac{E_r - \frac{n_1 qV}{2}}{\frac{\Gamma}{2}} \right) \right), \tag{3.4}$$

$$J_2(V) = H\left(e^{\frac{n_2 qV}{kT}} - 1\right) \tag{3.5}$$

where E_F is the Fermi energy, E_r is the energy of the resonant level, Γ is the resonant width, and n_1 and n_2 are model parameters. Those parameters are often obtained empirically and affect the slope of the curve in Figure 3.2(b).

The resonant tunneling occurs when the applied voltage across the diode is aligned to one of the quantized energy levels in the quantum well as shown in PDR region *I* in Figure 3.2(b). However, when the applied voltage is increased to be misaligned to the quantized energy level, the conduction is decreased as shown in the NDR region in Figure 3.2(b). The current subsequently increases as conduction through higher energy states becomes possible as shown in the PDR region in Figure 3.2(b). This characteristic enables the circuit switch fast and self-latching or bi-stable. Using this characteristic, a wide class of circuit applications, for instance high speed circuits, low power-delay product circuits, and multi-valued logic, can be implemented.

3.2.3. Proposed velocity-tuned filter

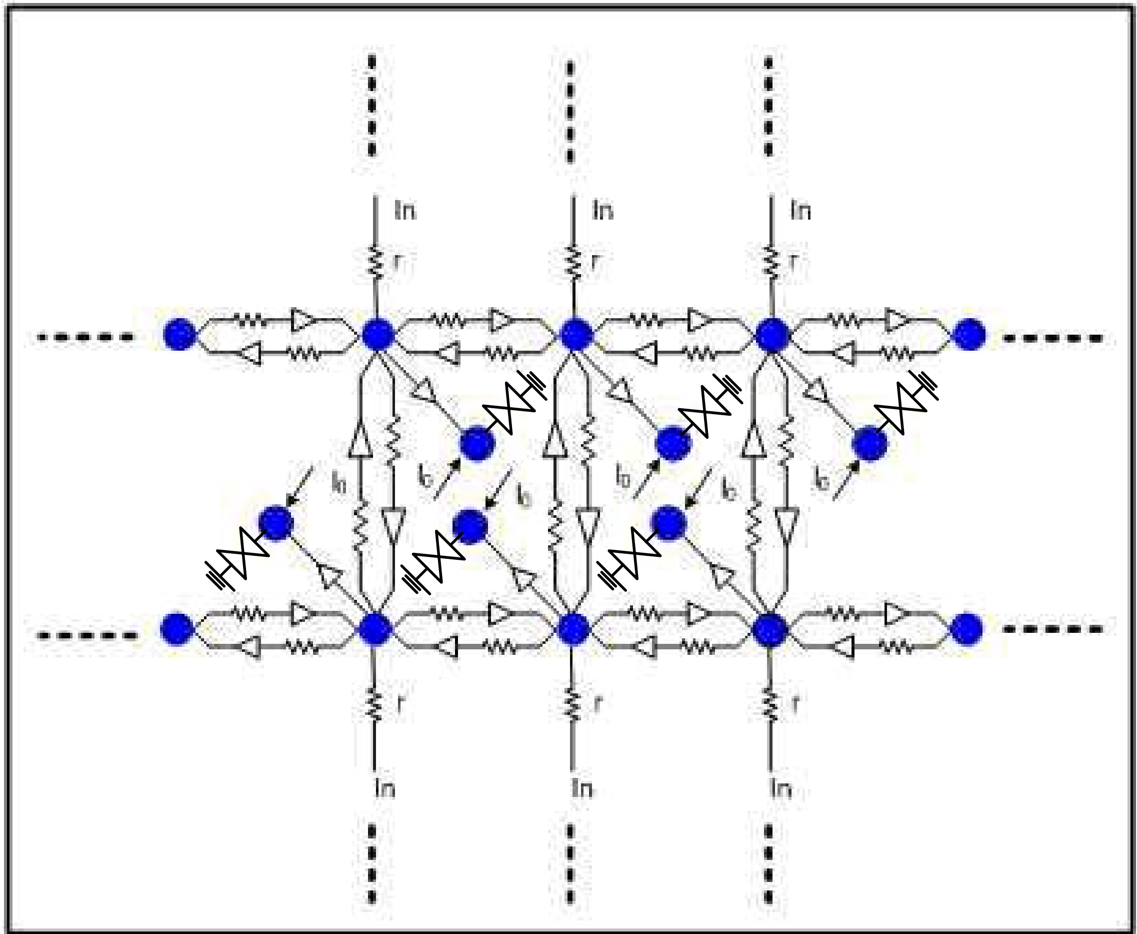


Figure 3.3: Proposed velocity-tuned filter with an array of filters and RTDs.

The proposed velocity-tuned filter consists of an array of filters and RTDs. The RTDs are connected to filters with diodes. Each filter cell is connected to four neighboring cells and

one output cell. The output cell consists of RTDs, and needs a static current source which is vertically connected to each RTD. Conventional velocity-tuned filters require local energy integration and shunting inhibition circuits to magnify the output difference. Our filter structure does not employ these circuits and instead, uses the RTD to magnify the filtered output differences. Through the differential resistance characteristic of the RTD, the final value is determined by the RTD's *I-V* characteristic and the external current source. In the proposed velocity-tuned filter, the output of the filter makes the RTD device operate when the output of the filter is higher than 1.4 *V*. This threshold voltage can be given by

$$V_{th} = V_d + \frac{V_p + V_v}{2} \tag{3.6}$$

where V_d is the diode junction voltage and V_p and V_v are the peak voltage and the valley voltage respectively in RTD *I-V* curve.

Figure 3.3 shows the schematic diagrams of the proposed velocity-tuned filter's side view and top view. Analysis of the proposed velocity-tuned filter requires a new state equation. This new state equation of the proposed structure is obtained by modifying the state equation of diffusion circuit in Eqn. (3.7). In Eqn. (3.7), $X_{n,m}$, $S_{n,m}$, and v represent input voltage, output voltage, and velocity respectively. Also, γ and τ are defined in Eqn. (3.8) and Eqn. (3.9). Here, γ is related with conductance between cells in the filtering part and τ affects the processing time of the filter. Also, v_{x0} and v_{y0} represent tuning velocity in the x direction and the y direction, respectively:

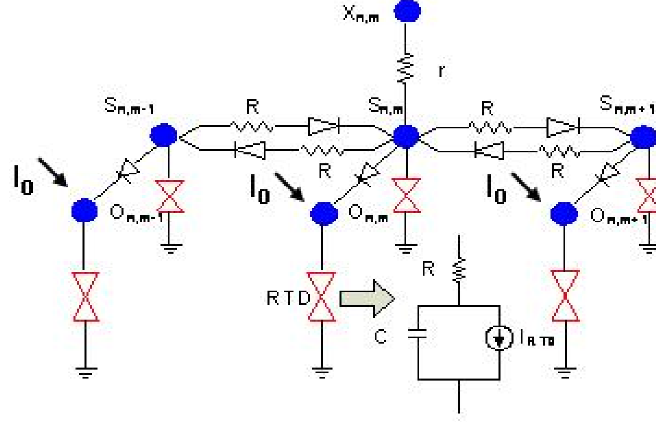


Figure 3.4: Circuit model of a basic cell of the proposed velocity-tuned filter.

$$
\begin{aligned}
\tau \frac{dS_{n,m}(t)}{dt} &= X_{n,m}(t) - S_{n,m}(t) + (\gamma^2 + \frac{v_{x0}\tau}{2})(S_{n-1,m}(t) - S_{n,m}(t)) \\
&+ (\gamma^2 + \frac{v_{x0}\tau}{2})(S_{n,m-1}(t) - S_{n,m}(t)) \\
&+ (\gamma^2 - \frac{v_{x0}\tau}{2})(S_{n+1,m}(t) - S_{n,m}(t)) \\
&+ (\gamma^2 - \frac{v_{x0}\tau}{2})(S_{n,m+1}(t) - S_{n,m}(t))
\end{aligned} \tag{3.7}
$$

$$
\gamma^2 = \frac{r}{\frac{kT}{qI_s(e^{qV_d/kT} - 1)} + R}, \tag{3.8}
$$

$$
\tau = r \cdot C. \tag{3.9}
$$

Considering the resonant tunneling diode, the state equation can be obtained as follows:

$$\frac{d}{dt}q_{n,m} = C_{n,m}\frac{dS_{n,m}}{dt} = -f(S_{n,m}(t)) + h(S_{n,m}), \tag{3.10}$$

$$f(S_{n,m}(t)) = F\cdot(\exp(n_2 S_{n,m} q/kT) - 1) \tag{3.11}$$

$$+A\cdot\ln\left[\frac{1+e^{((B-C+n_1S_{n,m})q/kT)}}{1+e^{((B-C-n_1S_{n,m})q/kT)}}\right]\cdot\left[\frac{\pi}{2}+\arctan\left(\frac{C-n_1S_{n,m}}{D}\right)\right]+H(e^{\frac{n2eV}{kT}}-1),$$

$$h(S_{n,m}(t)) = \sum_{Cell(k,l)\in N_r(n,m)} G(Nr(n,m),\vec{v}(t,x,y))\cdot(e^{q(N_r(n,m)-I_rR-S_{n,m})/kT}-1)\cdot I_s \tag{3.12}$$

where

$$G(N_r(n,m),\overrightarrow{v(t,x,y)}) = \left\{\frac{\gamma^2+\frac{\overrightarrow{v(t,x,y)}\cdot\tau}{2} \quad when \quad \cos\theta(\overrightarrow{v(t,x,y)},\overrightarrow{N_r(n,m)S_{n,m}})=0;}{\gamma^2-\frac{\overrightarrow{v(t,x,y)}\cdot\tau}{2} \quad when \quad \cos\theta(\overrightarrow{v(t,x,y)},\overrightarrow{N_r(n,m)S_{n,m}})\neq 0;}\right\} \tag{3.13}$$

and $N_r(n,m)$ is a r-neighborhood of a cell *Cell(n,m)* and is defined by

$$N_r(n,m) = \left\{\begin{matrix} Cell(k,l) \mid \max\{|k-n|,|l-m|\}\leq r \\ 1\leq k\leq N; 1\leq l\leq M \end{matrix}\right\}, \tag{3.14}$$

where $I_{n,m}$ is the bias current and $f(S_{n,m})$ describes the *I-V* characteristic of the resonant tunneling diode with physical model parameters, A, B, C, D, F, n_1, and n_2; $h(S_{n,m})$ represents the current from neighboring cells to $S_{n,m}$ where I_s is the saturation current in the diode.

The relationship between $S_{n,m}$ and $O_{n,m}$ can be described by combining Eqn. (3.10) and Eqn. (3.15):

$$g(O_{n,m}(t)) = (e^{q(S_{n,m}-O_{n,m})/kT} - 1) \cdot I_s. \quad (3.15)$$

Using these equations, we obtain a new state equation as given by

$$C_{n,m}\frac{dO_{n,m}}{dt} = -f(O_{n,m}(t)) + g(O_{n,m}(t)) + I_{n,m}(t), \quad (3.16)$$

which represents the velocity-tuned filter. The schematic diagram of the velocity-tuned filter for a basic cell is described in Figure 3.4. Figure 3.5 and Figure 3.6 describe experimental results of the proposed velocity-tuned filter. In Figure 3.5 and Figure 3.6, the dark round shaped dots represent input impulse signals. Observing signals based on the time sequence, the moving object and the standing object are divided with time, T. Hence, $T = 2$ represents the next time to $T = 1$ in the sampling clock sequence. In Figure 3.5, the standing object (S_4) is amplified in the output, O_4, while other moving objects from S_2 to S_1 hold low output values in O_1. In Figure 3.6, the moving object from S_2 to S_1 in the left direction is latched to the high value in O_1, while the standing object, S_4 is latched to the low output value in O_4. In Figure 3.7, the left figure is the moving input. The first row represents an object moving with a velocity of 2 pixels/sec to the right. The second row shows an object with a velocity of 1 pixel/second to the right. The third row represents a non-moving object. The fourth row shows an object with a velocity of 1 pixel/second to the left. Finally, the fifth row represents an object with a velocity of 2 pixels/second to the left. The right sides of Figure 3.7 and Figure 3.8 show the output of the filter. In Figure 3.7, only the target object, which does not move has a high output value shown with colors, red and yellow. Therefore, the non-moving object can be detected as shown in Figure 3.7(b). In Figure 3.8, the moving object with a velocity of 1 pixel/second to the left has a high output value. Hence, the object with a velocity of 1 pixel/second to the left can be detected as shown in Figure 3.8(b). The velocity to be filtered can be chosen by controlling the resistance values between the cells. Based on Eqn. (3.7), we control the resistance value to filter objects with a certain velocity. The output results in Figure 3.8(b) can be explained using Eqn. (3.16).

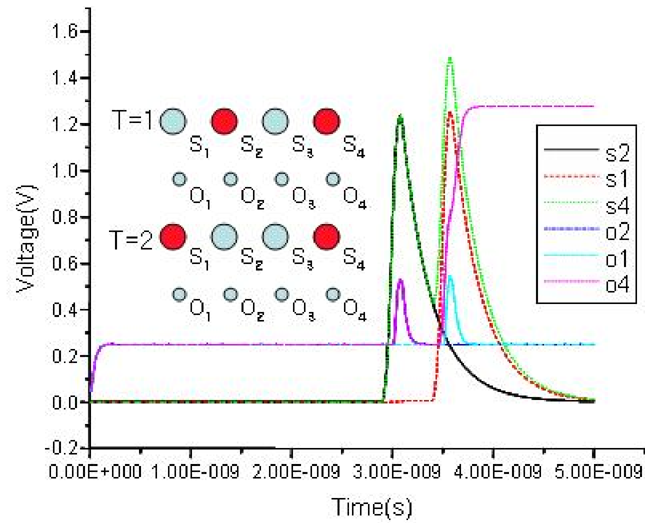


Figure 3.5: HSPICE simulation result for velocity-tuned filter with velocity 0.

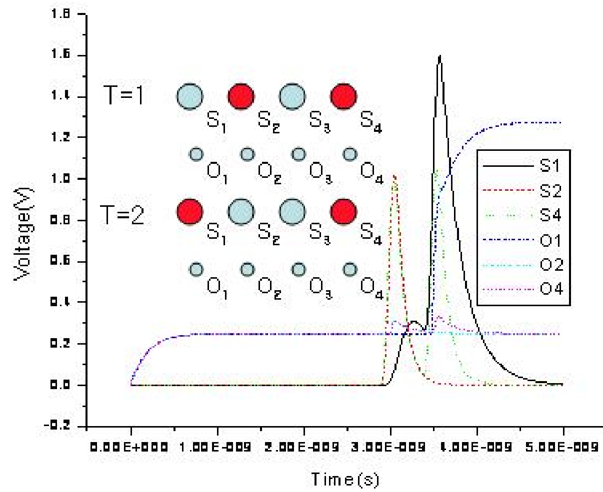


Figure 3.6: HSPICE simulation result for velocity-tuned filter with velocity 1.

3.3. System Analysis

3.3.1. Delay analysis of velocity-tuned filter

In this section, we investigate the delay of the input signal to the output node. To analyze the signal delay, we regard our velocity-tuned filter as an RC network in the form of a tree. To examine the system as an RC tree, we simplify our velocity-tuned filter as shown in Fig. 9. In Fig. 9, we represent the velocity-tuned filter only with capacitors and resistors. Hence the following Eqn. (3.17) always holds for the output node $O_{n,m}$:

$$\int_0^{\infty} V_{O_{n,n}}(t)dt = \sum_{k=1}^{N} C_k S_k(0) R_{O_{n,m},k}, \tag{3.17}$$

where

$$R_{O_{n.m},k} = \sum R_j \Rightarrow (R_j \in [path(O_{n,m} \to X_{n,m}) \cap path(k \to X_{n,m})]. \tag{3.18}$$

If we assume that the array of filter nodes is a 2*n* x 2*m* matrix, same as that of the output nodes, the delay of the signal from the input node to the output node in the position of *n, m* can be represented by

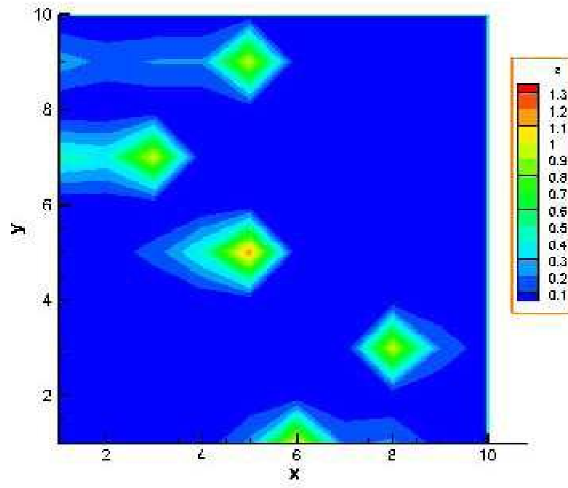
$$r\sum_{l=1}^{2m}\sum_{k=1}^{2m}(C_{S\,k,l} + C_{O_{k,l}}) - rC_{O_{n,m}} + (r + R_d(V) + 5R_{rtd}(V))C_{O_{n,m}}, \tag{3.19}$$

where $R_{rtd}(V)$ and $R_d(V)$ are the resistances with functions of applied voltages across each device. Since we assume that the output is stabilized after 5τ (time constant) due to the RTD, we add $5R_{rtd}(V)$ for the RTD stabilization time on the output. Based on the simulation results in Fig. 5 and Fig. 6, the RTD stabilization takes time from 200 ps to 500 ps in our simulation. Assuming the capacitances are same for all nodes, Eqn. (3.19) can be approximated as

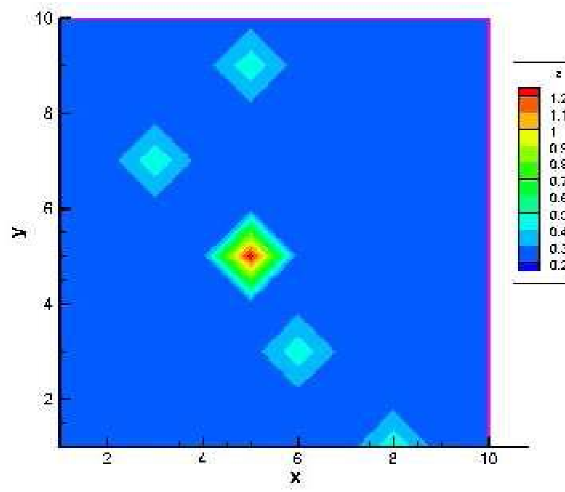
$$C(4rn \cdot 4rm + R_d(V) + 5R_{rtd}(V)). \qquad (3.20)$$

Given the size of the array, the delay is governed by the input resistance, the diode resistance, and the RTD resistance from Eqn. (3.20).

However, since the conventional velocity-tuned filter uses local energy integration and shunt inhibition circuitry after filtering, our velocity-tuned filter is much faster than the conventional velocity-tuned filter. To examine this in detail, the propagation delay of local energy integration and shunt inhibition circuitry needs to be investigated. In circuit implementation, the integration logic consists of multipliers and adders. Using cutting-edge technology, 0.11 *um* CMOS standard cell library, the whole processing of those combinational logics takes 4.5 *ns* to 5 *ns* for 4 bits (1 digit) multipliers [3.12]. Including adders, the processing time will be increased until 7 *ns* to 8 *ns*, which is around 100 times slower than that of our nanoelectronic circuitry as shown in Fig. 5 and Fig. 6.

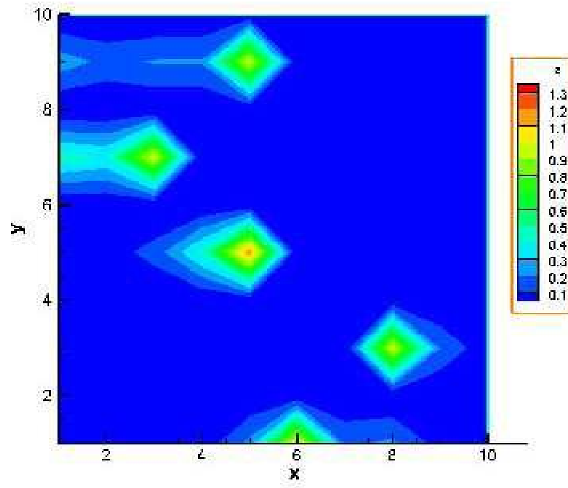


(a) input moving objects

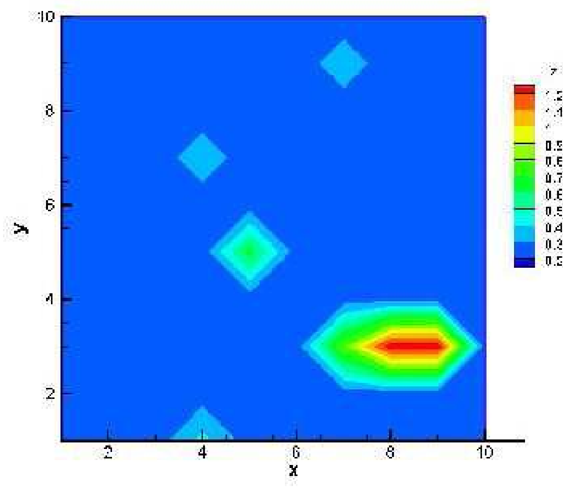


(b) output filtered objects

Figure 3.7: Experimental result of velocity-tuned filter for a velocity of 0 pixel/second.



(a) input moving objects



(b) output filtered objects

Figure 3.8: Experimental result of velocity-tuned filter for a velocity of 1 pixel/second.

3.3.2. Power consumption analysis of velocity-tuned filter

Since our proposed velocity-tuned filter has fewer device components than conventional velocity-tuned filters, the proposed velocity-tuned filter is assumed to consume less power. To analyze the power consumption, we need to partition the velocity-tuned filter into two components. The first part corresponds to the filter part, which includes filtering connections, input connections, metallic islands, and the resonant tunneling diode in Fig. 4 and the second part consists of output interconnections, output metallic islands, and the output resonant tunneling diode in Fig. 4. To calculate the power dissipation, we redraw Fig. 4 as shown in Fig. 10 which excludes the right hand side of the filters for circuit analysis, assuming that the current flow in the filter area is unidirectional. If we assume the input voltage to node $X_{n,m}$ in Fig. 9 is V_{supply} which is a pulse signal with a rising and a falling time, the average power consumption of resistor r is given as

$$P_r = r \cdot I^2 = r \cdot (YV_{\sup ply})^2 = r \cdot \left|\frac{1}{Z}\right|^2 \cdot \left|V_{\sup ply}\right|^2, \tag{3.21}$$

$$Y = \frac{1}{Z} \tag{3.22}$$

where

$$Z = r + \frac{(R + R_d(V) + \frac{A \cdot (R_d(V) + A)}{2A + R_d(V)} \cdot \frac{A \cdot (R_d(V) + A)}{2A + R_d(V)}}{R + R_d(V) + \frac{2A \cdot (R_d(V) + A)}{2A + R_d(V)}} \tag{3.23}$$

where

$$A = \frac{\frac{1}{j\omega C} \cdot R_{RTD}(V)}{\frac{1}{j\omega C} + R_{RTD}(V)}, \tag{3.24}$$

where $R_{RTD}(V)$ and $R_d(V)$ are functions of voltage, V, applied across the devices. $R_{RTD}(V)$ and $R_d(V)$ can be obtained from *I-V* curves when voltages are applied across the devices. The *I-V* curve of the diode is described by

$$I = I_s(e^{\frac{qV}{kT}} - 1). \tag{3.25}$$

Also, the *I-V* curve of RTDs is described by Eqn. (3.3)-(3.5). Since V_{supply} is not a DC source, the voltages across the devices change with time. This calculation can be processed by partitioning the simulation time and applying the equations with each interval repeatedly. The average power consumption of resistor R is given as

$$P_R = R \cdot I_1^2 = R \cdot \left| \frac{\frac{A \cdot (R_d(V) + A)}{2A + R_d(V)}}{R + R_d(V) + \frac{2A \cdot (R_d(V) + A)}{2A + R_d(V)}} \right|^2 \cdot \left| \frac{1}{Z} \right|^2 \cdot |V|^2. \tag{3.26}$$

In the case of a diode in the filter parts, the average power consumption is written as

$$P_{R_d(V)} = R_d(V) \cdot \left| \frac{\frac{A \cdot (R_d(V) + A)}{2A + R_d(V)}}{R + R_d(V) + \frac{2A \cdot (R_d(V) + A)}{2A + R_d(V)}} \right|^2 \cdot \left| \frac{1}{Z} \right|^2 \cdot |V|^2. \tag{3.27}$$

The output which includes the diode, metallic islands, and resonant tunneling diode consumes power described by

$$P_{output} = \frac{\left|\frac{A\cdot(R_d(V)+A)}{2A+R_d(V)}\right|^3}{\left|R+R_d(V)+\frac{2A\cdot(R_d(V)+A)}{2A+R_d(V)}\right|^2}\cdot\left|\frac{1}{Z}\right|^2\cdot|V|^2. \quad (3.28)$$

The total power dissipation can be obtained by measuring current from the power source and multiplying with the supplied voltage. We could obtain the total power dissipation by using HSPICE simulation which includes the above calculations. However, since the output part include the external current into the output cell, the power consumption due to the external current needs to be added to the output power consumption. The modified power consumption including the contribution from the external current is given as

$$P_{output-\mathrm{mod}} = P_{output} + I_{ext}^2 \cdot R_{RTD}(V). \quad (3.29)$$

In detail, the external current can be minimized to minimize the static power consumption by satisfying the following condition,

$$I_v + I_{leak} < I_{ext} < I_{peak} + I_{leak}, \quad (3.30)$$

where the I_{leak}, I_{peak} and Iv are the leakage current from the capacitance components of the RTDs, the peak current and the valley current, respectively as shown in Fig. 2(b). If we apply 0.5 mA/um^2 of the current density, we have two stable voltage points, 0.15 V and 1.1 V. Therefore, the corresponding power dissipations from the external current source are 15 nW and 110 nW. If we apply the experimental values of the resonant tunneling diodes' capacitance of 10^{-15} F and V_{supply} of 1.5 V to the HSPICE simulation, we get the total average power dissipation in the range from 29 nW to 114 nW depending on output voltages. Hence, if we have high output voltages, the external current is dominant in the total power dissipation; otherwise, it is comparable with dynamic power dissipation. However, the dynamic power dissipation is dominated by the power from the input resistance r and the resistances, R in the filter, each of which has much higher value than the other resistive components.

Based on the above calculations, Table I gives the performance comparison with the velocity-tuned filters in [3.5][3.13]. In terms of area, our VTF is 10 to 100 times smaller than the other VTFs. As shown in Table I, the processing time of our VTF is 3 to 6 orders of magnitude faster than the other VTFs and consumes around 100 times less power than the other VTFs.

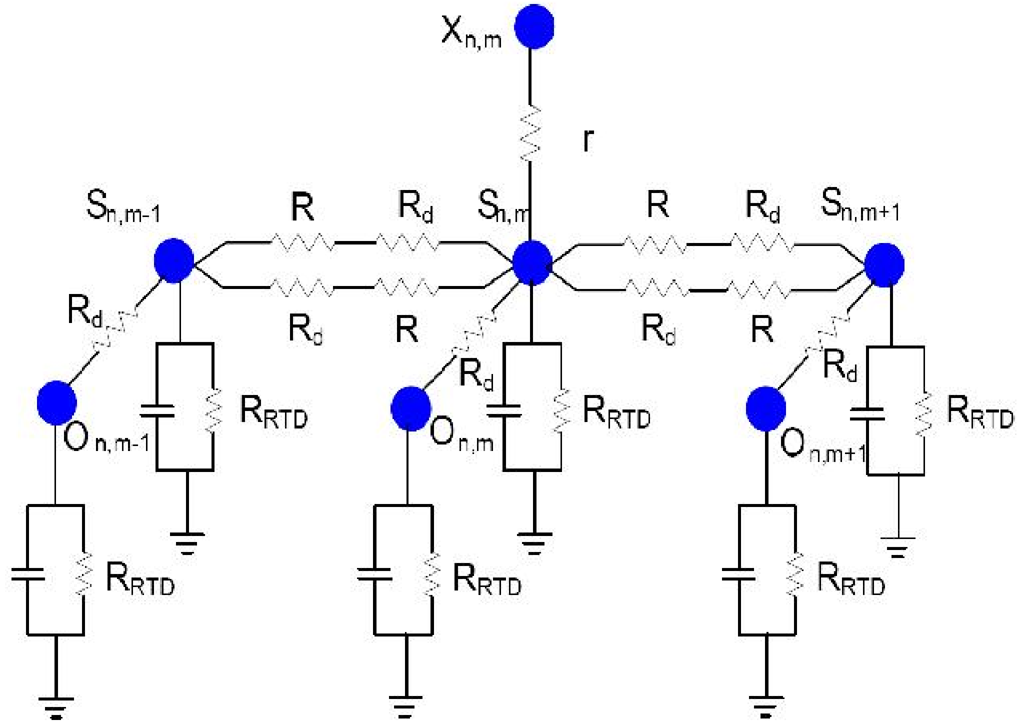


Figure 3.9: RC tree representation of the proposed velocity-tuned filter.

Table 3.1: Performance comparison of VTFs with 20 x 20 pixel images.

	Sequential VTF	Conventional network VTF	Proposed VTF
Area	~1000μm^2	~100μm^2	~10μm^2
Processing Time	0.1 ~ 10 ms (1 GHz)	0.1 ~ 10 μs	0.3 ~2 ns
Power	0.1 ~ 1 mW	0.1 ~ 1 mW	1 ~ 10 μW

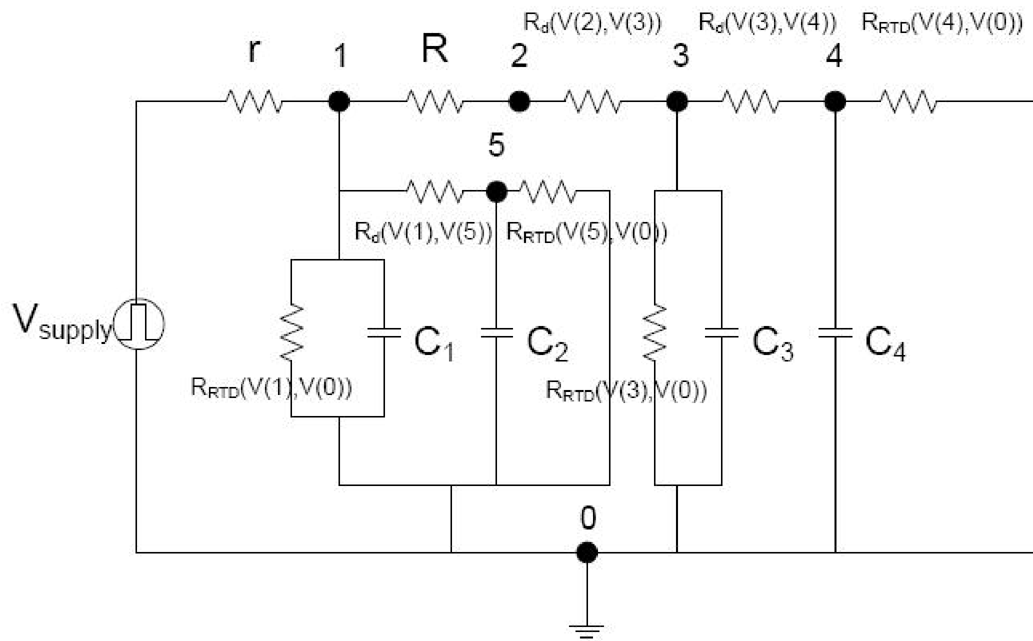


Figure 3.10: Equivalent circuit model of an unit cell of velocity-tuned filters.

3.3.3. Stability of velocity-tuned filter

Stability is an important issue in designing a system. Since the velocity-tuned filter consists of nonlinear elements, namely an array of RTDs, we need to use appropriate methods for examining the stability of the nonlinear system. One such method is the Lyapunov theorem. To apply the Lyapunov method, we define the Lyapunov function, $E(t)$, of the velocity-tuned filter in Eqn. (3.31):

$$E(t) = -\frac{1}{2}\sum_{n,m} h(S_{n,m}(t)) \cdot h(S_{k,l}(t)) + \frac{1}{2}\sum_{n,m} f^2(S_{n,m}(t))$$
$$-\sum_{n,m} I_{n,m}(S_{n,m}(t)). \tag{3.31}$$

Considering $f(S_{n,m})$ is a nonlinear equation, $E(t)$ can be described by

$$E(t) = -\frac{1}{2}\sum_{n,m} h(S_{n,m}(t)) \cdot h(S_{k,l}(t)) + \sum_{n,m}\int_0^{S_{n,m}} f(S_{n,m})dS_{n,m}$$
$$-\sum_{n,m} I_{n,m}(S_{n,m}(t)). \tag{3.32}$$

1) Theorem 1: The Lyapunov function, $E(t)$, of the velocity-tuned filter is bounded by E_{max} when a supplied voltage source is bounded.

Proof:

To prove $E(t)$ is bounded, we first examine $h(S_{n,m})$. Since we assumed that the supplied voltage source is bounded, if we prove that the differential of $h(S_{n,m})$ with respect to $S_{n,m}$ is bounded, we can conclude that $h(S_{n,m})$ is bounded. From Eqn. (3.10), we obtain a differential of $h(S_{n,m})$ with respect to $S_{n,m}$ as

$$h'(S_{n,m}(t)) = \sum_{C(k.l)\in N_r(n,m)} G(N_r(n,m), \overrightarrow{v(t,x,y)}) \cdot (q/kT)$$
$$\cdot e^{q(N_r(n,m)-I_rR-S_{n,m})/kT} \cdot I_s. \tag{3.33}$$

In Eqn. (3.33), $h'(S_{n,m}(t))$ is bounded with given bounded $S_{n,m}$. Therefore, it is obvious that $h(S_{n,m}) \cdot h(S_{k,l})$ is bounded. To prove $f(S_{n,m})$ is bounded, we obtain $f'(S_{n,m})$ from Eqn. (3.9) as

$$f'(S_{n,m}(t)) = b\exp(bn_2S_{n,m})Fn_2 + \frac{\alpha\beta(\frac{\pi}{2} + \arctan(\frac{C-n_2S_{n,m}}{D}))]}{1+b\exp(B-C+n_2S_{n,m})} - \gamma, \tag{3.34}$$

where

$$\alpha = A[1 + b \cdot \exp(B - C - n_2 S_{n,m})], \tag{3.35}$$

$$\beta = \frac{b \cdot \exp(B - C + n_2 S_{n,m})}{1 + b \cdot \exp(B - C - n_2 S_{n,m})} + b \cdot \exp(B - C - n_2 S_{n,m}) \cdot \frac{[1 + b \cdot \exp(B - C + n_2 S_{n,m})] n_2}{[1 + b \cdot \exp(B - C - n_2 S_{n,m})]^2}, \tag{3.36}$$

$$\gamma = \frac{A \cdot n_2 \cdot \ln\left[\dfrac{1 + b \cdot \exp(B - C + n_2 S_{n,m})}{1 + b \cdot \exp(B - C - n_2 S_{n,m})}\right]}{D\left[1 + \dfrac{(c - n_2 S_{n,m})^2}{D^2}\right]} - H \cdot n_2 \cdot b \cdot \exp(n_2 b S_{n,m}), \tag{3.37}$$

where b is q/kT. In Eqn. (3.34), all the exponential functions are bounded with the given function, $S_{n,m}$, which is bounded. Therefore, it is proven that $f(S_{n,m})$ is bounded. In addition, since we have the bounded function, $S_{n,m}$, the current source is bounded. Hence, $I_{n,m}$ is bounded. From the fact that every element of $E(t)$ is bounded, we can conclude that $E(t)$ is bounded.

2) Theorem 2: The differential of the Lyapunov function, $E(t)$, of velocity-tuned filter is less than or equal to zero in the region where $f'(S_{n,m}) \geq 0$, that is

$$\frac{dE(t)}{dt} \leq 0, \; f'(S_{n,m}) \geq 0. \tag{3.38}$$

Proof:

From Eqn. (3.31), the differential of $E(t)$ with respect to time t can be described by

$$
\begin{aligned}
\frac{dE(t)}{dt} &= -\sum_{n,m} \frac{df(S_{n,m}(t))}{dS_{n,m}(t)} \cdot \frac{dS_{n,m}(t)}{dt} \cdot h(S_{k,l}(t)) \\
&+ \sum_{n,m} \frac{df(S_{n,m}(t))}{dS_{n,m}(t)} \cdot \frac{dS_{n,m}(t)}{dt} \cdot f(S_{n,m}(t)) \\
&- \sum_{n,m} I_{n,m} \cdot \frac{df(S_{n,m}(t))}{dS_{n,m}(t)} \cdot \frac{dS_{n,m}(t)}{dt} \cdot f(S_{n,m}(t)) \\
&= -\sum_{n,m} \frac{df(S_{n,m}(t))}{dS_{n,m}(t)} \cdot \frac{dS_{n,m}(t)}{dt} \cdot (h(S_{k,l}(t)) - f(S_{n,m}(t)) + f(S_{n,m}(t)) \cdot I_{n,m}) \\
&= -\sum_{n,m} \frac{df(S_{n,m}(t))}{dS_{n,m}(t)} \cdot \left(\frac{dS_{n,m}(t)}{dt} \right)^2 \cdot C.
\end{aligned}
\tag{3.39}
$$

Since we can assume C is positive in physical meaning, the polarity of $E(t)/dt$ depends on $f'(S_{n,m})$. Assuming $f'(S_{n,m}) \geq 0$ leads to $E(t)/dt \leq 0$. From this theorem, the velocity-tuned filter is stable in a limited region where $f'(S_{n,m}) \geq 0$.

However, our proposed circuit is stable ultimately. If the initial condition of $f'(S_{n,m}) < 0$, $S_{n,m}$ enters the positive differential resistive region where $f'(S_{n,m}) \geq 0$ due to the external current source. The external current source gives a driving force to $S_{n,m}$ from negative differential region to positive differential region.

3) Corollary 1: The output system is stabilized when velocity-tuned filter is stabilized.

Proof:

Assume the Lyapunov function of the output system is $V(t)$. From Eqn. (3.15), the stability of $g(O_{n,m}(t))$ is affected only by $O_{n,m}$ since we know $S_{n,m}$ is a stable function. As a result, the stability of the output system can be examined by the same procedure as that of the velocity-tuned filter. This leads to the same condition of stability, that is, the output system is stable only when $f(S_{n,m}) \geq 0$.

4) Corollary 2: The velocity-tuned filter is asymptotically stable.

Proof:

From theorem 1, 2, and corollary 1, there exists any initial value of $S_{n,m}$ which satisfies the condition below

$$\|S_{0,0}\| < \delta \Rightarrow \|S_{n,m}(t, t_0, S_{0,0})\| < \varepsilon, \ \forall t \geq t_0. \tag{3.40}$$

In addition, it is attractive so that there exists a number $\delta > 0$ such that, for all $t_0 \geq 0$ which can be described by

$$\|S_{0,0}\| < \delta \Rightarrow \|S_{n,m}(t, t_0, S_{0,0})\| \rightarrow 0, \ t \rightarrow \infty. \tag{3.41}$$

However, the velocity-tuned filter does not satisfy the above condition if we assume an infinite voltage source. Because the infinite voltage source can generate infinite current source, the Lyapunov function $E(t)$ is unbounded from theorem 1.

L. O. Chua's work demonstrates the stability of general CNNs and shows that the system is globally stable [1.3]. In his work, the cell function is defined as a function which has a maximum and minimum and a positive slope region. However, our system is different from [1.3] since we use physical model of RTD which is used as a cell function. Also, we claim that our system is asymptotically stable, while Chua's system is stable at all conditions.

3.4. Conclusion

In this chapter, we have described a new velocity-tuned filter that is used for real-time motion estimation. Since the applications of the velocity-tuned filter necessitate tiny, energy efficient and speedy architecture, we have designed the new velocity-tuned filter based on the resonant tunneling diode, which can also be replaced by quantum dots (QDs) at nanoscale fabrication technology. Since it is feasible to fabricate quantum dots with single peak NDR characteristic, RTD's can be replaced by QDs and similar VTF functionality can be realized without requiring a major change in the proposed VTF

architecture. The RTD based design can be compared with operational amplifier based analog neural network design similar to CNN, as proposed by Antonio Torralba in his doctoral dissertation. The experimental results show that the proposed velocity-tuned filter is 10, 100, and 1000 times better than conventional velocity-tuned filters in area, power consumptions, and speed respectively. Since the nanoelectronic architecture embraces nonlinear circuitry, a stability analysis was performed. Based on the analysis, we concluded that the new velocity-tuned filter is asymptotically stable where the derivative of I-V curve of the resonant tunneling diode with respect to I is greater than or equal to 0.

CHAPTER IV
CROSSBAR MEMORY SIMULATION AND PERFORMANCE EVALUATION

4.1. Introduction

4.1.1. Motivation

Scaling of CMOS technology has encountered a serious setback due to the increasing leakage currents of CMOS FET devices in the turned off state and their wide threshold voltage fluctuations due to the process parameter scattering. To overcome the insurmountable limitations of CMOS scaling problems, nanoscale devices such as carbon nanotubes, nanowires and molecular devices are now sought to replace the conventional CMOS devices in ultra-dense digital chips [4.1]-[4.6].

Implementation of these nanoscale devices requires the invention of novel architectures that take advantage of ultra small feature sizes of nanodevices [4.7]. Innovations for nano-architectures at the circuit level and nanocomputer system architectures at the board level are warranted to reap the benefits of nanoscale CMOS technology. Also, defect-tolerant architectures are required since the architectures become severely unreliable due to high defect densities and process variation at nanoscale. Crossbar structures [4.1]-[4.6] in which an active material is sandwiched between two sets of conducting nanowires crossing each other, show promising properties to address these characteristics at both the nano-architecture and the system-architecture level.

The crossbar structure offers many advantages as memory devices can be realized since each cell can be realized as a two-terminal device formed by the two crossed nanowires

trapping a composite material at each cross point. With its simple design, terabit-scale memories excluding peripheral circuits can be realized with the width of the metal lines of the array less than 5nm. The crossbar structure can be non-volatile based on the active material between the two sets of conducting nanowires. This non-volatile characteristic provides instant restart and longer standby operation resulting in protracted battery life.

However, it is not easy to implement general logic operations with the crossbar structure by itself since the structure has difficulty in achieving enough gain and inversion. This difficulty comes from the number of the terminals of the crossbar structure. With two terminal devices, it is very difficult to make an inversion function. Even if inversion function can be achieved, the gain will not be sufficient for it to be used for general inversion logics. To overcome this difficulty, hybrid CMOS/crossbar structures have been proposed [1.34] [1.35]. With these structures, some of the logic functions will be shifted to the underlying CMOS circuitry and the crossbar provides reconfigurable interconnects and wired-OR operation. The system as a whole is expected to present at least two-orders of magnitude higher function density at the same power per unit area and comparable logic delay compared with their CMOS connterparts fabricated with the same design rules. However, fabrication of the crossbar structures on top of CMOS layers present significant challenges such as registration of the CMOS components with the crossbar array. Here, the constraint in attaching a CMOS circuit to crossbar array is how to match the decoder length to the width/length of the array.

In this chapter, a Si-based crossbar memory device has been investigated in Professor Lu's group. Rigorous investigation on the feasibility of volatile and nonvolatile switching devices on Si substrate has been conducted [1.36]-[1.39] because such devices can be fully fabricated with CMOS processing technology. In the perspective of processing technology, these devices are not costly to fabricate and have a simple structure to operate.

4.1.2. Contrast with competing technologies

Among nonvolatile switching nanoelectronic devices, phase change memory devices, magnetic memory devices and molecular memory devices have been considered to be strong candidates to replace the conventional memory device.

Phase change memory devices (PCRDs) have been rigorously investigated for commercialization since 2000. The basic function of the phase change memory devices is to convert the chalcogenide glass state between crystalline and amorphous with the application of heat. PCRDs show high performance arising from fast switching speed than conventional non-volatile memory devices. However, PCRDs have a notable drawback in temperature sensitivity. Since the temperature varies in fabrication processing time and even in memory operation time by users, the temperature sensitivity is a challenging characteristic to PCRDs. In addition, the device requires high programming power and the density of the PCRD is not compatible to a conventional non-volatile memory device, for instance, FLASH [4.11].

Magnetic memory devices (MMDs) have been studied for replacing the conventional memory devices since the 1990s and have made a continuous increase in density. MMDs use magnetic storage elements for storing data, while the conventional devices use electric charge for data memorization. MMD employs electric resistance change of the memory device cell. This change is triggered by the magnetic tunnel effect which changes the orientation of the fields in the two electrodes. Exploiting this characteristic, MMDs achieve fast writing and reading time and low power dissipation. The low power dissipation is achieved because they do not require refreshing all the cells as DRAM and the applied voltage for writing does not need to be much higher than for reading which is common for FLASH memory. However, MMDs retain a notable drawback in fabrication cost. Since MMDs employ the magnetic tunnel effect, the fabrication process should be absolutely changed from the conventional manufacture. This increases the price of the MMDs and became the main factor of the hesitation to investors in industry [4.8]-[4.11].

Molecular crossbar memory devices (MCMDs) are one of the strong candidates for a nanoelectronic memory device. The primary advantage of MCMD is the potential to

achieve high density-an ultimate device may be fabricated with just one molecule in principle. MCMDs can achieve a density of 10^{11} bit/cm2 which is projected technology beyond 2015 for conventional DRAMs. However MCMDs suffer from a number of problems such as low percentage of yield, low on/off ratio, poor thermal stability and slow switching speed [4.12]. There is no known process to reliably fabricate high-density memory arrays based on molecules with high yield. The poor thermal stability further results in reliability problems [1.42].

Crossbar memory device using amorphous Silicon is a striking device which overcomes many of the drawbacks of the other candidates described above. Amorphous Si crossbar memory device can achieve similar cell density but with much better yield percentage compared with MCMDs. Also, it retains temperature stability in processing time and operating time. The most attractive aspect of the amorphous Si crossbar memory device is the compatibility to conventional CMOS process, while other candidates require new processing technology which can not be realized with the conventional CMOS process fabrication facilities. This makes amorphous Si crossbar memory device more attractive than other crossbar memory devices.

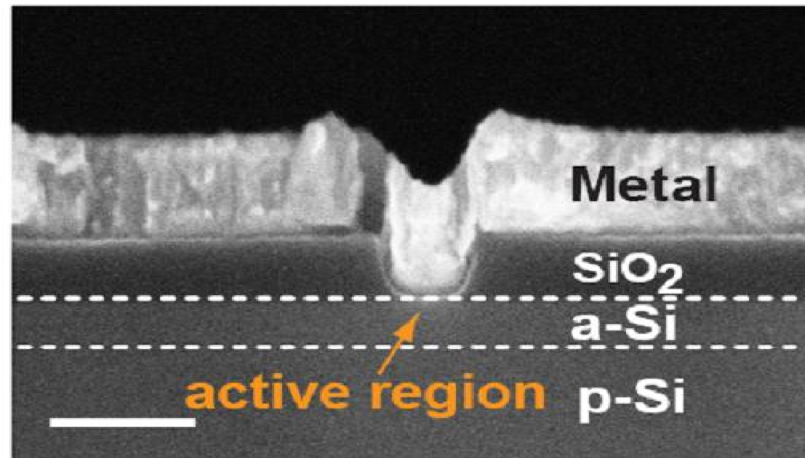
4.1.3. Amorphous Si crossbar memory cell

A M/a-Si/p-Si resistive switching device structure with two terminals was developed in Professor Lu’s group by Sung-Hyun Jo and other students, and has been shown to be a strong candidate for the crossbar architecture [1.40]-[1.41]. These devices promise ultra-high density and intrinsic defect tolerable capability. Also, the M/a-Si/p-Si devices depict comparable scalability and significantly better performance compared with the crossbar structure devices based on molecules [1.42].

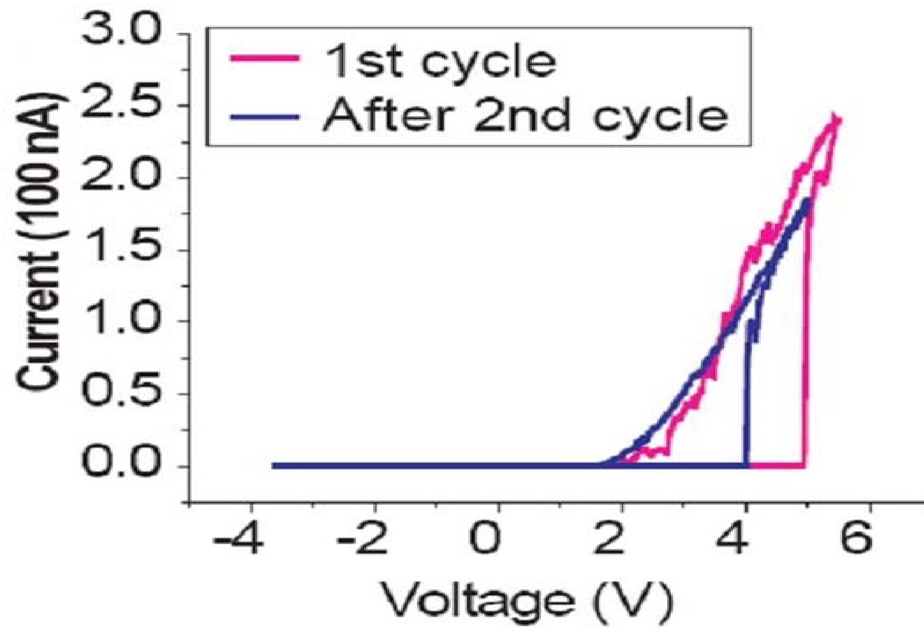
A-Si devices have been studied since the 1960s and 1970s to observe their feasibility for memory application based on resistance switching behavior of amorphous Si [4.14]-[4.22]. However, these conventional metal/a-Si/metal based devices suffer from low yield during a forming process which requires a long and high voltage applied to the device and leading to reliability issues [4.18][4.21][4.22]. Micro-scale filaments have also been observed in these metal/a-Si/metal devices [4.24][4.25] limiting their scaling potential.

For these reasons, few studies on a-Si resistance switching devices have been conducted for memory devices since the 1980s.

In the following, we study the new structure, the M/a-Si/p-Si crossbar devices and the feasibility to apply the structure to ultra-high density memory architecture. To apply it to memory system, electrical modeling of the devices is conducted and the hybrid CMOS/crossbar structure is used to implement the peripheral circuitry in the memory system.



(a)



(b)

Figure 4.1: A crosscut view of an amorphous Silicon crossbar memory cell (a) and *I*-*V* curve (b).

Figure 4.1 shows a basic structure of the amorphous Silicon crossbar memory cell and *I-V* curve of the device. The principle characteristic of the amorphous Silicon crossbar memory cell is programmable resistance of the amorphous Silicon in the crossbar point between Ag and Si as shown in Figure 4.1(b). The resistance of the amorphous Silicon can be changed based on the voltages applied to the crossbar point. Positive high voltage (3.5V to 5V) on the top electrode and ground on the Si bottom electrode lowers the resistance of the amorphous Silicon. The device will remain in the low-resistance '1' state until a negative high voltage reverses the resistance to high resistance state, and this process can be repeated as shown in Figure 4.1(b) [1.41]. Therefore, the high resistance of amorphous can be bit 0 and low resistance bit 1 and this bit assignment can be changed depending on the read circuitry. Compared with earlier studies on metal/a-Si/metal devices, the M/a-Si/p-Si structure offers well controlled forming process, high yield and excellent scalability down to < 20 nm [4.26]. The different behaviors between M/a-Si/M devices and M/a-Si/p-Si devices are likely caused by the reduced density of defects at the a-Si/bottom-electrode interface in the M/a-Si/p-Si structure [4.26]-[4.28].

4.2. Structure

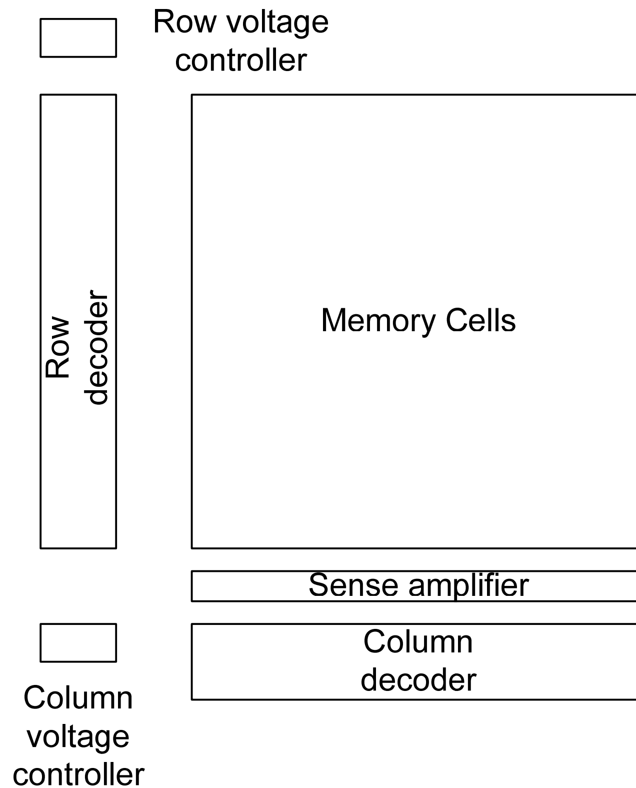


Figure 4.2: Floorplan of the amorphous crossbar memory architecture.

A floorplan of the architecture using crossbar memory cells is shown in Figure 4.2. The crossbar memory units proposed here consist of memory cells, row decoders, column decoders, read circuitry and write circuitry. The difference between the amorphous crossbar memory architecture and conventional memory architectures are the basic memory cells, decoders and row and column voltage controllers.

In conventional CMOS memories, the cell consists of a capacitor in DRAM and a latch which is implemented with two inverters in SRAM. The bi-state values are determined by the charges on a capacitor in DRAM and charges on the latch in SRAM. In the decoding strategy, the conventional CMOS memories uses transistors which operate to select one word line out of n word lines in n addresses memories. The number of transistors used to decode increases as the number of word lines augment sharply. This results in the

incrementing the decoding time for conventional CMOS memories.

In the amorphous crossbar memory, the basic memory cells are implemented with two terminal resistive switching devices. The row decoders and column decoders proposed here are realized by using the diode characteristics of the devices inside another crossbar to minimize the numbers of transistors used. Read circuitry using a trans-impedance amplifier is suggested for the crossbar memory design.

Two terminal resistive switching devices are the main part of the basic memory cells. Two terminal devices are fabricated with metal, amorphous Silicon and crystal Silicon materials. The switching devices are implemented by stacking materials serially in the order of crystal Silicon, amorphous Silicon, and metal from the bottom to top. Device fabrication is conducted by CMOS processing with the exception of the active device area which is defined by electron-beam lithography. As-fabricated devices have characteristics of high resistance between the two electrodes, but when a voltage is applied on the top metal electrode, repeatable resistance switching can be observed as shown in Figure 4.1(b). As a hysteretic resistance switching devices, the M/a-Si/p-Si device demonstrates high yield, fast programming speed, high on/off ratio, long endurance, retention time, and multi-bit/cell capability [1.40].

In memory design, decoders occupy more area as the memory size increases. To reduce the area overhead for large arrays, we suggest a new type of decoders which is compatible with the amorphous Si crossbar array. In addition, to reduce the access time to the target memory as address bits are increased. In this regard, we suggest a new decoder which uses the same device structure as the crossbar memory cell. This decoder design takes advantage of the diode characteristics from amorphous silicon devices which provides the wired-OR function and reduces overall size of the decoder and the number of transistors used compared with a conventional design.

The row decoders and column decoders consist of the diode characteristic devices, NMOS, PMOS, and control signals. Figure 4.3 shows a schematic diagram of the row decoder of a simple 4 x 4 memory array. The decoder requires 4 types of voltage source

to apply appropriate voltages across the crossbar points since we have 3 types for writing and 1 type for reading. For the write operation, high voltage, mid voltage and ground are needed. As shown in Figure 4.3, high voltage across the crossbar points needs to be applied to write “1”. Negative high voltage across the crossbar points is required to write “0”. A mid voltage source is needed to prevent unnecessary cells from being written with “0” or “1”. If we do not have mid voltage, the unselected cells may be accidentally written “0” when we write “1” to the selected cell. In the case of reading, low voltage should be applied to the row of the target cell. The diode model is used for the diode characteristic devices as shown in Figure 4.4. The devices can be obtained by using the same structure as the crossbar memory cell. The row that is connected with PMOS is only enabled to apply voltages based on the function across the target crossbar point and the other rows that are linked with NMOS are enabled so that different outside voltage sources are applied to the rows. This is because when the decoder inputs are assigned whatever inputs it may have, only one row is not enabled (00) since $V_{gs} = 0 < V_{th}$ of the NMOS, while all other rows (11), (10) and (01) will be enabled. In Figure 4.3, the NMOS in the top first row is not enabled, while the other NMOS FETs are enabled by the decoder inputs. The selection of row for writing or reading is conducted by controlling the gate voltage to the NMOS or PMOS.

The main purpose of the decoder is to select the target row and apply appropriate voltages to the target crossbar point while other rows are not selected for programming. In case of writing “1”, the target crossbar point is applied with high voltage and the other points are applied with low voltage. To achieve this function, a high voltage is applied to the target row and mid voltage to the other rows through the selected NMOS FETs and the power supply, as shown in Figure 4.3. For the columns, low voltage is applied to the target row and mid voltage to the other rows.

When we read the value, the read circuitry consists of sense amplifiers, voltage controllers and output buffers. A read circuit which is based on a trans-impedance amplifier is suggested. The difference of resistances of crossbar points causes current difference so that the sense amplifier works based on the charge difference from the current flow of the crossbar points. The trans-impedance amplifier magnifies the current

difference and converts it into the voltage outputs. When implementing the peripheral circuitry, the position of sense amplifiers affects the performance of the memory chip in terms of the area, power dissipation and speed. Also, the number of sense amplifiers can be adjusted by using MUX logic. The number of sense amplifiers also affects the performance of the memory. The performance comparison according to the location and number of sense amplifiers will discussed later in this chapter.

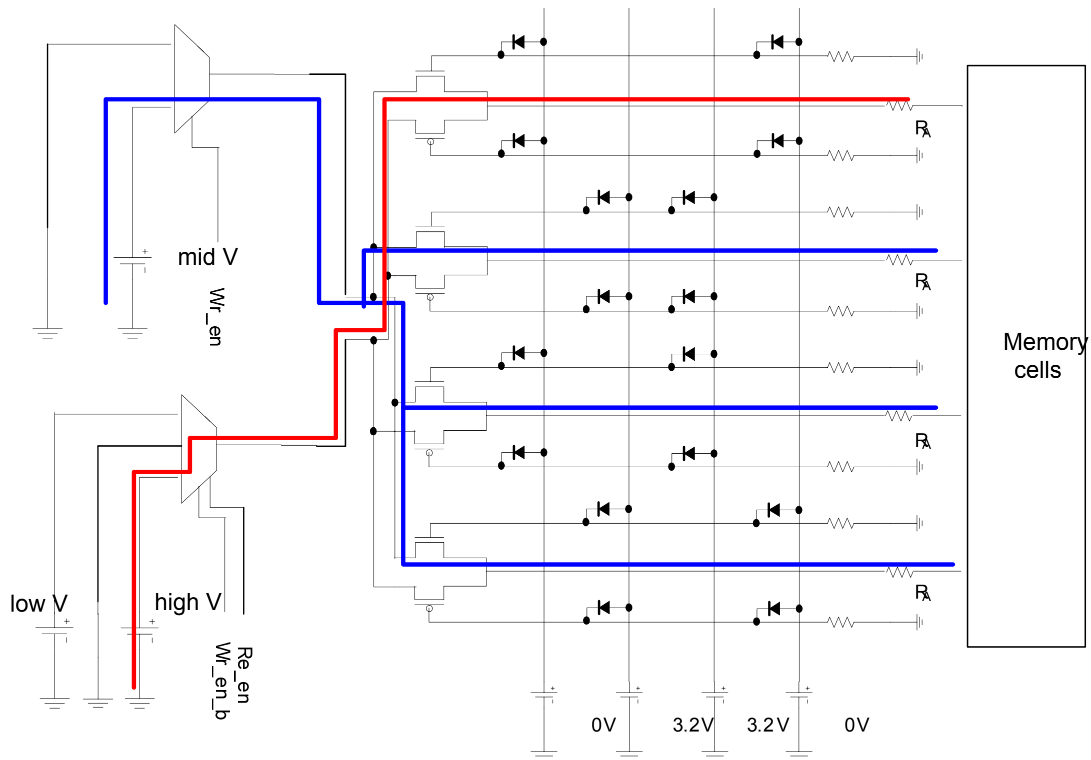


Figure 4.3: A schematic of the row decoder of a 4x4 array with highlighted write “1” operation.

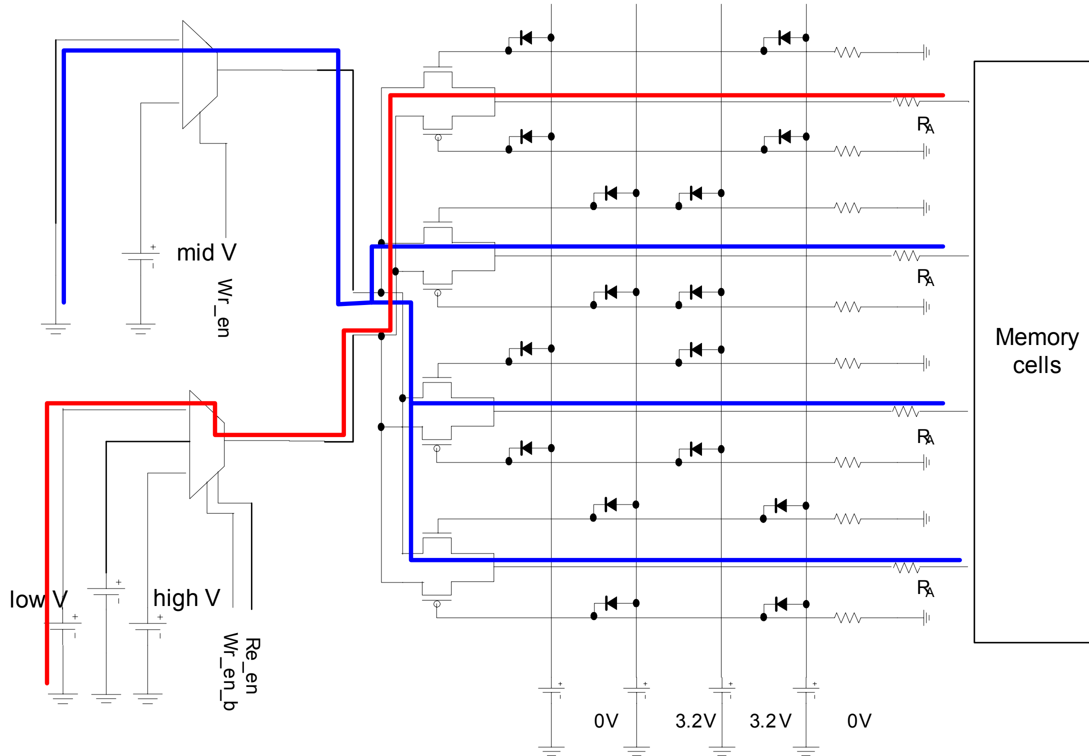


Figure 4.4: A schematic of the row decoder of a 4x4 array with highlighted read operation.

4.2.1. Crossbar modeling

Modeling the crossbar device as a memory device is required for simulation performance analysis. We suggest RC (resistance capacitance) modeling of the device for our simulation. To model the device with resistance and capacitance elements, a cross cut view of the crossbar memory device as shown in Figure 4.5 is convenient. For electrical modeling, we first calculate intrinsic capacitance and resistance of the top metal layer and bottom Silicon electrode and then estimate the coupled capacitance between top metal and bottom Silicon electrode.

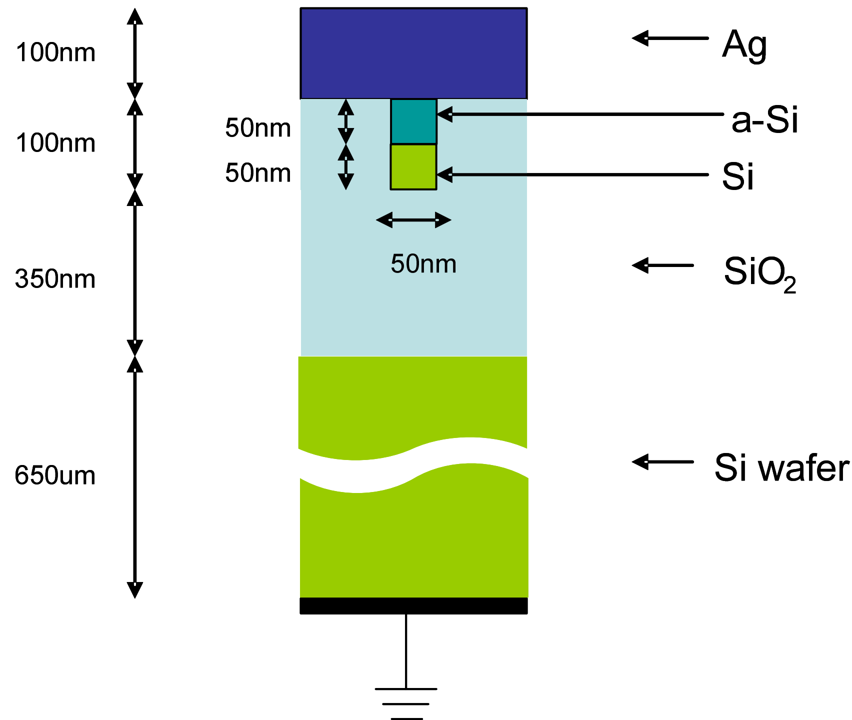


Figure 4.5: A cross cut of the crossbar resistive memory device.

In Figure 4.5, the top electrode is fabricated with silver metal and the bottom electrode is made of Silicon. Amorphous Silicon (a-Si) is inserted in the crossing area of the two electrodes. The two electrodes and a-Si are surrounded with SiO_2 that is mounted on a Silicon wafer. The RC model consists of 2 constant resistors, 1 variable resistor, and 3 capacitors. The 2 constant resistors represent the resistive element of top electrode. The variable resistor demonstrates the electrical characteristic of the amorphous Silicon of the crossbar memory cell. Since the amorphous Silicon changes its resistance according to the applied voltages across it, it is represented as the variable resistor. The resistance values of the amorphous Silicon of the crossbar memory cell ranges from a few KΩ to a few MΩ. The intrinsic resistance and capacitance of the top metal electrode is obtained as follows:

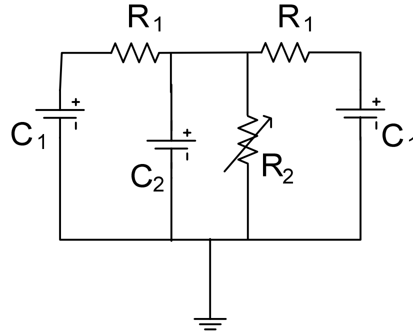


Figure 4.6: A simple RC model of the crossbar resistive memory device.

In Eqn. 2, C_1 represents the capacitance between the top electrodes to the ground. The first term in Eqn. 2 demonstrates the top capacitance and bottom capacitance and the second term represents fringe capacitance between the sides of metal electrode and ground. Similarly, C_2 in Eqn. 3 shows the capacitance between the top electrode and the bottom electrode. The first term shows the top capacitance and bottom capacitance between the top electrode and the bottom electrode and the fringe capacitance between the sides of top electrode and the upper side of the bottom electrode and the fringe capacitance between the sides of the bottom electrode and the base side of the top electrode.

$$\begin{aligned} R_1 &= \rho \frac{l}{S} \\ &= \frac{1.59\times10^{-8}\times200\times10^{-9}}{(100\times10^{-9})^2} \\ &= 0.317\Omega \end{aligned} \tag{4.1}$$

$$C_1 = \varepsilon_1 L 1.15\left(\frac{w}{h}\right) + \varepsilon_2 L 2.80\left(\frac{t}{h}\right)^{0.222} \tag{4.2}$$

$$= \frac{11.9+3.9}{2} \times 8.854\times10^{-12} \times 100\times10^{-9} \times 1.15\left(\frac{100\times10^{-9}}{650.3\times10^{-6}}\right)$$

$$+3.9\times8.854\times10^{-12}\times100\times10^{-9}\times2.80\left(\frac{100\times10^{-9}}{650.3\times10^{-6}}\right)^{0.222}$$

$$=1.377aF$$

$$C_2 = \varepsilon_1 L 1.15\left(\frac{w}{h}\right) + \varepsilon_2 L 2.80\left(\frac{t}{h}\right)^{0.222} \tag{4.3}$$

$$= \frac{11.9+3.9}{2} \times 8.854\times10^{-12} \times 100\times10^{-9} \times 1.15\left(\frac{100\times10^{-9}}{50\times10^{-9}}\right)$$

$$+3.9\times8.854\times10^{-12}\times100\times10^{-9}$$

$$\times2.80\left[\left(\frac{50\times10^{-9}}{50\times10^{-9}}\right)^{0.222} + \left(\frac{100\times10^{-9}}{50\times10^{-9}}\right)^{0.222}\right]$$

$$=37.02aF$$

Modeling the active amorphous Silicon layer requires careful examination of the switching mechanism. Several mechanisms of the have been suggested in the past. One leading theory is that the position of the metallic particle inside the amorphous Silicon is shifted according to the applied voltages. The shift of the position affects the resistance of amorphous Silicon at the crossbar memory cell. This effect is demonstrated in Figure 4.7. When the crossbar memory device has a low resistance value, the metallic particles inside the amorphous Silicon are close to the bottom electrode; otherwise, the metallic particles have enough distance from the bottom electrode to lead to a high tunneling resistance. Since the switching element requires two distinct states, the resistance difference is the key. This device characteristic is electrically modeled as shown in Figure 4.8. The variable resistance from the shift of metallic particles is represented with R_2 and the variable capacitance induced from the distance between the metallic particles and the

bottom electrode is demonstrated with C_2.

However, the resistance of the bottom p-Si electrode in the original memory cell is in the range between 200 Ω and 2 KΩ such that it is not suitable as an electrode because of the propagation delay, voltage drop and power dissipation caused by the high resistance of the bottom electrode as shown in Figure 4.9(a) and (b). Hence, we suggest an improved design by attaching a thin metal line to the Silicon bottom electrode in parallel [4.23]. Using this design, we can reduce the resistance of the bottom electrode in a memory cell to around 2Ω. Figure 4.9(c) shows a 2-dimensional view of the crossbar memory with the improved bottom electrode. Four cells of the crossbar memory are shown in Figure 4.9(c). Also, a circuit schematic of the four crossbar memory cells is shown in Figure 4.9(d). In Figure 4.9(d), R_4 is calculated based on the bottom electrode which consists of a metal line and a semiconductor line connected in parallel.

For the 1Kb crossbar memory architecture shown in chapter 4.5 using this model, we did not exclude nor reduce the circuit elements for the simulation. All the elements described for one cell was included and the coupling capacitance between adjacent metal lines are included in the whole array.

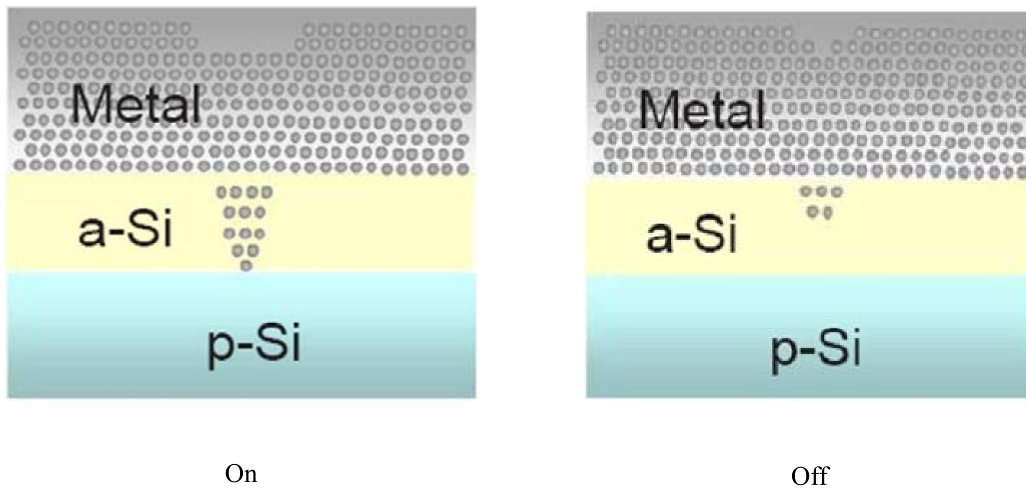


Figure 4.7: A diagram of a cell of the crossbar resistive memory device with on state (left) and off state (right).

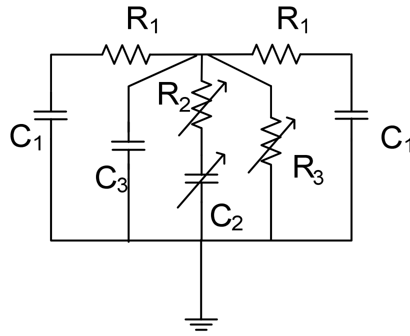
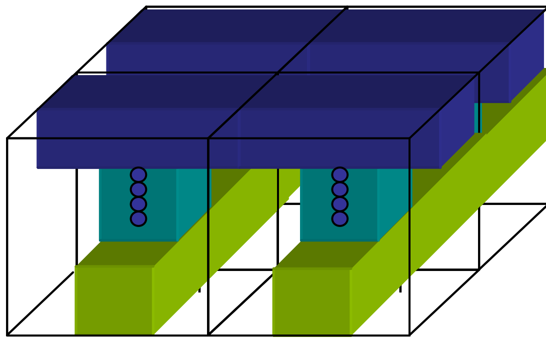
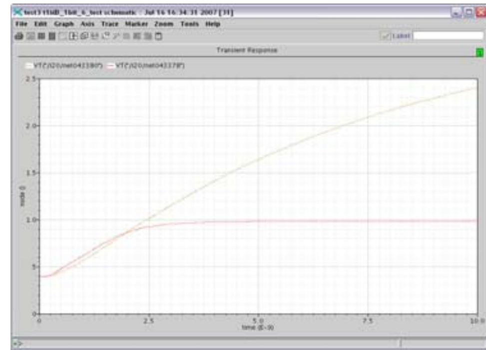


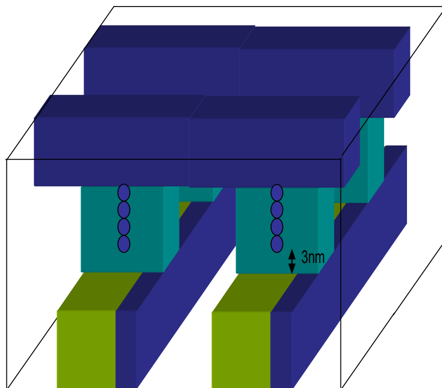
Figure 4.8: A simple RC model of the crossbar resistive memory device.



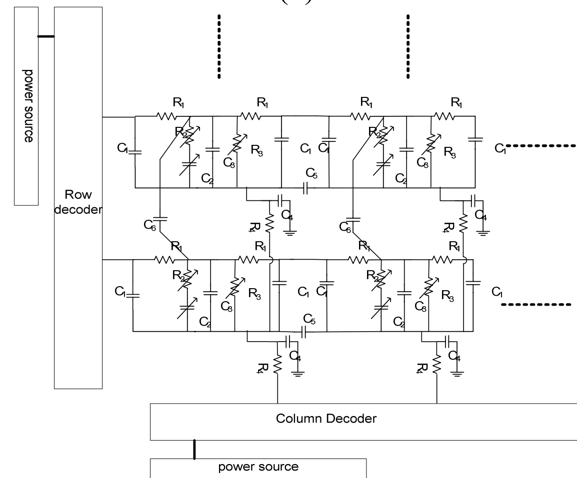
(a)



(b)



(c)



(d)

Figure 4.9: A simple 2-D RC model of the crossbar resistive memory device.

4.3. Write Strategy and Circuit Implementation

Writing is the main function of a memory. By way of writing, the memory has data which is stored in a memory cell. Each memory has its own write strategy. For crossbar memory architecture, we employ power assignment to both sides of the crossbar point for writing. The power assignment is attached with the decoder circuits. Hence, we assign an appropriate power supply to the crossbar point through the decoder circuit.

For writing "1", we need to assign a high voltage to the crossbar point. To implement this function, we assign a high voltage to one row and the other rows are assigned with intermediate voltage in the row decoder as shown in Figure 4.10. Therefore, we need to select a high voltage in the lower multiplexer (MUX) and an intermediate voltage in the upper MUX in the row decoder. We assign voltages to the column decoder oppositely as shown in Figure 4.11. For example, to write "1" to the crossbar memory cell in the fourth row and the first column, high voltage is assigned to the sources of NMOS FETs which are connected to lower MUX, while middle voltage is assigned to the sources of PMOS FETs. For the decoder inputs, the fourth row is assigned to high voltage and the other three rows are assigned to middle voltage. We select a ground voltage to the lower MUX and an intermediate voltage to the upper MUX in the column decoder.

For writing "0", a negative high voltage is applied to the crossbar point. This function is implemented by assigning a ground voltage to one row and the other rows are assigned with intermediate voltage in the row decoder as shown in Figure 4.12. Therefore, we choose a ground voltage in the lower MUX and an intermediate voltage in the upper MUX in the row decoder. However, opposite voltages need to be assigned in the column decoder as shown in Figure 4.13. We choose a high voltage in the lower MUX and an intermediate voltage in the upper MUX in the column decoder.

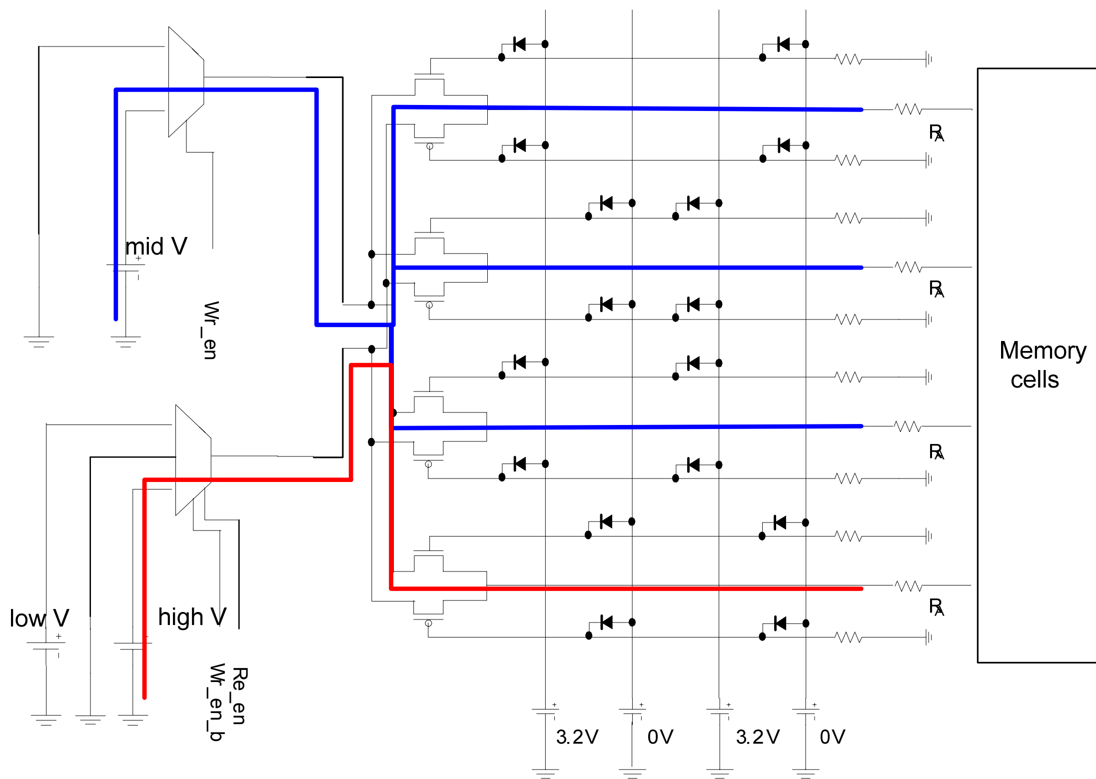


Figure 4.10: Row decoder and row power assignment for writing “1”.

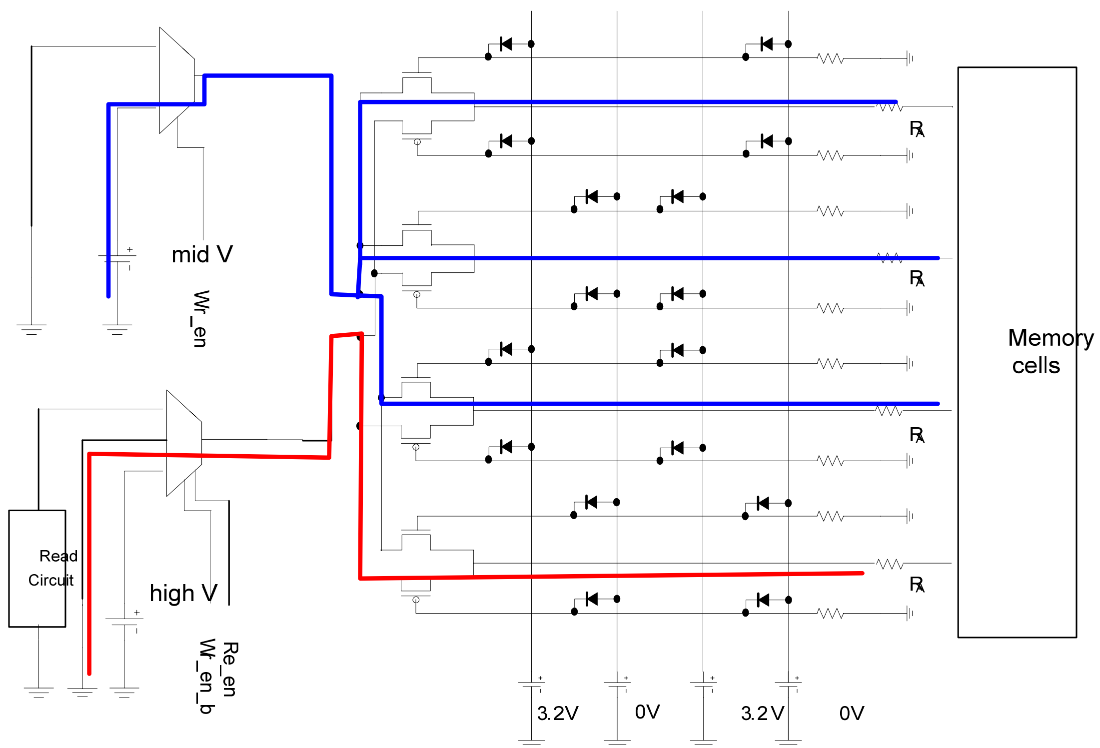


Figure 4.11: Column decoder and column power assignment for writing “1”.

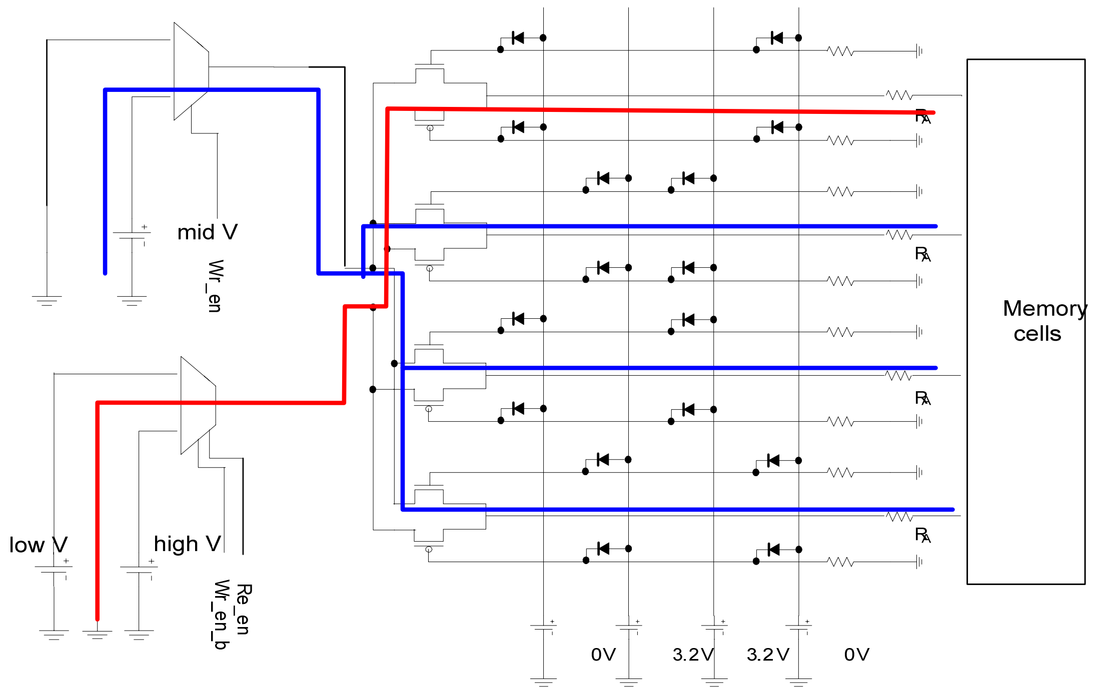


Figure 4.12: Row decoder and row power assignment for writing “0”.

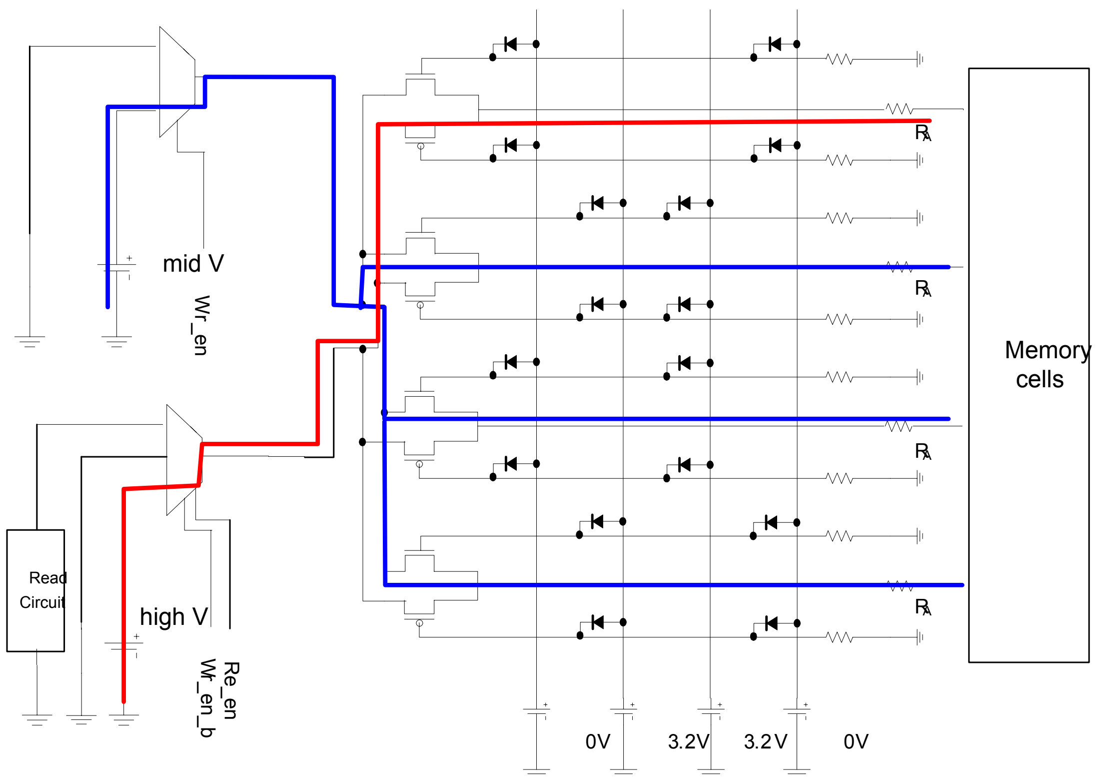


Figure 4.13: Column decoder and column power assignment for writing “0”.

4.4. Read Strategy and Circuit Implementation

Reading in the crossbar memory requires a different strategy. The main difference of the reading in the crossbar memory is that we use a small voltage to measure the resistance value of the cell. In the new circuit design, we assign zero voltage in the upper MUX and low voltage in the lower MUX in the row decoder in Figure 4.4. However, we connect a ground voltage in the upper MUX and a read circuit in the lower MUX in the column decoder in Figure 4.4.

As mentioned in Section 2, an amplifier can be implemented to detect the value of the resistances. Since we apply a low voltage in the row decoder, the different resistance values in the crossbar points result in differences of current through the crossbar points. Therefore we suggest a differential trans-impedance amplifier which detects the difference in the current and outputs in the form of voltages.

Figure 4.14 shows a sense amplifier for a conventional CMOS memory circuit. This sense amplifier consists of current-mirror amplifier and the bit lines are connected to the gate which controls the current flow from Vdd. When Bit_bar is higher than Bit, M1 and M3 turn on and increase the current through M1 and M3. This increases the current through M2, and decreases the current through M4. Therefore, the Out voltage increases and decreases Out_bar voltage. When Bit_bar is lower than Bit, M1 and M3 are turned off and the current through M1 and M3 decrease. This results in turning off M2 and makes M4 and M5 operate in the deep triode region. This results in turning off M2 and makes M4 and M5 operate in the deep triode region. Because there is no current flow from M2, M4 and M5, the voltage at Out is decreased to zero. This type of sense amplifier is used for single ended output.

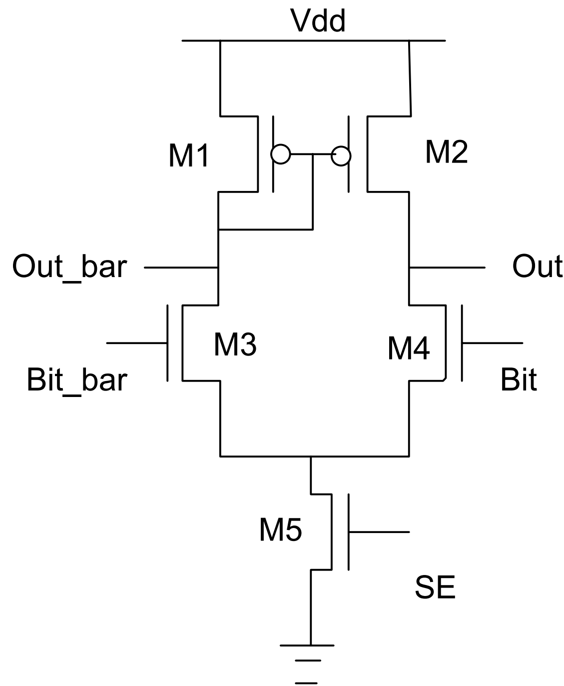


Figure 4.14: Sense amplifier using the single-ended current-mirror amplifier.

Figure 4.15 illustrates a sense amplifier which is commonly used for conventional CMOS memory design. This sense amplifier employs a cross-coupled amplifier. The Bit line and the Bit_bar line are connected with gates of the pull-down differential pairs. This sense amplifier can be implemented with combining two single-ended current mirror amplifiers. When Bit is higher than Bit_bar, Bit, M1 and M3 are turned off and the current through M1 and M3 decrease. This will turn off M2 and makes M4 and M5 operate in the deep triode region and decrease the voltage at Out node to zero. However, when Bit is lower than Bit_bar, Bit, M2 and M4 are turned off and the current through M2 and M4 decrease. This will turn off M1 and makes M3 and M5 operate in the deep triode region and decrease the voltage at Out_bar node to zero.

Figure 4.16 demonstrates a sense amplifier which is commonly used for SRAM. This sense amplifier manipulates a latch which consists of two inverters. To operate this sense amplifier, equalization between Bit node and Bit_bar node should be conducted. Providing a PMOS which connects source to Bit and drain to Bit_bar, the equalization function can be operated. After the equalization, SE is enabled to make the sense amplifier function. When Bit is higher than Bit_bar, the latch makes Bit increase to Vdd

and Bit_bar decrease to Gnd. When Bit is lower than Bit_bar, the latch makes Bit increase to Gnd and Bit_bar decrease to Vdd.

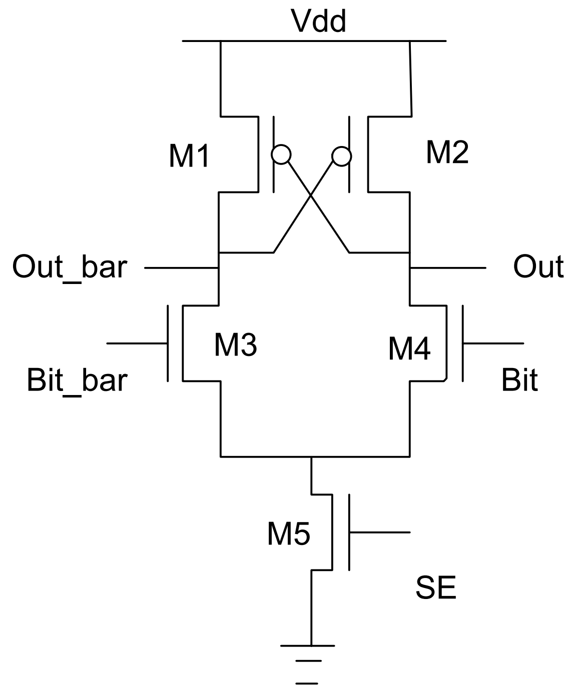


Figure 4.15: Sense amplifier using cross-coupled amplifier.

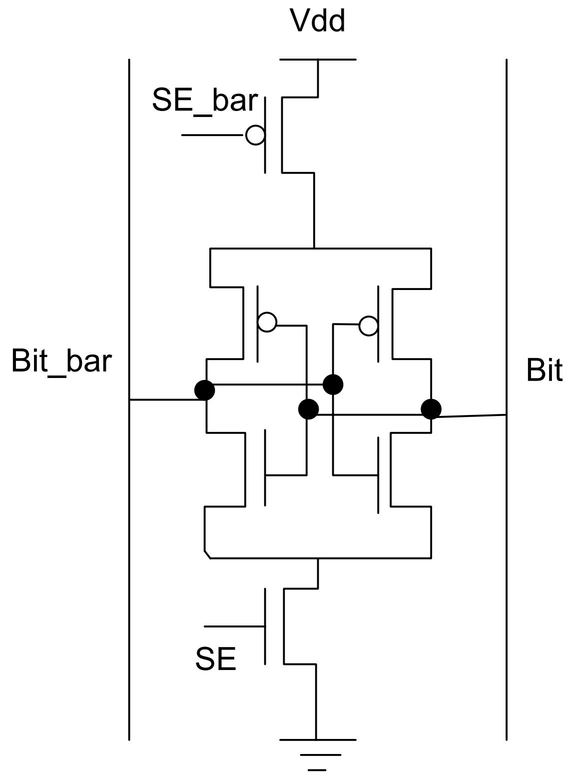


Figure 4.16: Sense amplifier using a cross-coupled CMOS inverter latch.

However, these sense amplifiers can not be used for the amorphous crossbar memory architecture. The cross-coupled amplifier is not suitable for the crossbar memory read circuitry since it operates based on the difference in input voltages. To detect the change of resistance, we should differentiate the current change through the crossbar resistance with a supplied voltage. The cross-coupled CMOS inverter latch can not be used by itself for the crossbar memory circuitry since the crossbar memory device does not have a latch based cell as SRAMs. The latch based cell can flip the Bit or Bit_bar depending on the stored value in the cell during read mode.

Hence, we employed a reference resistor and a differential amplifier which can produce the current difference. Then, this current difference is detected through a latch which is connected to the output node.

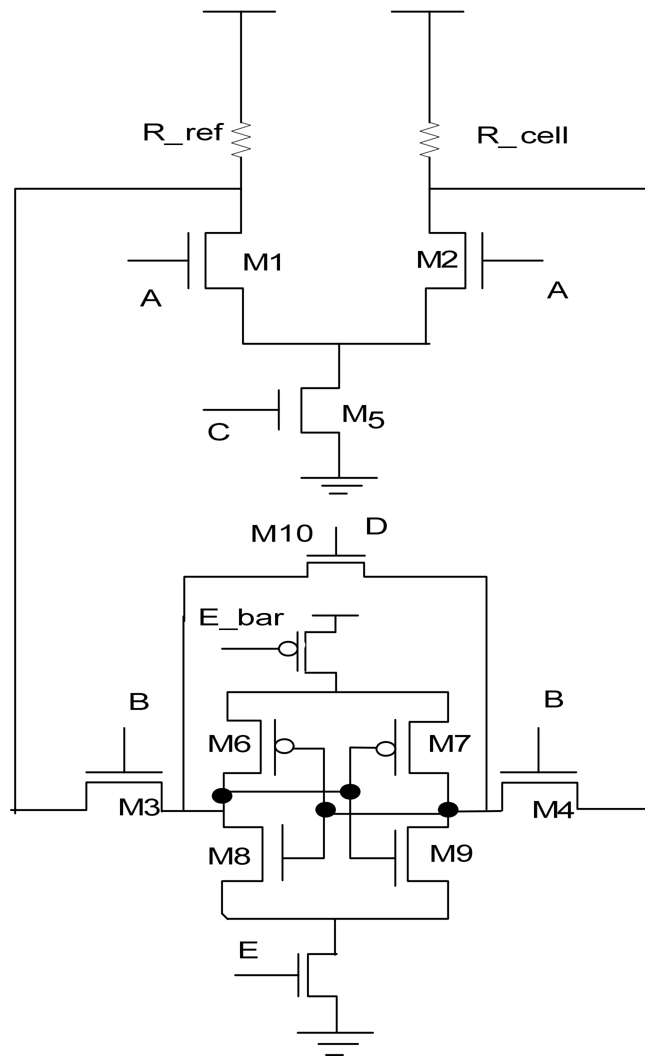


Figure 4.17: Suggested sense amplifier for the amorphous crossbar memory using a trans-impedance differential amplifier which senses current difference.

Figure 4.17 shows the read circuit which senses the current difference through the resistance elements. This sense amplifier uses the principles of a cross-coupled inverter latch in Figure 4.16 and differential amplifiers in Figure 4.14 and Figure 4.15. The

suggested amplifier magnifies the current flow using the differential amplifier, and the magnified current affects the values inside a latch. Based on experimental results of the resistances of the crossbar points, we assigned the resistances of the crossbar points. They have low resistance values in the range from 1 KΩ to 5 KΩ or high resistance values in the range from 1 MΩ to 5 MΩ. Hence, we select a reference resistance with a value of 100 KΩ between the two ranges. The currents through the resistance of the crossbar point and reference resistance are different and produce the signal. This difference is amplified through a sensing circuit, which consists of 2 inverters connected in series. The current flows from the resistances to the sensing circuit are controlled by external signals. This control is implemented with 2 pass gates in Figure 4.17. Finally, since the sensing circuit requires equalization before sensing, a pass gate which connects the two inverters is implemented.

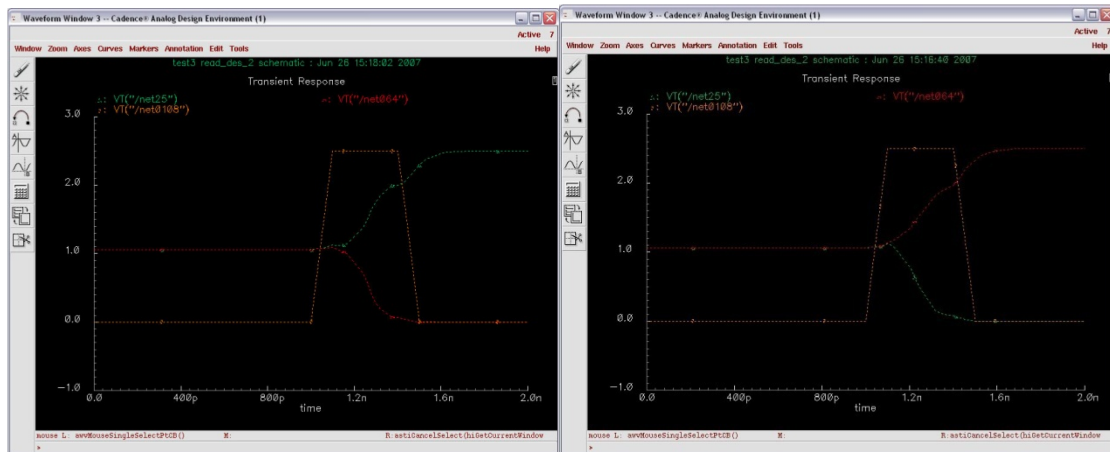


Figure 4.18: HSPICE simulation results of the read circuit with resistances of 1kΩ (top) and 1MΩ (bottom).

Figure 4.18 demonstrates the HSPICE simulation results of the read circuit. The left figure depicts the simulation results with resistance of the crossbar point of 1 KΩ. As shown in Figure 4.18, the output values of the two inverters are equalized until 1 ns and then differentiated with the two current flows when the two pass gates are enabled. The output is stabilized after 1.6 ns. The right figure demonstrates the simulation results with resistance of the crossbar point of 1MΩ.

Generally, sense amplifiers are sensitive to transistor sizing. The sizing influences the slew rate of the differential amplifier, the power dissipation and the speed of operation. At first, the slew rate depends on the ratio between PMOS and NMOS of the latch. The PMOS and NMOS ratio can be controlled by the width manipulation since the length is generally fixed based on the process. To control the slew rate, equal width of the PMOS and NMOS was implemented. Then, the width of PMOS was gradually increased until the rising and falling time are equalized. In this simulation, the PMOS and NMOS width ratio was 1.6:1. Also, the equalization voltage is the major factor for the slew rate. We selected the mid voltage between Vdd and Gnd for the equalization voltage.

After generating optimal slew rate, gate sizing should be conducted for the optimal performance. In this simulation, sizing the gates M1, M2 and M5 are main factors in the performance perspective. To be operated as a differential amplifier, the gate M5 should operate as a current source. The current source can be implemented with sizing gate M5 to be operated in the saturation region. In the saturation region, the current is not affected by Vds. In the sense amplifier circuit, we have to bias M5 so that the voltage fluctuation in the drain of M5 does not affect the current flow through M5. Then, when the M1 and M2 are enabled by the input A, There is current difference between the two paths which are attached to registers. Obviously, the bias voltage should be above the threshold voltage Vth. However, if we increase the voltage near Vdd, the power dissipation will be worse. In contrast, if we bias M5 near threshold voltage, there might be a chance for M5 operate in the triode region. Also, it will degrade the speed of the sense amplifier. In this sense, we tried to find the optimal bias voltage of M5 for the power-delay product minimization. Based on the simulation results, we biased M5 with 1.1V. For the optimal power-delay product, the widths of the gates, M1, M2 and M5 should be chosen

optimally. If we increase the widths of the gates, M1, M2 and M5, then the speed will be improved but this degrades the power dissipation and vice versa. In this simulation, we chose the minimum size of width for the savings because the writing time and reading time are more affected by the decoders and long array lines.

4.5. Memory Architecture

In this section, a 1 Kb (32 x 32) crossbar memory design using writing and reading circuits is presented. The 1 Kb crossbar memory design requires 5 bits for column decoder inputs and 5 bits for row decoder inputs. The column and row decoder inputs are decoded in 32 bits respectively. The decoded bits are connected to 1 Kb arrays in the crossbar memory. By controlling the decoder inputs, one and only one of the 1 Kb array cells is selected.

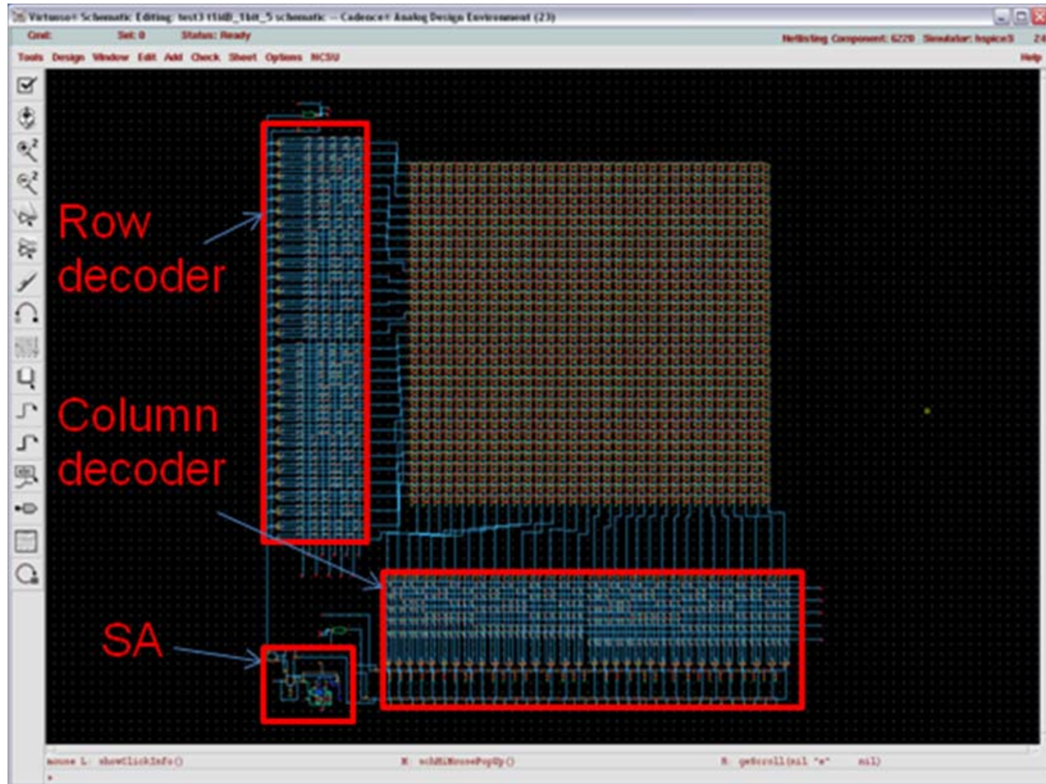


Figure 4.19: Schematic of a 1 Kb memory design with a single sense amplifier after the column decoder.

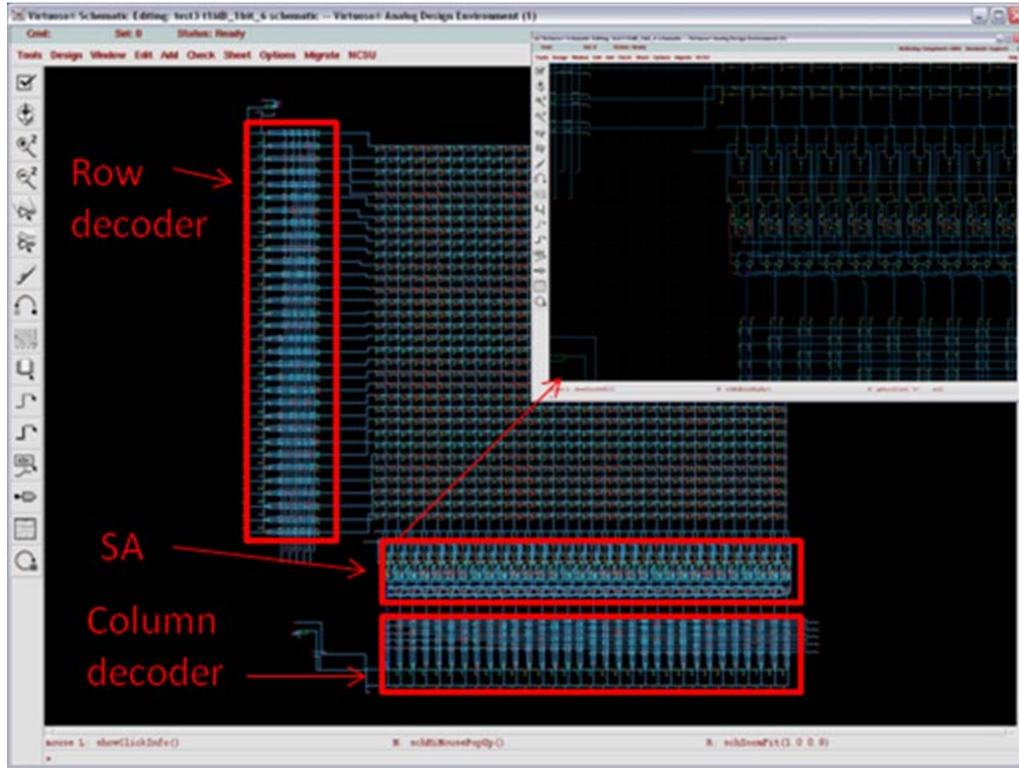


Figure 4.20: Schematic of a 1 Kb memory design with an array of sense amplifiers before the column decoder.

Two different approaches to design the 1 Kb crossbar memory design will be discussed. The first approach is to implement the memory with a single sense amplifier which is attached after the column decoder as shown in Figure 4.19. In the second approach, an array of sense amplifiers can be attached with each bit line as shown in Figure 4.20.

The 1 Kb memory design in Figure 4.19 uses a single sense amplifier, and different bit lines are read after selection by the column decoder. This memory architecture uses fewer numbers of sense amplifiers than the memory in Figure 4.20 by the number of the bit line times. Hence it has less area overhead and static power consumption than the 1 Kb

memory design in Figure 4.20.

However, the memory as shown in Figure 4.20 require less switching time for bit line changes. This is because the sense amplifiers latch values which are ready for the decoder. When we read multiple bits sequentially, the design in Figure 4.20 only requires additional decoding time for additional read after initial sense amplification time. Therefore, when the bit line selection is changed, the output only requires additional decoding time. However, in the memory design in Figure 4.19, when reading multiple bits in a word, extra time is needed for the sense amplifier to initialize when reading every single bit. Therefore the memory design in Figure 4.19 is much slower than the design in Figure 4.20 when we read multiple bits sequentially.

Figure 4.21 describes a HSPICE simulation result of the memory as shown in Figure 4.20. In Figure 4.21, curve (a) represents a signal of the sense amplifier connected with the selected decoder. The inverse signal of the sense amplifier is indicated as curve (b). The input signal which connects between sense amplifier and the column of the memory array is shown as curve (c). The output signal “high” is marked as curve (d).

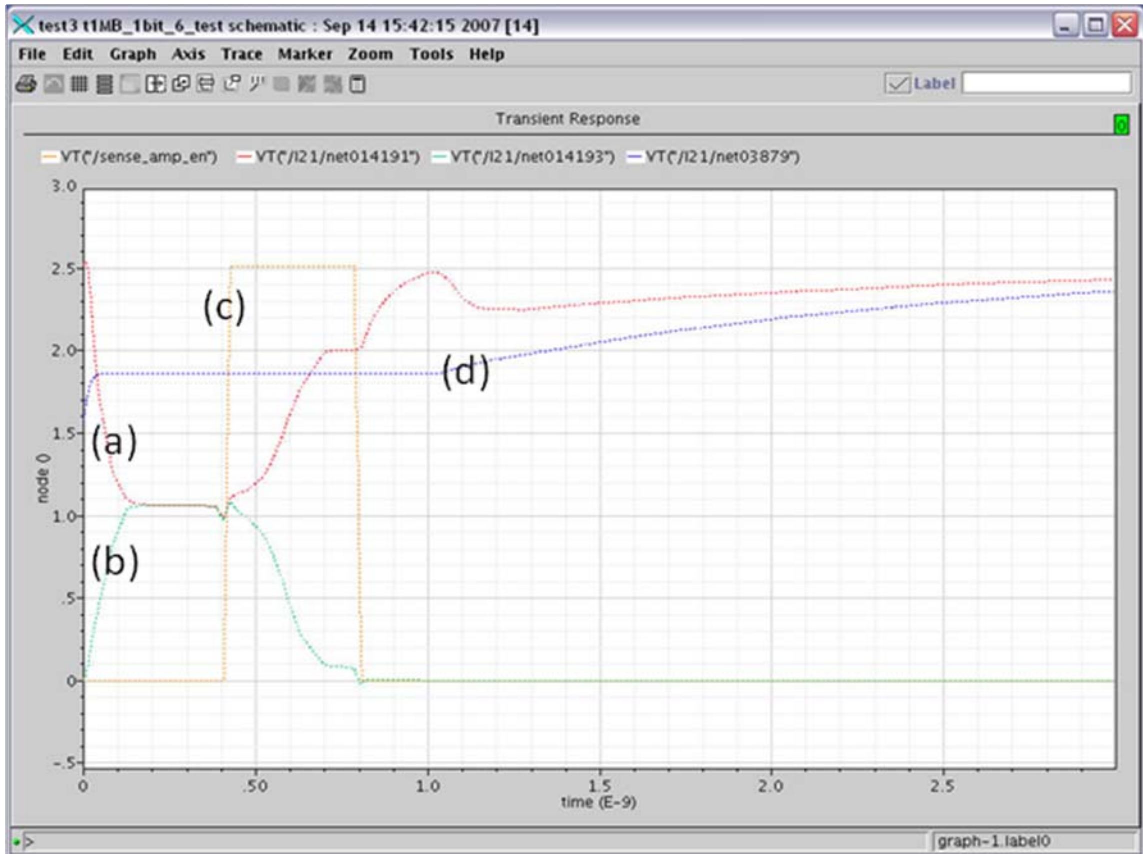


Figure 4.21: HSPICE simulation results of the memory with sense amplifiers before the column decoder for reading "high".

The HSPICE simulation result of the memory in Figure 4.22 shows a case when the output signal is "low". In Figure 4.22, (a)-(c) can be referred to signals. However, (d) describes the output signal of the decoder. This is attributed to the threshold voltage of PMOS of the column decoder. Since the low output signal passes through the PMOS, the output signal of the decoder is not the ground voltage. To address this issue, a buffer is added after the decoder. The output signal after the output buffer is described as (e).

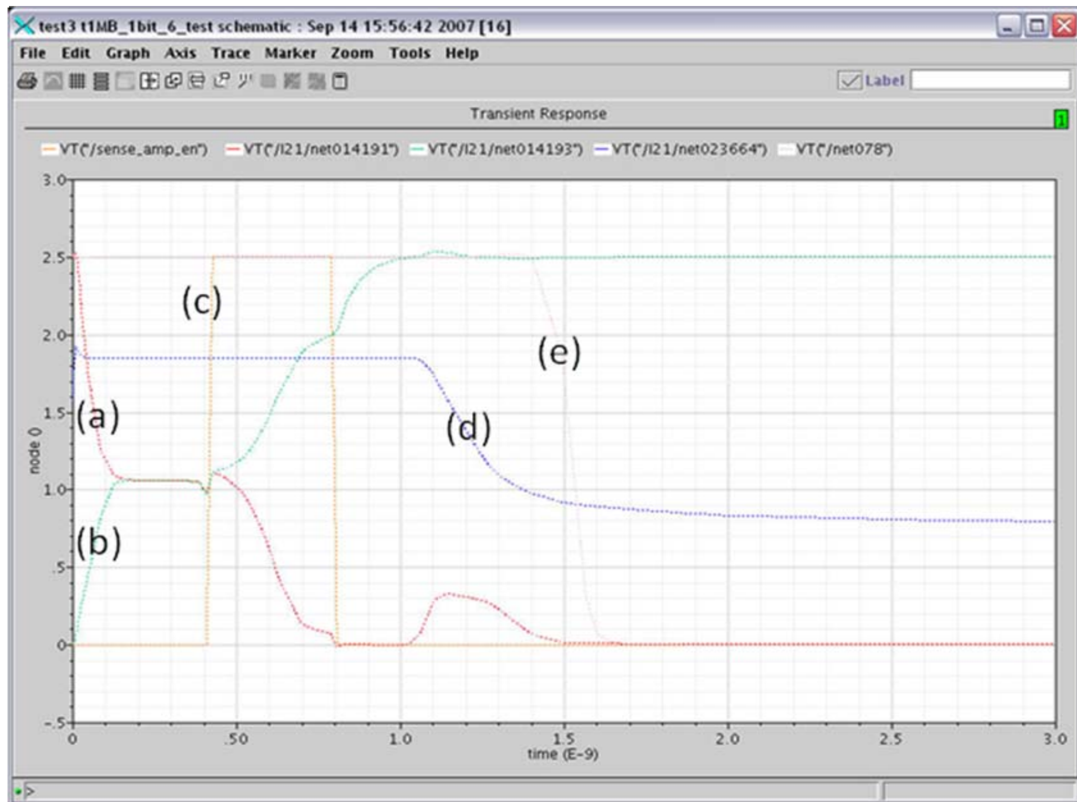


Figure 4.22: HSPICE simulation results of the memory with sense amplifiers before the column decoder for reading “low”.

Table 4.1: Area composition of the 1 Kb crossbar memory design with a SA attached to all the bit lines.

	Horizontal width	Vertical width
Array	32 x 1.2 μm	6.4 μm
Decoder	2 μm	2 μm
SA	32 x 1.2 μm	20 μm
Total	68.8 μm	28.4 μm

Table 4.2: Area composition of the 1 Kb crossbar memory design with a single SA at the final stage.

	Horizontal width	Vertical width
Array	6.4 μm	6.4 μm
Decoder	2 μm	2 μm
SA	6.6 μm	2 μm
Total	8.4 μm	10.4 μm

Table 4.1 shows area estimation for the 1Kb crossbar memory design with sense amplifiers attached to all the bit lines. In this design, the sense amplifiers occupy 32 x 20 x 1.2 um^2 of area. The sense amplifiers occupy most area of the design since it uses 10 transistors to implement. The fact that horizontal width is much longer than vertical width is because the sense amplifiers are attached with all bit lines. Hence, the sense amplifiers determine the pitch width of the crossbar memory design.

Table 4.2 shows area estimation for the 1 Kb crossbar memory design with a single sense amplifier in the final stage. In this design, the crossbar arrays occupy the most area of the

design. Horizontal width and vertical width are similar in the design because the sense amplifier does not affect pitch width. Therefore, the 1 Kb crossbar memory design with a single sense amplifier in the final stage occupies less area than the 1 Kb crossbar memory design with sense amplifiers before decoders, as discussed earlier.

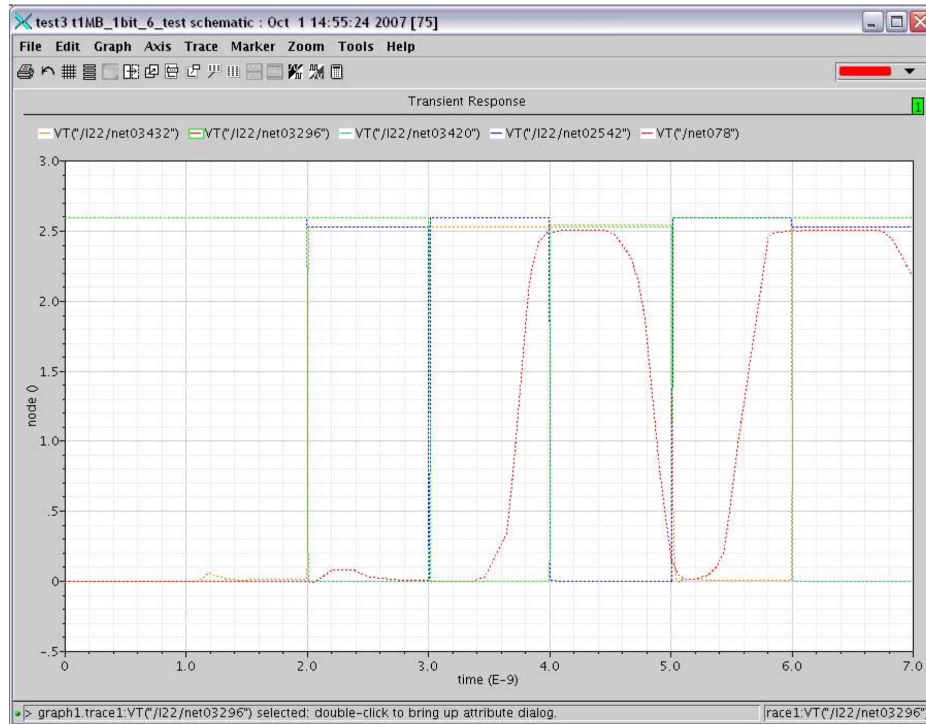


Figure 4.23: HSPICE simulation results of the 1Kb crossbar memory design with sense amplifiers before the column decoder for reading.

The HSPICE simulation result on reading operation of the 1 Kb crossbar memory design with sense amplifiers before decoders is shown in Figure 4.23. In Figure 4.23, the net078 with red line represents the output signal of the memory. The output signal changes its

value according to the resistance of the crossbar memory cell. In this simulation, the crossbar memory cell is selected by the column address. In other words, the column address selects a crossbar memory cell, whose resistance affects the output signal. In Figure 4.23, the other nets represent column address bits. The column address bits change by 1 ns and the corresponding output also changes by 1 ns. In this simulation, the output can be read as "0101" and this output implies that the resistances of the target row are "low high low high".

Table 4.3: Comparison of data read time between the 1Kb memories with an array of SAs before decoders and the 1 Kb memory design with a single SA after decoders.

	SA after decoders	SA before decoders
Reading	32 ns + 4ns	3.5 x 32 ns
Reading / bit	1.1 ns / bit	3.5 ns / bit

Table 4.3 shows the comparison of reading times on both two crossbar memory architectures. As shown in Table 4.3 the crossbar memory architecture with sense amplifiers before decoders are approximately three times faster than the other architecture in reading 32 bits of 1Kb crossbar memory. This originates from the difference in initial time for the sense amplifiers as discussed earlier. Since the crossbar memory design with a sense amplifier at the last stage requires initialization time for the sense amplifiers when reading a bit, this architecture demands initialization 32 times when reading 32 bits. On the other hand, the crossbar memory with sense amplifiers before decoders needs initialization only once when reading 32 bits. Hence, while the

memory with sense amplifiers before decoders requires 3.5 ns per bit for reading, the other memory needs 1.1 ns per bit for reading.

The simulation results shown in section 4.5 demonstrate pros and cons between the two design approaches, SAs before decoder design and a SA after decoder design. Pros of one design become the cons for the other design and vice versa. Pros of SAs before decoder design are the fast processing time and better noise immunity and cons are the area overhead and increased power consumption due to larger number of SA's. The processing time for SAs before decoder design arises from early and multiple latches of the bit information. In this design the latched information is decoded by decoders and as the decoder changes the address the output only requires one time of SA processing for one row of information. However, the SA after decoder design requires the decoding time and SA processing time for every bit leading to much more time for reading and writing. In area overhead, the SA after decoder design requires one SA while the SAs before decoder design requires as many SAs as the number of bit lines which results in large area overhead. In the noise point of view, the SAs before decoder design is better than the other counterpart because the SAs before decoder design holds the latched data before decoding which is connected to power source passing through one closed transistor. However the SA after decoder is vulnerable to noise because the decoded data are not connected to the power source directly through the transistor. In the power dissipation, the SAs before decoder design induces larger number of switching activity of SA for reading or writing within a period of reading a word because of the number of SA. In addition, the design consumes more static power due to leakage from gate or drain to source which will be discussed in the next section.

4.6. Power Dissipation

4.6.1. Power estimation

Power dissipation of the memory system is one of the critical factors in performance measurement. The reason is that power dissipation affects how the circuit is designed, e.g.

the number circuits on a single chip, power-supply capacity, etc., and it also affects feasibility, cost and reliability.

In the crossbar memory system design, we compare power dissipation of both the designs. Since both designs have different circuit configuration, we investigate how it affects the power dissipation. First, we measure the power dissipation related with the sense amplifiers. Subsequently, decoders and other circuits are examined for the power dissipation.

The power dissipation of the sense amplifiers can be categorized into 3 parts. One is the power dissipation from the reference voltage source which works for the differential amplification. This dissipation is continuous during the reading period because the reference voltage source should be enabled through the whole reading operation. Also, the reference voltage source is related with direct path power dissipation from the power supply to the ground. This direct path is constructed because the gate to source voltage is above the threshold voltage of the NMOS and there is no device that blocks the current path from power supply to the ground.

Another power dissipation source of the sense amplifiers is control signals on the equalization and the latch inside sense amplifiers. This power dissipation is correlated with dynamic power consumption since the control signal repeats on and off during the reading operation. The dynamic power consumption can be defined as

$$P_{dyn} = C \cdot V_{dd}^{2} \cdot f$$

where C is load capacitance and f is switching frequency. The last power dissipation source of sense amplifier is static power consumption. Since the CMOS devices are not ideal, the NMOS and PMOS devices do not operate in steady-state simultaneously.

Therefore, there is a leakage current from the drain to source or the substrate to source resulting from reverse-bias diode junctions of the transistors. Another source of leakage current is sub-threshold current of the CMOS devices. Even if the voltage across the gate and source is below the threshold voltage, there is a leakage current from drain to source. This leakage current depends on the threshold voltage. If the threshold voltage is close to zero volts, the leakage current is larger resulting in larger static power dissipation. For both sources of leakage current, the static power dissipation can be expressed as follows:

$$P_{stat} = I_{leakage} \cdot V_{dd} .$$

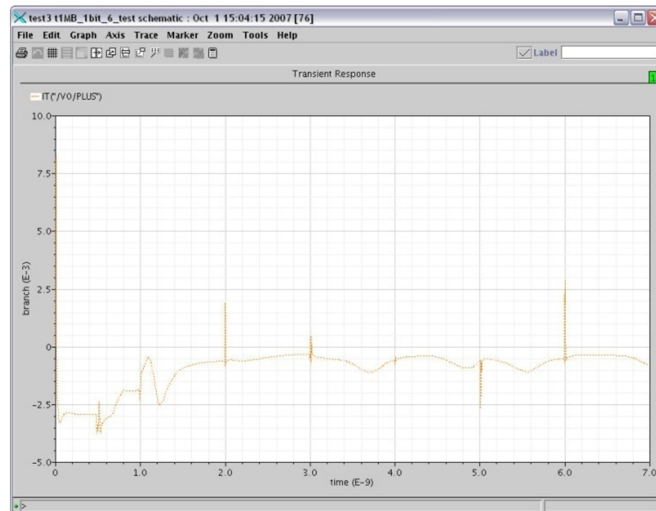


Figure 4.24: Static power dissipation of the SA part in the 1Kb crossbar memory design with a SA after decoders from 0 ns to 7 ns.

Figure 4.24 shows the static power dissipation of the 1 Kb crossbar memory design with a sense amplifier after the column decoders. The glitches in Figure 4.24 originate from the short period time when the input signal changes. The power dissipation of the sense amplifier part is shown in Figure 4.25.

Generally, width of PMOS and NMOS in a sense amplifier should be increased as size of an array increases because of the increased capacitance of bit line in an array. The current change during a read operation is decreased due to the increased capacitance of the bit line. It becomes more difficult for the sense amplifier to detect the current change, which affects the output voltage of the sense amplifier. However, the width of a sense amplifier is increased less than 1.5 times for the 1 Kb crossbar array compared with that of the smallest sense amplifier satisfying the design rules. We observe that that amount of the increment is enough to sense the current difference and does not increase time required for sense amplifier operation through the bit line in the 1 Kb crossbar memory design. Therefore, we should find width of the sense amplifier in the 1 Kb crossbar memory designs which optimize the power-delay product for the crossbar memory design.

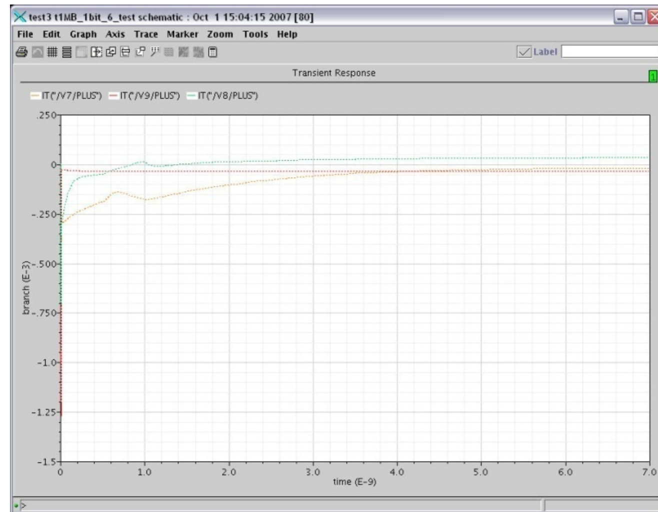


Figure 4.25: Power dissipation of the SA part in 1Kb crossbar memories with a SA after decoders from 0 ns to 7 ns.

As shown in Table 4.4, total power dissipation of a sense amplifier after decoders is three times less than the other circuit configuration. The main reason is that the crossbar memory with sense amplifiers before decoders requires as many sense amplifiers as the bit lines. Since the number of sense amplifiers of crossbar memory design with sense amplifiers before decoders is 32 times more, the power dissipation of the memory is 32 times larger than the power dissipation of the counter design if we assume that the changes of the input signal is equal during the operation period.

However, the variations of input signals can also be different during a reading operation. Input signals of the crossbar memory with sense amplifiers before decoders requires only one time equalization operation and one time enabling the latch operation for reading 32 bits. On the other hand, the crossbar memory with a sense amplifier at the final stage demands both equalization operation and enabling the latch operation every time reading a bit. Therefore, the time for reading 32 bits in the crossbar memory with a sense

amplifier at the final stage is approximately 3 times longer than the counterpart. Data regarding the sense amplifiers in Table 4.4 shows what is mentioned above.

In the case of the decoder, power dissipation is dynamic power consumption. Variation of the decoder input affects the dynamic power dissipation. Variations of the decoder input in both the designs are different. The crossbar memory with a sense amplifier at the final stage alters its decoder input approximately three times less than the counterpart. Therefore, the power dissipation of the crossbar memory with a sense amplifier at the final stage is approximately three times less than the counterpart as shown in Table 4.4. In total power dissipation, the crossbar memory with a sense amplifier at the final stage is approximately three times less than the counterpart design.

Considering the size of memory, the power dissipation illustrated in Table 4.4 is high. To build a larger memory array, we need to provide ways to reduce power dissipation at the expense of other performance. Since decoder consumes most of the power, reduction of power dissipation from the decoder will be most effective. One way is to replace the crossbar decoder with conventional CMOS decoders. Then, we will lose the area overhead advantage but will save power more than 90% since the power dissipation in crossbar decoder comes from direct path current flow from Vdd to Gnd. Another way is to reduce the voltage source for reading. Then, while we save the power, we will lose reading speed. Finally, power gating will be effective to the power saving for a large memory cells.

However, we measured the power dissipation for the worst case which means continuous writing to the crossbar memory cells. Also, when the crossbar memory is large (e.g. contains more than 50 billion memory cells), the power dissipation is not proportional to the size of memory directly since when we read or write, we only read or write a word each time. Hence the issue is how we can effectively minimize the static power dissipation while maintaining high performance. Since the array does not consume any power when the applied voltage is Gnd, the static power from decoder, SA and control logics are the only concerns. Therefore, the static power dissipation of the crossbar memory design is much less than the conventional memory designs.

Table 4.4: Comparison of power dissipation between the two types of 1 Kb crossbar memory cell designs.

	SA after decoders	SA before decoders
SA	1.5 mW	4.5 mW
Decoder	110 mW	310 mW
Others	4 mW	4 mW
Total	115.5 mW	318.5 mW

As shown in Table 4.3 and Table 4.4, there is a tradeoff between the reading time and the power dissipation. Superiority in the reading time results in inferiority in the power dissipation. The crossbar memory design with sense amplifiers before decoders shows better performance in the reading time, while it demonstrates the poorer performance in the power dissipation. Hence, the counterpart design shows the opposite performance in the reading time and the power dissipation.

4.6.2. Analytical modeling on static power

In the crossbar memory design, power dissipation is one of important parameter for the memory. As discussed in Section 4.6.1, dynamic power dissipation and static power dissipation are the power dissipation types. In long channel devices, dynamic power is important since the leakage current is negligible in this case. However, as the device size scales down, static power dissipation becomes important. The absolute and the relative contribution of leakage power to the total power become more important since the leakage current increases exponentially with the technology scaling. Based on

International Technology Roadmap for Semiconductors, the leakage current will contribute 50% of the total power dissipation in the next generation.

In the ideal transistor, current only flows when Vgs is greater than Vth. However, the current still flows in the real transistor even below the threshold voltage. This conduction can be expressed as given in Eqn. (4.4) and Eqn. (4.5) where V_T is thermal voltage [4.13].

$$I_{ds} = I_{ds0} e^{\frac{Vgs - Vt}{nV_T}} (1 - e^{\frac{-Vds}{V_T}}) \quad (4.4)$$

$$I_{ds0} = \beta V_T{}^2 e^{1.8} \quad (4.5)$$

I_{ds0} is the current at threshold voltage and this value varies with process and device geometry. Also, n is a process-dependent term depending on the depletion region characteristics. For the equations, we can think the leakage current is a function of temperature and threshold voltage when we turned off the device because the gate voltage is zero and Vds is Vdd for CMOS operations.

In the beginning, row decoders and column decoders can be analytically modeled. To investigate the decoders we need to look into the circuit at the gate level. As shown in Figure 4.26, when a selecting bit is enabled, the leakage power dissipation occurs in the upper NMOS and PMOS which are circled. This power dissipation can be expressed as given in Eqn. (4.6).

$$I_{LMUX} = I_{LNMOS} + I_{LPMOS} \quad (4.6)$$

Considering the width and length of the device, the equation can be transformed as given in Eqn. (4.7).

$$I_{LMUX} = \frac{W_1}{L_1} I_{LNMOS} + \frac{W_2}{L_2} I_{LPMOS} \quad (7)$$

Since we use the middle voltage source for the MUX, the leakage power dissipation of

the MUX can be expressed as given in Eqn. (4.8).

$$P_{LMUX} = \frac{W_1}{L_1} V_M I_{LNMOS} + \frac{W_2}{L_2} V_M I_{LPMOS} \tag{4.7}$$

In the case of 3x1 mux shown in Figure 4.3, the leakage current can be calculated in the same way and can be expressed as given in Eqn. (4.8).

$$I_{LMUX3} = \frac{W_1}{L_1} I_{LNMOS} + \frac{W_2}{L_2} I_{LPMOS} \tag{4.8}$$

Since three power sources are used, the power dissipation for writing "1" is expressed as given in Eqn. (4.9).

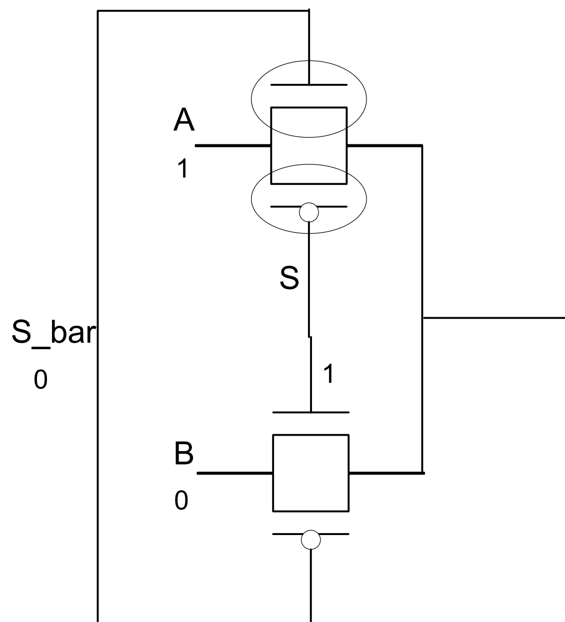


Figure 4.26: A 2x1 MUX for power selection which is connected to row decoders.

$$P_{LMUX3} = \frac{W_1}{L_1} V_H I_{LNMOS} + \frac{W_2}{L_2} V_H I_{LPMOS} \tag{4.9}$$

For reading, the power dissipation can be expressed as given in Eqn. (4.10).

$$P_{LMUX3} = \frac{W_1}{L_1} V_L I_{LNMOS} + \frac{W_2}{L_2} V_L I_{LPMOS} \tag{4.10}$$

However, when we write "0", we do not consider the leakage current since the power source is connected to Gnd in row decoders.

In the case of decoders, static power dissipation occurs since the power source is directly connected through diode. If we assume the resistance of the diode is R, the power dissipation through one diode can be expressed as given in Eqn. (4.11).

$$P_{DIODE} = \frac{V_H^2}{R} \tag{4.11}$$

The number of addresses, N_{addr}, decides the number of diodes in the decoder and the power dissipation from the diodes with N address decoders can be represented as given in Eqn. (4.12).

$$P_{DNaddr} = N \log_2 N \frac{V_H^2}{R} \tag{4.12}$$

For the column decoders, since the MUX uses high voltage when writing "0", the leakage power dissipation can be expressed as given in Eqn. (4.13).

$$P_{LCMUX} = \frac{W_1}{L_1} V_H I_{LNMOS} + \frac{W_2}{L_2} V_H I_{LPMOS} \tag{4.13}$$

If we assume the number of columns is the same as the number of rows, the power dissipation from the diodes in the column decoders can be represented as given in Eqn. (4.14).

$$P_{DCNaddr} = N \log_2 N \frac{V_H^2}{R} \tag{4.12}$$

Secondly, the sense amplifier is analyzed for the leakage current. As shown in Figure 4.

17, the sense amplifier should be analyzed based on the operation mode. At first, an idle mode is defined when A = 0, B = 0, C = 1, D = 0 and E = 0. In the idle mode, two NMOS FETs in the differential amplifier connected with A suffers from the leakage current and the NMOS connected with E also suffers from the leakage current. The leakage power dissipation in the idle mode can be expressed as given in Eqn. (4.13).

$$P_{SEIDLE} = 2\frac{W_1}{L_1}V_H I_{LNMOS} + \frac{W_2}{L_2}V_H I_{LNMOS} \tag{4.13}$$

Next, the precharging mode should be analyzed and this mode is defined when A = 0, B = 0, C = 1, D = 1 and E = 0. In this mode, two NMOS FETs in the differential amplifier connected with A suffers from the leakage current and the NMOS connected with E also suffers from the leakage current. The leakage power dissipation in the idle mode can be expressed as same as the idle mode leakage power given in Eqn. (4.14).

$$P_{SEPRE} = 2\frac{W_1}{L_1}V_H I_{LNMOS} + \frac{W_2}{L_2}V_H I_{LNMOS} \tag{4.14}$$

In the reading mode, the transistors which are connected with A, B, C, D and E are enabled. In this mode, the transistor M5 suffers from the leakage current since C is in between Vth and Vdd. The latch suffers from the leakage current since either of one node of the latch should be zero. Two transistors in the off state in the latch will suffer from the leakage current. M6 and M9 or M7 and M8 depending on the output value suffer from the leakage current in the latch. Therefore, the leakage power dissipation in the read mode can be expressed as given in Eqn. (4.15).

$$P_{SEREAD} = \frac{W_1}{L_1}V_H I_{L2NMOS} + \frac{W_2}{L_2}V_H I_{LNMOS} + \frac{W_3}{L_3}V_H I_{LPMOS} \tag{4.15}$$

In Eqn. (4.15), I_{L2NMOS} is much higher than I_{LNMOS} or I_{LPMOS} since V_{gs} is not zero in this device. Therefore the sizing M5 is very significant to minimize the leakage current of the sense amplifier.

Considering the number of sense amplifiers connected to all the bit lines, the power dissipation will be increased as the number of bit lines are increased. If we assume that we read 1 bit in each read operation and do not use any MUX and the sense amplifiers are

connected to the bit lines, the leakage power dissipation can be represented as given in Eqn. (4.16).

$$P_{SEREAD} = N\left(\frac{W_1}{L_1}V_H I_{L2NMOS} + \frac{W_2}{L_2}V_H I_{LNMOS} + \frac{W_3}{L_3}V_H I_{LPMOS}\right) \tag{4.16}$$

Considering the frequencies in the idle, precharging and read mode, those frequencies are combined to generate the overall leakage dissipation in the sense amplifiers. If we assume that the idle time, Ti, precharging time, Tp and reading time, Tr, then the overall leakage power dissipation can be written as given in Eqn. (4.17).

$$P_{SE} = \frac{N \cdot Tr}{Ti + Tp + Tr}\left(\frac{W_1}{L_1}V_H I_{L2NMOS} + \frac{W_2}{L_2}V_H I_{LNMOS} + \frac{W_3}{L_3}V_H I_{LPMOS}\right) + \frac{N \cdot (Tp + Ti)}{Ti + Tp + Tr}\left(2\frac{W_1}{L_1}V_H I_{LNMOS} + \frac{W_2}{L_2}V_H I_{LNMOS}\right) \tag{4.17}$$

It is intriguing to research the effect of the usage of MUX logics to reduce the number sense amplifier on the leakage current. If we use a 2 by 1 MUX, then the number of sense amplifiers becomes half while additional leakage current may be added from the 2 by 1 MUX. Therefore, the leakage current of both sense amplifiers and the 2 by 1 MUX can be represented as given in Eqn. (4.18).

$$P_{SE_MUX} = \frac{N \cdot Tr}{2(Ti + Tp + Tr)}\left(\frac{W_1}{L_1}V_H I_{L2NMOS} + \frac{W_2}{L_2}V_H I_{LNMOS} + \frac{W_3}{L_3}V_H I_{LPMOS}\right) + \frac{N \cdot (Tp + Ti)}{2(Ti + Tp + Tr)}\left(2\frac{W_1}{L_1}V_H I_{LNMOS} + \frac{W_2}{L_2}V_H I_{LNMOS}\right) + \frac{W_1}{L_1}V_M I_{LNMOS} + \frac{W_2}{L_2}V_M I_{LPMOS} \tag{4.18}$$

In a similar manner, extending to an N by 1 MUX will leads to the leakage power dissipation as given in Eqn. (4.19).

$$P_{SE_MUX} = \frac{Tr}{(Ti+Tp+Tr)}\left(\frac{W_1}{L_1}V_H I_{L2NMOS} + \frac{W_2}{L_2}V_H I_{LNMOS} + \frac{W_3}{L_3}V_H I_{LPMOS}\right) + \frac{(Tp+Ti)}{(Ti+Tp+Tr)}\left(2\frac{W_1}{L_1}V_H I_{LNMOS} + \frac{W_2}{L_2}V_H I_{LNMOS}\right) + \log_2 N \cdot \left(\frac{W_1}{L_1}V_M I_{LNMOS} + \frac{W_2}{L_2}V_M I_{LPMOS}\right) \quad (4.19)$$

Based on the analytical modeling, it is imperative to properly size the transistor and the diode characteristic device to minimize the leakage power dissipation. Especially, the NMOS and PMOS of the latch can be sized to reduce leakage current, and the bias transistor for constant current source should be carefully sized to meet the performance criteria.

4.7. Noise Analysis

Noise has been an issue in the design of digital circuits of comparable importance to timing and power. Increased noise has been attributed to increasing interconnect densities, faster clock rates and scaling threshold voltages. Increasing interconnect densities represent a significant increment in coupling capacitance. Faster clock rates mean faster slew rate, and threshold voltages scale lower with scaling supply voltages. These effects merge to create more sources of on-chip noise.

In the crossbar memory architecture, the array of the memory cells contains dense metal lines. Interconnect density is increased due to the dense metal lines. The increased interconnect density becomes a source of on-chip noise which is called interconnect coupling noise.

The coupling noise is mainly due to capacitive coupling between metal lines. This noise is more relevant to the crossbar memory since it contains more cells than conventional memories resulting in denser interconnect capacitance.

To examine the interconnect capacitance in the crossbar memory design, we ignored C_2, which simplifies the RC model of the amorphous Silicon crossbar memory as shown in Figure 4.27.

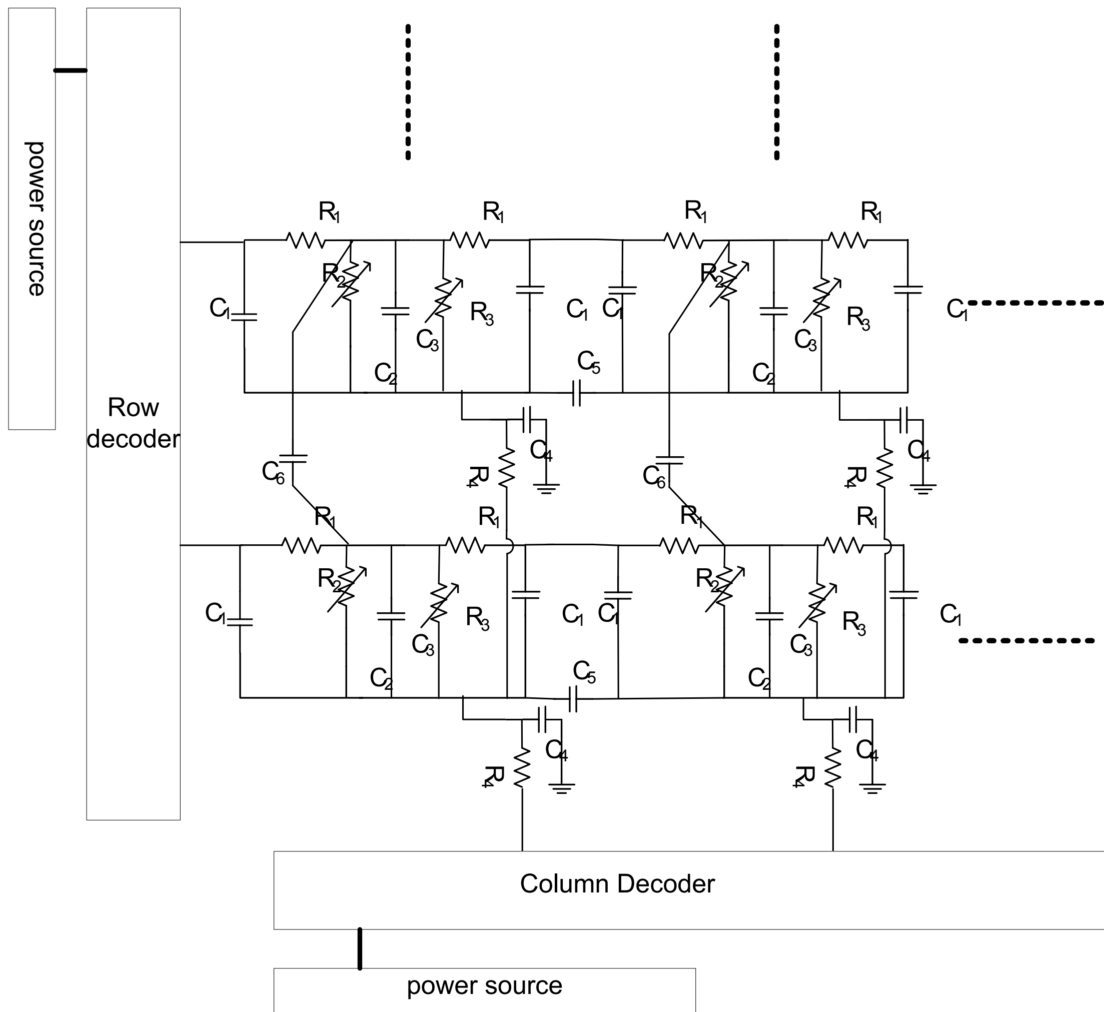


Figure 4.27: Simplified RC model of crossbar memory cells.

If we assume that the upper metal line is an aggressor while the bottom one is a victim, we can model the circuit as a capacitive voltage divider to compute the victim noise as shown in Figure 4.28. The aggressor's voltage becomes V_{high} as we apply high voltage to the selected row. If we assume a victim is a floating node, ΔV_{victim} without a victim's driver is represented in Eqn. (4.20).

$$\Delta V_{VICTIM} = \frac{C_6}{\frac{(2C_1 + C_3)\cdot C_4}{(2C_1 + C_3) + C_4} + C_6} \Delta(V_{AGGRESSOR}) \quad (4.20)$$

Assuming an N x N array and V_{mid} is one half of V_{high}, Eqn. (4.20) can be modified as Eqn. (4.21).

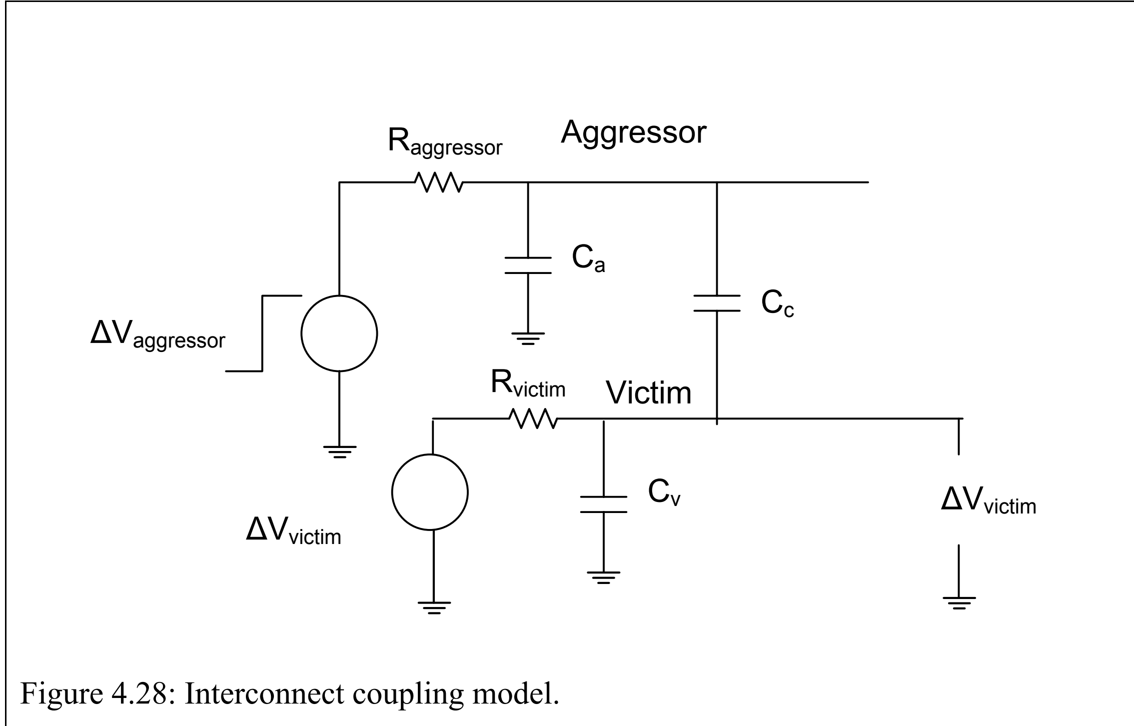


Figure 4.28: Interconnect coupling model.

Since we supply V_{mid} to the victim, if we assume V_{mid} is one half of V_{high}, Eqn. (4.20) can be modified as Eqn. (4.21).

$$\Delta V_{VICTIM} = \frac{C_6}{\frac{(2C_1 + C_3)\cdot C_4}{2C_1 + C_3 + C_4} + C_6} \Delta(V_{AGGRESSOR}/2)\frac{1}{\alpha + 1} \quad (4.21)$$

where

$$\alpha = \frac{\tau_{aggressor}}{\tau_{victim}} = \frac{R_{aggressor}}{R_{victim}}$$

Assuming $R_{aggressor}$ and R_{victim} are the same, Eqn. (4.21) is modified to Eqn. (4.22).

$$\Delta V_{VICTIM} = \frac{C_6}{2(\frac{(2C_1 + C_3)\cdot C_4}{2C_1 + C_3 + C_4} + C_6)} \Delta(V_{AGGRESSOR}/2) \quad (4.22)$$

Based on Eqn. (4.22), the victim's voltage deviation by the aggressor depends on the capacitance C_1, C_3 and C_6. Since C_3 is at least three times bigger than the other capacitances, the worst case victim's voltage deviation by the aggressor will be around $0.064V_{aggressor}$. Hence, the crossbar memory is quite immune to the interconnect noise even if it has higher cell density than the competing memories.

4.8. Area Overhead

Since the crossbar memory with sense amplifiers attached to all the bit lines uses large area, we added MUX logics to reduce the number of sense amplifiers. Below we will show that proper choice of the MUX size can lead to memory with high performance in terms of reading time, power dissipation and area overhead. Before simulating the whole 1 Kb crossbar memory design with this modification, a simplified memory circuit is discussed for analytical understanding.

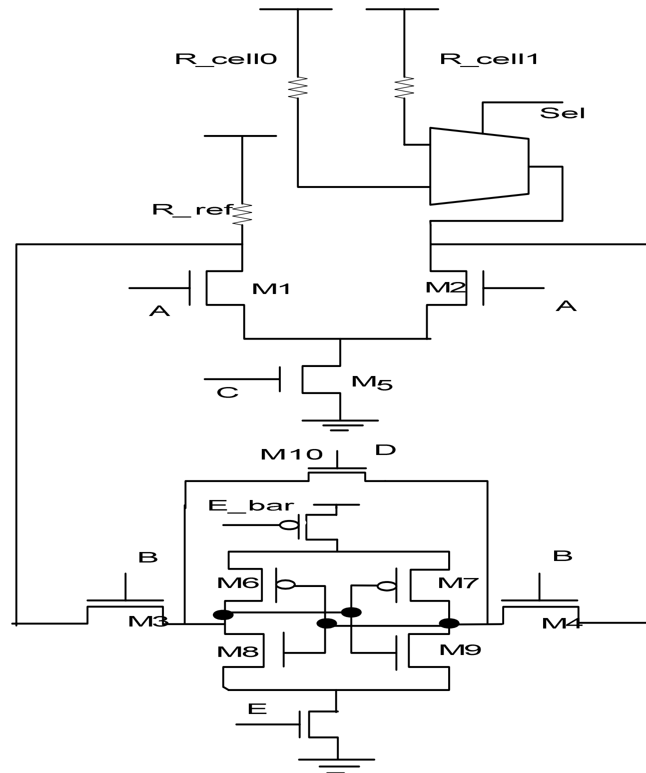


Figure 4.29: A simplified sense amplifier part to investigate the feasibility of usage of MUX logics to reduce the numbers of the sense amplifiers attached to all the bit lines.

Figure 4.29 represents a simplified sense amplifier part of a 1 Kb crossbar memory design with sense amplifiers before decoders. In Figure 4.29, the crossbar memory cell is modeled with a resistor for this simulation. The MUX is used to select one of the two cells which are represented as resistors. Figure 4.29 uses 2 by 1 MUX so that the number of the sense amplifiers can be reduced by half.

Simulation results in Figure 4.30 are transient results of the circuit in Figure 4.29. The first graph of Figure 4.30 is the output signal of Figure 4.29. As shown in Figure 4.30, the output is changed based on selection of the MUX signal in the third graph of Figure 4.30.

The second graph of Figure 4.30 is a control signal for the equalization.

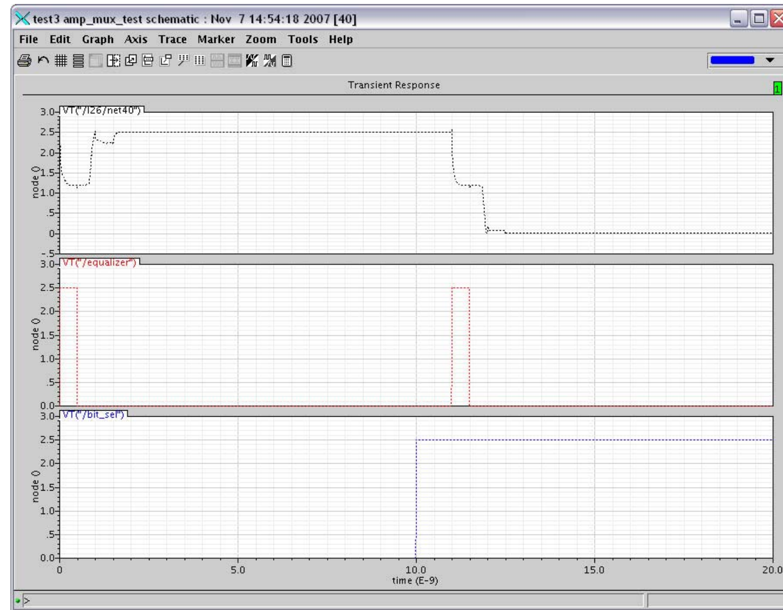


Figure 4.30: Simulation results of the simplified sense amplifier with MUX logics.

Figure 4.31 is a schematic of a 1 Kb crossbar memory design using 2 by 1 MUX logics for reducing the number of sense amplifiers by half. Principles of the circuit in Figure 4.29 are applied to a 1 Kb crossbar memory circuit. In Figure 4.31, we read the former 16 bits and change the selection bit of the MUX logics, and then the latter 16 bits are read sequentially. Also, the column decoder is reduced by one half and the decoder inputs are reduced from 10 to 8. There should be a routing overhead to connect the MUX logics, the array, and the column decoder. The reading time is increased compared with the memory in Figure 4.20 because it requires the additional time to select the MUX logics and initialization time for the sense amplifiers.

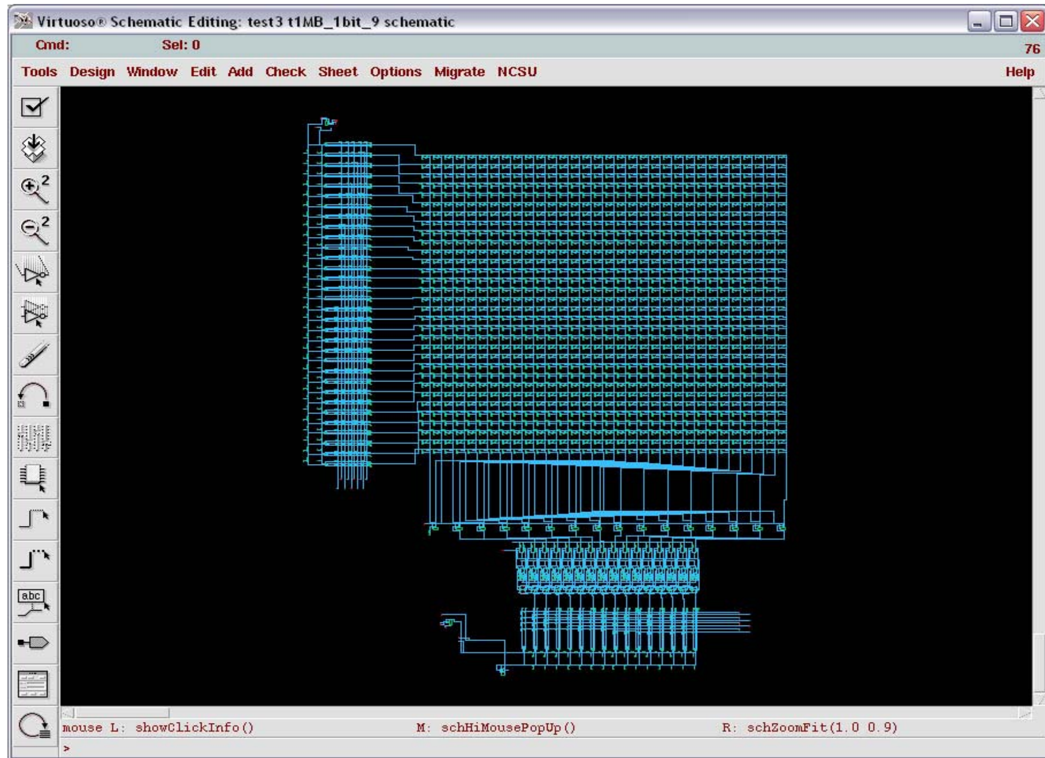


Figure 4.31: Schematic of the 1 Kb crossbar memory design with using 2 by 1 MUX logics to reduce the numbers of sense amplifiers by one half.

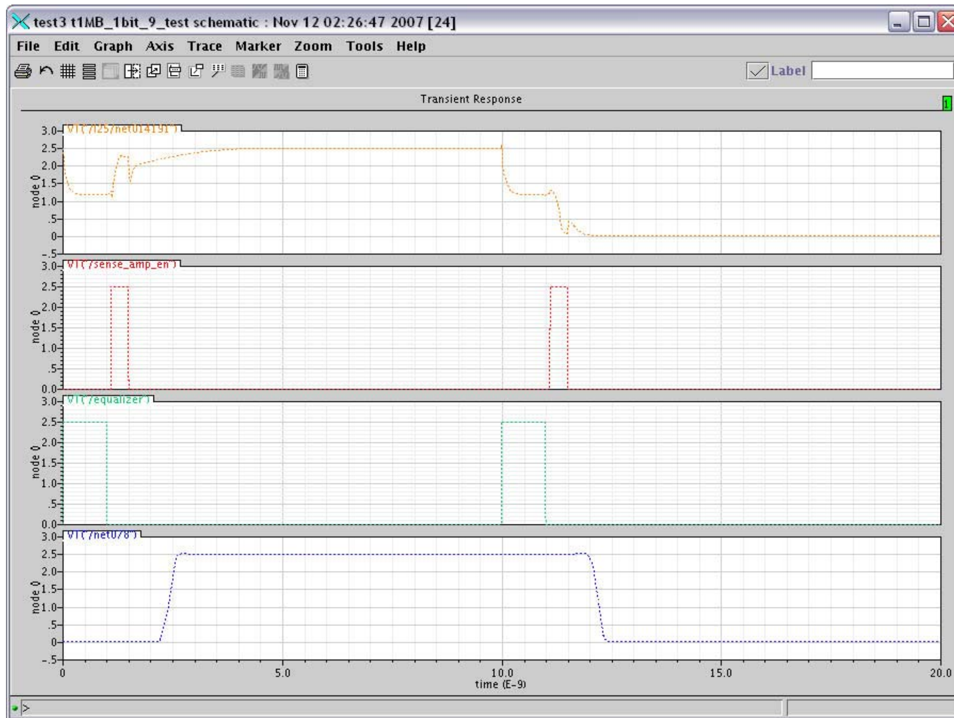


Figure 4.32: Simulation results on the 1 Kb crossbar memory design with 2 by 1 MUX logics to reduce the numbers of sense amplifiers.

Figure 4.32 represents the simulation results on the crossbar memory shown in Figure 4.31. The simulation results in Figure 4.32 are the transient results of the circuit in Figure 4.31. The first graph of Figure 4.32 is the output signal of Figure 4.31. As shown in Figure 4.32, the output is changed based on the selection of the MUX signal in the fourth graph of Figure 4.32. The third graph of Figure 4.32 is a control signal for the equalization, and the second graph of Figure 4.32 is a control signal for the latch inside of the sense amplifier.

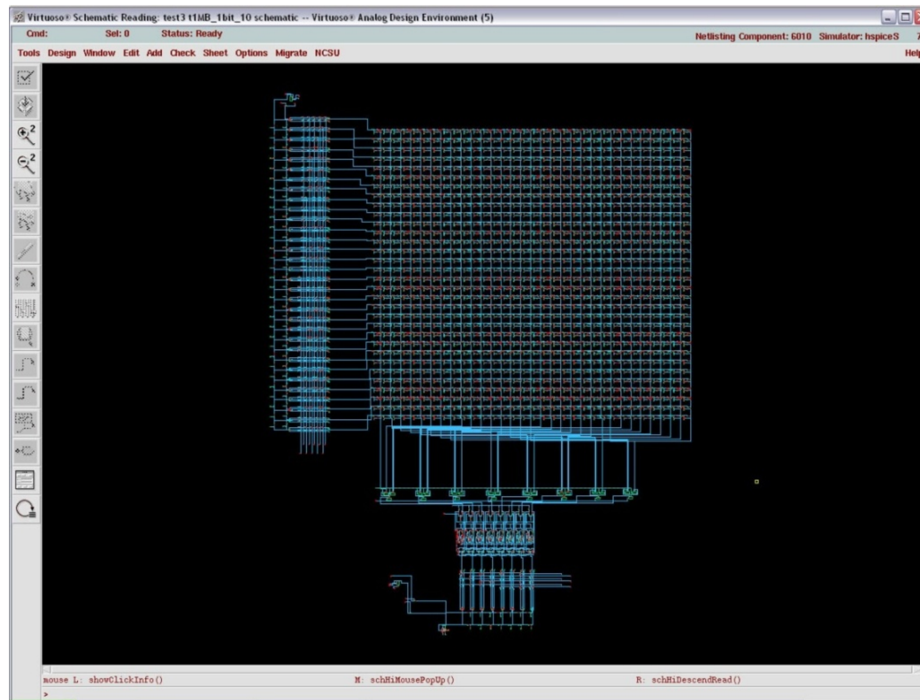


Figure 4.33: Schematic of the 1 Kb crossbar memory design with using 4 by 1 MUX logics to reduce the numbers of sense amplifiers by one quarter.

Figure 4.33 is a schematic of the 1 Kb crossbar memory design using 4 by 1 MUX logics for reducing the number of sense amplifiers to one quarter. In Figure 4.33, we read the former 8 bits and change the selection bit of the MUX logics. The latter 8 bits are read sequentially until 32 bits are read. Also, the column decoder is reduced to one quarter and the decoder inputs are reduced from 10 to 6. The routing overhead to connect the MUX logics, the array, and the column decoder is increased more than that of the memory design in Figure 4.31. The reading time is increased compared with the memory in Figure 4.31 because it requires a longer time to select the MUX logics and a longer initialization time for the sense amplifiers.

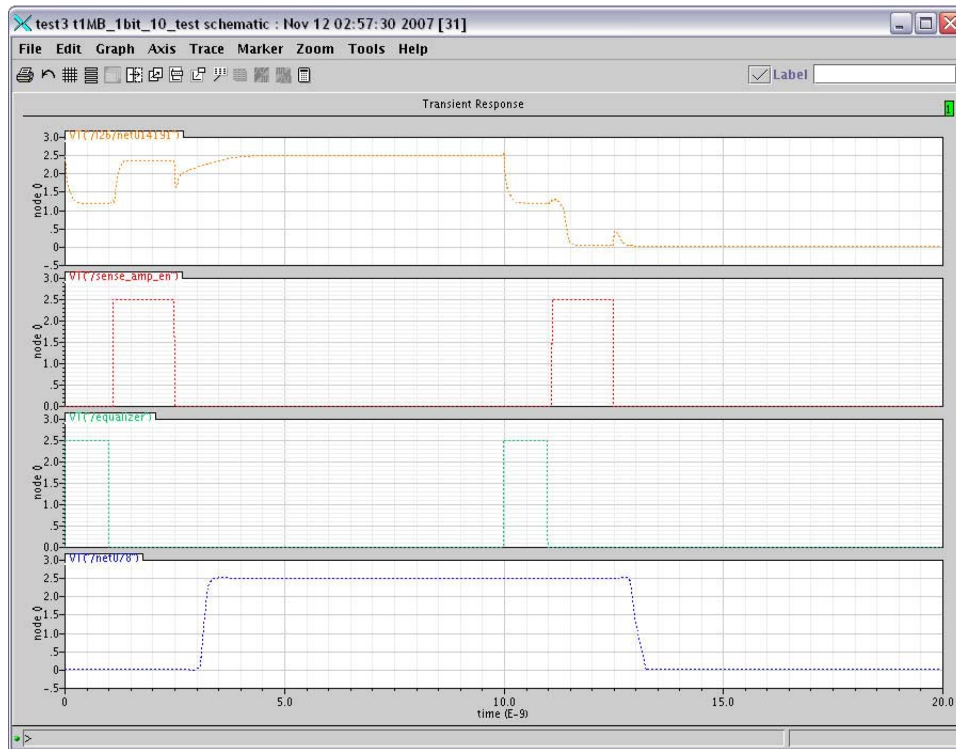


Figure 4.34: Simulation results on the 1 Kb crossbar memory design with 4 by 1 MUX logics to reduce the numbers of sense amplifiers.

Figure 4.34 represents the simulation results on the crossbar memory shown in Figure 4.33. The simulation results in Figure 4.34 are the transient results of the circuit in Figure 4.33. The first graph of Figure 4.34 is the output signal of Figure 4.33. As shown in Figure 4.34, the output is changed based on the selection of the MUX signal in the fourth graph of Figure 4.34. The third graph of Figure 4.34 is a control signal for the equalization, and the second graph of Figure 4.34 is a control signal for the latch inside the sense amplifier. As shown in Figure 4.34, the crossbar memory using 4 by 1 MUX

logics requires longer time to initialize and enable the latch inside the sense amplifier.

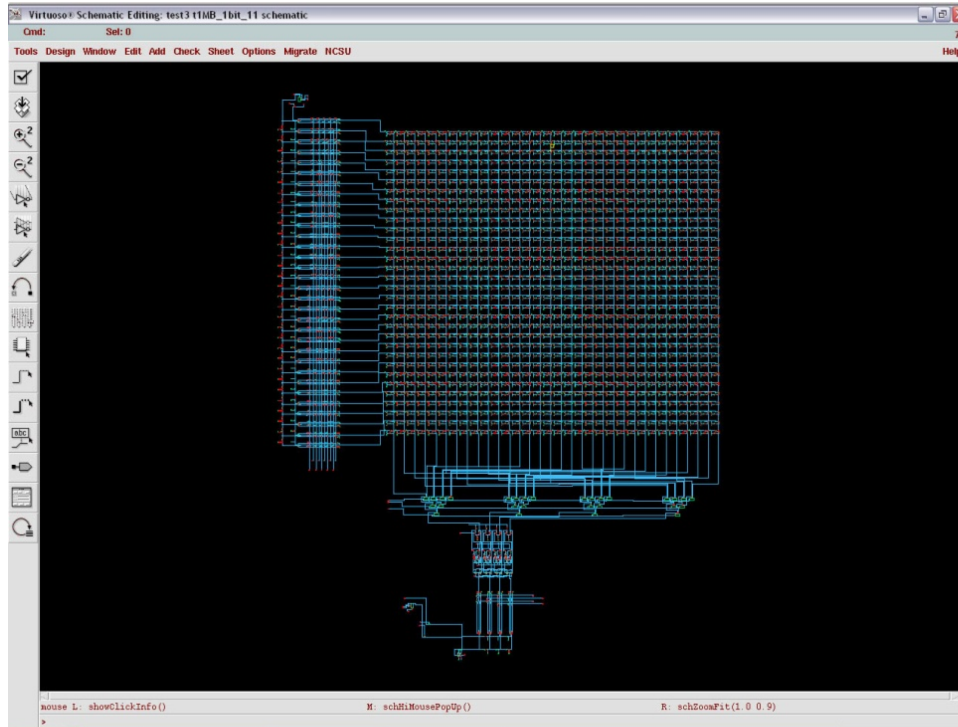


Figure 4.35: Schematic of the 1Kb crossbar memory design with using 8 by 1 MUX logics to reduce the numbers of sense amplifiers by one quarter.

Figure 4.35 is a schematic of the 1 Kb crossbar memory design using 8 by 1 MUX logics for reducing the number of sense amplifiers to one eighth. In Figure 4.35, we read the former 4 bits and change the selection bit of the MUX logics, and then the latter 4 bits are read sequentially until 32 bits are read. Also, the column decoder is reduced to one eighth, and the decoder inputs are reduced from 10 to 4. The routing overhead to connect the MUX logics, the array, and the column decoder is increased more than that of the

memory in Figure 4.33. The reading time is increased compared to the memory in Figure 4.33 because it requires a longer time to select the MUX logics and a longer initialization time for the sense amplifiers.

Figure 4.36 represents the simulation results on the crossbar memory shown in Figure 4.35. The simulation results in Figure 4.36 are the transient results of the circuit in Figure 4.35. The first graph of Figure 4.36 is the output signal of Figure 4.35. As shown in Figure 4.36, the output is changed based on the selection of the MUX signal in the fourth graph of Figure 4.36. The third graph of Figure 4.36 is a control signal for the equalization, and the second graph of Figure 4.36 is a control signal for the latch inside the sense amplifier. As shown in Figure 4.36, the crossbar memory using 8 by 1 MUX logics requires a longer time to initialize and enable the latch inside the sense amplifier than does the crossbar memory using 4 by 1 MUX logics.

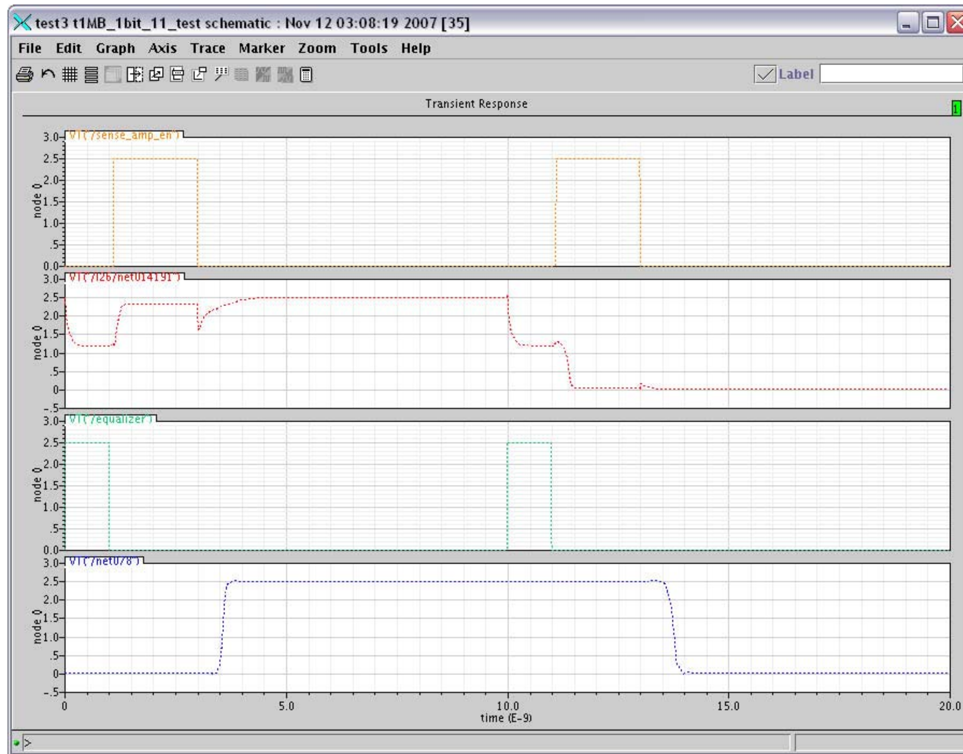


Figure 4.36: Simulation results on the 1Kb crossbar memory design with 8 by 1 MUX logics to reduce the numbers of sense amplifiers.

Table 4.5 shows the area comparison among the 1Kb crossbar memory design using various sizes of MUX logics. As shown in Table 4.5, the best memory design for area overhead is the crossbar memory using 8 by 1 MUX logics. The larger size of MUX will reduce the pitch width of the 1 Kb crossbar memory design. However, it will also increase the area overhead of routing. If the size of MUX is above 8 by 1, the pitch width of the 1 Kb crossbar memory design will not be further decreased since the width of sense amplifiers will become less than the width of the array. Therefore, the 1 Kb crossbar memory design using 8 by 1 MUX logics occupies the smallest area with 67 percent of the area being reduced. However, this ratio between cell area to peripheral

circuit area is proper to be used as memory chips.

Figure 4.37 represents the simulation results with the changes of the MUX selection bits and addresses on the 1Kb crossbar memory design using 8 by 1 MUX logics. The first graph of Figure 4.37 depicts the first selection bit of the MUX logics and the second graph of Figure 4.37 shows the second selection bit of the 8 by 1 MUX logics. The third graph of Figure 4.37 represents output signal of the 1Kb crossbar memory design. Also, the fourth graph and last graph of Figure 4.37 demonstrate the first decoder bit and the control signal for enabling the latch inside the sense amplifiers, respectively.

Finally, Table 4.6 shows the area comparison among the 1 Mb crossbar memory designs using various sizes of MUX logic. As shown in Table 4.5, the area overhead is decreased as the size of MUX is increased. We achieve reduction in the area overhead since the area portion of the array in the memory is increased as the size of the memory is enlarged. In other words, the area portion of the peripheral circuitry is decreased so that the area overhead of MUX logic is not critical in the overall area overhead. Also, the ratio between cell area and peripheral circuit area is appropriate for 1 Mb crossbar memory circuit with MUX logic.

Table 4.5: Area comparison among the 1Kb crossbar memory designs with various sizes of MUX.

Unit: μm^2	No mux	2 by 1 mux	4 by 1 mux	8 by 1 mux	16 by 1 mux
Array	$32^2\times0.2\times4.8$	$32^2\times0.2\times2.4$	$32^2\times0.2\times1.2$	$32^2\times0.2\times0.6$	$32^2\times0.2\times0.3$
Decoders	(32×1×0.2)+(32×1×4.8)	(32×1×0.2)+(32×0.8×2.4)	(32×1×0.2)+(32×0.6×1.2)	(32×1×0.2)+(32×0.4×0.6)	(32×1×0.2)+(32×0.2×0.3)
SA	32×4.8×10	32×10×2.4	32×10×1.2	32×10×0.6	32×10×0.3
Mux	0	32×10×2.4	32×25×1.2	32×61×0.6	32×153×0.3
Global Routing	0	32×1×2.4	32×2×1.2	32×4×0.6	32×8×0.3
Total	2679.04	2172.16	1696	1557.76	1682.56
Cell / peripheral	0.58	0.29	0.17	0.08	0.03

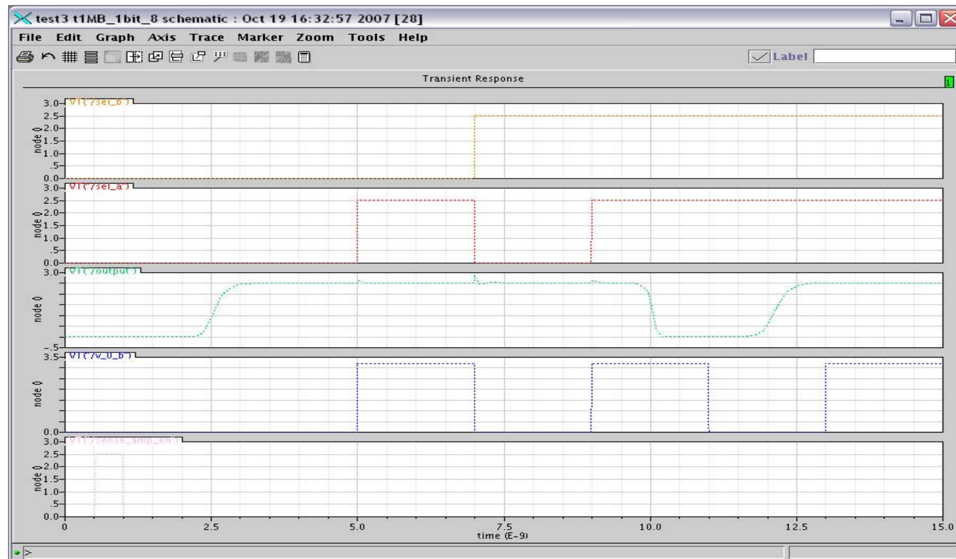


Figure 4.37: Simulation results with the changes of the MUX selection bits and addresses on the 1Kb crossbar memory design with 8 by 1 MUX logics.

Table 4.6: Area comparison among the 1Mb crossbar memory designs with various sizes of MUX.

Unit: μm^2	No mux	2 by 1 mux	4 by 1 mux	8 by 1 mux	16 by 1 mux
Array	$1K^2\times0.2\times4.8$	$1K^2\times0.2\times2.4$	$1K^2\times0.2\times1.2$	$1K^2\times0.2\times0.6$	$1K^2\times0.2\times0.4$
Decoders	(1K×2×0.2)+(1K×2×4.8)	(1K×2×0.2)+(1K×1.6×2.4)	(1K×2×0.2)+(1K×1.2×1.2)	(1K×2×0.2)+(1K×0.8×0.6)	(1K×2×0.2)+(1K×0.4×0.4)
SA	1K×4.8×10	1K×10×2.4	1K×10×1.2	1K×10×0.6	1K×10×0.4
Mux	0	1K×10×2.4	1K×25×1.2	1K×63×0.6	1K×200×0.4
Global Routing	0	1K×1×2.4	1K×2×1.2	1K×4×0.6	1K×8×0.6
Total	1017600	534640	286240	167080	169360
Cell / peripheral	16.7	8.78	5.2	2.6	0.9

Table 4.7: Read time comparison among the 1Mb crossbar memory designs with various sizes of MUX.

	No mux	2 by 1 mux	4 by 1 mux	8 by 1 mux	16 by 1 mux
Read time	~ 6.4 ns	~ 6.6 ns	~ 6.8 ns	~ 7.2 ns	~ 8.0 ns

Table 4.7 shows the estimated read time of 1 Mb crossbar memory designs by extrapolating the read time of 1 Kb crossbar memory designs. The read time difference between the worst case and best case is around 1.6 ns. Compared with area deviation with various sizes of MUX, the read time deviation is not so large. Therefore the 1 Mb crossbar memory design with 8 by 1 MUX is the best choice compromising the area overhead and fast read time.

Since we used a 0.25 μm process library for the peripheral circuit design, it is worthwhile noting how the total area is changed as process libraries change. If we extrapolate the peripheral circuit area based on the gate length ratio between the process libraries, we observe that the total area is minimized when we use 16 by 1 MUX logics in Table 4.8. Based on this result, the total area can be changed depending on the process libraries and the optimal MUX logic is also changed as the process library is changed.

Table 4.8: Comparison of total area of 1 Mb crossbar memory designs according to CMOS process technology.

Unit: μm^2	No mux	2 by 1 mux	4 by 1 mux	8 by 1 mux	16 by 1 mux
250 nm	1060905	534640	286240	165880	113800
130 nm	1020201	516761.6	263260.2	147404.8	90134.8
45 nm	1008442	505109.2	253205.2	128705.9	69526.19

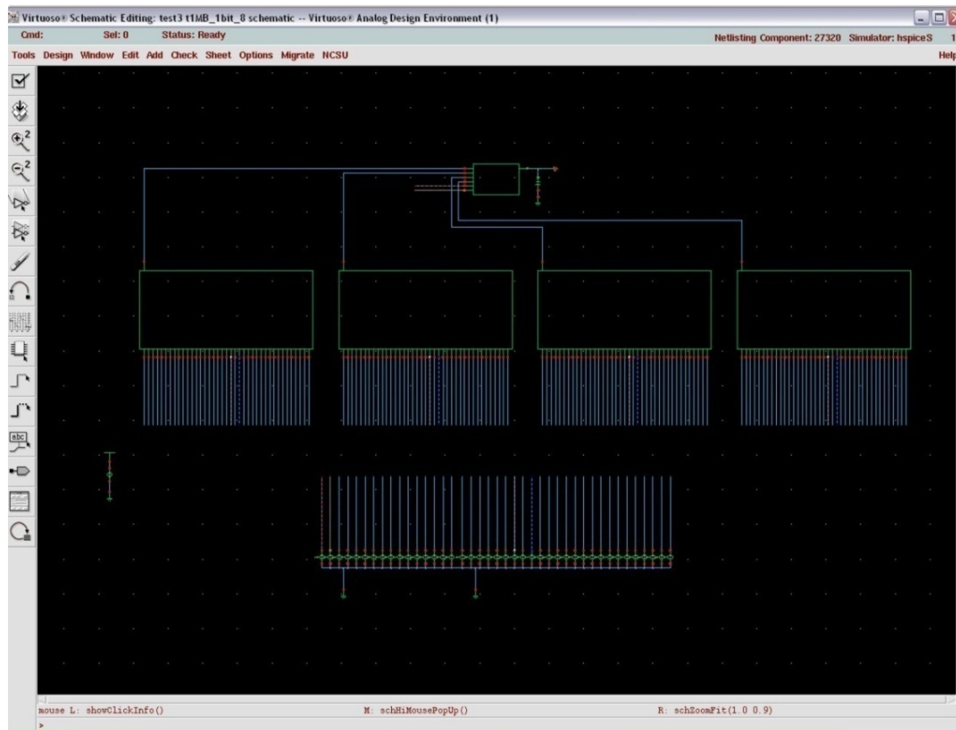


Figure 4.38: Schematic view of 4 Kb crossbar memories with 4 banks.

4.9. Bank Based System Design

From the completed 1 Kb crossbar memory design demonstrated in the previous sections, we show how we can design much larger systems. We suggest a large system design

technology using bank based system design. A complete 4 Kb crossbar memory system has been simulated due to the working memory space limitation for simulation, however this technique can also be applied to make a large array such as 1 Gb or even more with the same design technique. Figure 4.38 shows the 4 Kb crossbar memory implemented with 4 banks. Each bank is selected by the 4 by 1 MUX on the top. Since bank based design removes the huge delay to access a large array, this technique is very essential for a large memory design.

The reading time comparison is demonstrated between the 1 Kb crossbar memory design and the 4 Kb crossbar memory design with 4 banks, where each bank consists of a 1 Kb crossbar memory design. The 4 Kb crossbar memory can be implemented with four 1 Kb memory designs using a 4 by 1 MUX to select each 1 Kb memory design which is called a bank. As shown in Table 4.9, the reading time per bit is not changed significantly with the addition of the 4 by 1 MUX. This is because the reading time of 32x4 bits is much longer than the propagation delay time of the 4 by 1 MUX.

Using the bank based system design technique for a large memory system design, we need basic memory cells, decoders, bank selection address, column and row addresses, and global data bus and global amplifier and driver for input and output and control circuit which is usually implemented in CPU. For a large memory system design with crossbar memory technique, we provide all the core components which consist of the complete memory system in this chapter.

Table 4.9: Read time comparison between 1 Kb and 4 Kb with 4 banks.

	1 Kb	4 Kb (4 banks)
Read	32 ns + 4 ns	4+32+3(0.5+4+32) ns
Read per bit	1.1 ns / bit	1.14 ns / bit

Assuming we use a bank design system with 1ns of global routing delay and three stage hierarchies to make 1Gb crossbar memories with a cell size of 200 nm by 200 nm, the read time and area overhead with an unit array size are described in Table 4.10. The read time is increased as the size of the unit array is increased due to the increased time to

access the target crossbar cell for read or write through decoders. In contrast, the delay due to the MUX is decreased as the size of the unit array is increased because of the reduced size of the MUX. The delay due to the MUX is not comparable to the delay due to the increased decoder size and the array size so that the overall delay is increased as the size of the unit array is increased. The area overhead is decreased as the unit array size is increased since the area overhead due to the peripheral circuitry is decreased. Therefore, the optimal unit array size can be obtained by finding the minimum product of the read time and the area overhead. The estimated optimal unit array size to make 1Gb memories is 1 Mb as shown in Table 4.10.

Since the peripheral CMOS circuit takes around 800 ps to read or write in a 1 Kb crossbar memory, the read time can be extrapolated for a 1Mb crossbar memory design. As the time for crossbar memory cell access excluding CMOS circuit will be around 200 ps, the read time can be around 7.2 ns. Assuming we use a bank design system with 1ns of global routing delay and four stage hierarchies with 8 by 1 MUXes which depends on the process library, read time for a 4 Gb crossbar memory can be calculated as shown in Table 4.11. Also, read time for a 32 Gb crossbar memory can be calculated in the same manner as shown in Table 4.11. From the area data from Table 4.8, we estimate how many cells can be inserted in 1 x 1 cm^2 assuming that we use 16 by 1 MUX logics as shown in Table 4.12.

For power dissipation, since we only access one word each read or write operation, the total power dissipation is only increased by MUXes for 1 Gb compared with 1 Mb. Considering the power measurement for 1 Kb, we estimate the power dissipation for 1Gb is around 7 W. However, this power dissipation depends on the decoder designs and the design structure. We can lower the power dissipation to less than 1 W by sizing the SA and changing the decoder design at the expense of read or write time. This will be the future work of this chapter.

However, it is problematic to incorporate library cells to the crossbar memory design as crossbar memory metal line width is narrower than the feature size for the 250 nm process library. This problem can be solved as the feature size is decreased. With cutting

edge technology, the 45 nm process, we can incorporate our crossbar memory cell with the library cell. This situation will become even better as the feature size is decreased in the next technology generation.

Table 4.10: Read time and area overhead comparison for the optimal size of a unit array to make 1 Gb crossbar memories with a cell size of 200 nm x 200 nm.

	256 Kb	1 Mb	4 Mb	16 Mb
Read time	~ 7.4 ns	~ 9 ns	~ 12.5 ns	~ 17 ns
Area	~ 1.1 cm^2	~ 0.9 cm^2	~ 0.7 cm^2	0.6 cm^2
Read time x Area	8.14 $ns \cdot cm^2$	8.1 $ns \cdot cm^2$	8.75 $ns \cdot cm^2$	10.2 $ns \cdot cm^2$

Table 4.11: Estimated read time for 4 Gb and 32 Gb crossbar memory designs.

	4 Gb	32 Gb
250 nm	~ 9 ns	~10 ns
130 nm	~ 6.6 ns	~7.6 ns
45 nm	~5 ns	~6 ns

Table 4.12: Estimation of memory capacity to be realized in a 1 cm x 1 cm die.

Cell size	200 nm x 200 nm	100 nm x 100 nm	10 nm x 10 nm
2-D technology	~1.3 G	~4 G	~350 G
3-D technology	~2.6 G	~8 G	~700 G

4.10. Technology Comparison

Table 4.13: Technology comparison between the amorphous crossbar memories and the conventional memory or other candidate memory for the future [4.8]-[4.11].

Category	*Parameter*	*SRAM (130nm)*	*DRAM (130nm)*	*NOR Flash (130nm)*	*MRAM (180nm)*	*Crossbar*
Cost	**Cell area**	**0.16 μm2**	**0.14 μm2**	**0.19 μm2**	**0.7-1.4 μm2**	**0.005um2 – 0.01um2**
	Cost/Mb	**~80 cent**	**~20 cent**	**~3 cent**	**6 $ [4.31]**	**~ 1 cent**
	Process cost adder	**0%**	**25%**	**25%**	**25%**	**< 5%**
Performance	**Read access**	**5~10 ns**	**10~20 ns**	**80 ns**	**5–20 ns**	**5-10ns**
	Write cycle	**3.4 ns**	**20 ns**	**1 us**	**5–20 ns**	**5 -10ns**
Power	**Data retention**	**0.6 nA per bit at 85°C**	**0.2 nA per bit at 85°C**	**0**	**0**	**0**
Miscellaneous	**Data retention**	**Volatile**	**Volatile**	**Nonvolatile**	**Nonvolatile**	**Nonvolatile**

Finally, the technology parameters of the amorphous crossbar memory are compared with the conventional memories and other types of crossbar memory. In the cell area perspective, the amorphous crossbar memory demonstrates better size than other memories. Even if the conventional memories are scaled down to 45 nm in 2008, the cell area of the amorphous crossbar memory is still less than the conventional memories. MRAM has been scaled down to 90nm with toggle mode, but it has half-select problem which is caused by the induced field overlaps between adjacent cells in the small area. This problem is the barrier for MRAM scaling even if an advanced technique such as spin-torque-transfer is researched. MRAM has another problem in the cost as described in the introduction. It requires fundamental changes in the fabrication facilities resulting in a barrier to the investors. In terms of writing and reading times, the amorphous silicon represents similar performance compared with CMOS-based or other emerging memory technologies. The power dissipation for data retention of the amorphous crossbar memory

is zero as the other nonvolatile memories. PRAM (not shown in table 4.13) may provide comparable cell area and writing and reading performance. However, the most difficult problem of PRAM is that it requires high programming density. Additionally, it is vulnerable to temperature in the fabrication process and the user operation. This affects the performance of the device severely [4.8]-[4.11].

4.11. Conclusion

Crossbar memory system demonstrates better overall performance than conventional memory systems, such as flash memory system, DRAM, SRAM, MRAM, and etc.. For measuring the performance, we electrically modeled the crossbar memory cell and designed specific peripheral circuitry, for instance, the column and row decoders, and we suggested a sensing circuitry for detecting the difference in the resistance of the cell and the control circuit for reading and writing. In implementing the sensing circuitry, two methods, attaching the sense amplifiers before decoders and at the final stage are investigated. The first approach, attaching the sense amplifiers before decoders, shows better performance in speed. The other approach demonstrates better performance in area overhead and power dissipation. To suggest better design for the power dissipation, an analytical modeling on static power dissipation which is challenging issue as the device size scales down was conducted. To improve the area overhead with the first approach, we introduced MUX logics to reduce the numbers of sense amplifiers which are attached to all the bit lines. We achieved 67 percent of reduction in the area overhead through this approach for a 1 Kb crossbar memory design. For a 1 Mb crossbar memory, 400 percent of reduction in the area overhead is achieved because the portion of the array in the memory is increased as the size of the memory is enlarged. Looking into the future, the crossbar memory structure can be used to form reconfigurable architectures for hybrid

digital circuits. There have already been proposals combining the crossbar device with CMOS to achieve high density field-programmable gate arrays to perform general logic functions [4.30]. High yield memory chips can also be realized from faulty crossbars with incorporated error correction techniques [4.29] and we believe with further studies at the circuit level and device development it is very feasible to tera-bit storage potential with the hybrid crossbar/CMOS approach.

CHAPTER V

Summary and Future Work

5.1. Summary

In this thesis, various approaches to see the possibility of the applications of nanotechnology to conventional circuit systems have been attempted. In the application to the image processors, a new architecture to process color images using an array of multi-peak resonant tunneling diodes is studied and various image functions pertaining to color image processing are implemented. The proposed architecture has been demonstrated through HSPICE simulation to be faster in processing speed than that of a single core DSP and lower power dissipation.

To enhance the performance of conventional velocity-tuned filters, a nanoelectronic architecture using CNNs and an array of RTDs was suggested for a velocity tuned filter. We demonstrate the superiority of our velocity-tuned filter to the conventional velocity-tuned filter, which is based on analog circuitry using circuit level simulations. The simulation shows that the new velocity-tuned filter is 10, 100 and 1000 times better than the counterpart in area, power consumption, and speed, respectively. Through analytical modeling, we confirmed the filter is asymptotically stable only when the resonant tunneling diode is the positive differential resistance region.

To suggest a new memory system, a crossbar memory system based on amorphous Silicon material is implemented in this thesis. In the design of the crossbar memory system, new device modeling, new decoders, control logic for handling supply voltage selection, and new sense amplifiers are used. Through transistor level simulation, we

demonstrate that our crossbar memory system is overall better than the conventional flash memory system.

5.2. Future Work

RTDs combined with CNNs for image processors can be further studied for high density of RTDs in a given area and fast processing time. Since the RTDs are the two terminal devices with external current source leading to power dissipation, the size of RTDs should be minimized to reduce the power consumption. The reduced size of RTDs also decreases the processing time because of the reduced capacitance of RTDs.

To increase the density of RTDs and the processing speed, quantum dot based image processors can be further studied in the future. For this implementation, an advanced process technology should be combined into the processor design since the process deviation of size of quantum dots is critical to the RTD characteristics. Also, interconnection technology between quantum dots should be developed to incorporate more functions using template methods in the future.

In the color image processing work, we can further study to find other types of transistors to replace the CMOS or HEMT to realize the templates into hardware. Also, 3-D quantum dot modeling can be incorporated with the color image processing and the velocity tuned filter.

In the crossbar memory work, new types of decoder designs can be further investigated to minimize the power dissipation maintaining the low area overhead and access time. The crossbar memory structure can be used to form reconfigurable architectures for hybrid digital circuits. There have already been proposals combining the crossbar device with CMOS to achieve high density field-programmable gate arrays to perform general logic functions [4.30] and high yield memory chips with incorporated error correction techniques [4.29]. We can further study at the circuit level and device development for tera-bit storage potential with the hybrid crossbar/CMOS approach.

BIBLIOGRAPHY

[1.1] V. P. Roychowdhury, D. B. Janes, and S. Bandyopadhyay, “Collective Computational Activity in Self-Assembled Arrays of Quantum Dots: A Novel Neuromorphic Architecture for Nanoelectronics,” *IEEE Trans. on Electron Devices,* vol. 43, no.10, 1996, pp. 1688-1699.

[1.2] V. P. Roychowdhury, D. B. Janes, and S. Bandyopadhyay, “Nanoelectronic Architecture for Boolean Logic,” *Proceedings* of *IEEE,* vol. 85, no. 4, 1997, pp. 574-587.

[1.3] L. O. Chua and L.Yang, “Cellular neural networks: Theory,” *IEEE Trans. Circuits and Systems*, vol. 35, October 1988, pp. 1257-1272.

[1.4] T. Roska and A. Vazquez, “Toward visual microprocessors,” *Proceedings of the IEEE*, vol. 90, no. 7, July 2002, pp. 1244-1257.

[1.5] E.R. Brown, “Hot Carriers in Semiconductor Nanostructures,” Physics and Application, Jagdeep Shah, ed., Academic Press, 1992.

[1.6] R.T. Bate, G. A. Frazier, W. R. Fresley, J. W. Lee, and M. A. Reed, *Proc. SPIE*, vol. 792, 1987; R. T. Bate, *Solid State Technol.*, vol. 32, pp.101, 1989, pp. 26-35.

[1.7] P. Balshi, D. Broido, and K. Kempa, *J.Appl. Phys.*, vol. 70, 1991, pp. 5150.

[1.8] A. N. Korotkov, “Wireless single-electron logic biased by alternating electric field,” *Appl. Phys. Lett.*, 67, 1995, pp. 2412.

[1.9] A. C. Seabaugh, Y. C. Kao and H. T. Yuan, “Nine-state resonant tunneling diode memory,” *IEEE Electron Device Lett.*, vol. 13, no. 9, 1992, pp. 479-481.

[1.10] Sing-Rong Li, P. Mazumder, L. O. Chua, “Cellular Neural/Nonlinear networks using resonant tunneling diode,” *IEEE conference on nanotechnology*, 2004, pp. 164-167.

[1.11] E.R. Brown, “Hot Carriers in Semiconductor Nanostructures,” Physics and Application, Jagdeep Shah, ed., Academic Press, 1992.

[1.12] M. A. Kastner, "The Single-electron Transistor," *Rev. Mod. Phys.* 64, 1992, pp. 849-858.

[1.13] D. B. Strukov and K. K. Likharev, "A reconfigurable Architecture for Hybrid CMOS/Nanodevice circuits," *Proceedings of the 2006 ACM/SIGDA 14th international Symposium on FPGA*, 2006, pp. 131-140.

[1.14] R. J. Luyken and F. Hofmann, "Concepts for hybrid CMOS-molecular non-volatile memories," *Nanotechnology*, 2003, pp. 273-276.

[1.15] Ming-Huei Shieh, Hung Chang Lin, "A multiple-dimensional multiple-state SRAM cell using resonant tunneling diodes," *IEEE Journal of Solid State Circuits*, vol. 29, no. 5, 1994, pp. 623-630.

[1.16] K. Karahaliloglu, S. Balkir, S, "Image processing with quantum dot nanostructures," *International Symposium on Circuits and Systems*, vol. 5, 2002, pp. 217-220.

[1.17] K. Runge, *et. al.*, "Silicon Bipolar Integrated Circuits for Multi-Gb/second Optical Communication Systems," *IEEE Trans. on communications*, vol. 9, 1991, pp. 636-644.

[1.18] Tsai, I.-M., Kuo, S.-Y., "Quantum Boolean circuit construction and layout under locality constraint," *IEEE conference on nanotechnology*, 2001, pp. 111-116.

[1.19] E. H. Adelson, "Layered representation for image coding," Technical Report 181, Vision and Modeling Group, The MIT Media Lab, Dec. 1991.

[1.20] N. Franceschini, "Early processig of color and motion in a mosaic visual system," *Neuro-science research*, suppl. 2, 1985, pp.17-49.

[1.21] A. B. Torralba, J. Herault, "An efficient neuromorphic analog network for motion estimation" *IEEE Trans. on Circuits and Systems*, vol. 46, no 2, 1999, pp. 269-280.

[1.22] Yang *et. al.*, "Unequal packet loss resilience for MPEG-4 video over the Internet," *ISCAS*, vol. 2, 2003, pp. 832-835.

[1.23] A. Khitun, K. L. Wang, "Cellular nonlinear network based on semiconductor tunneling nanostructure," *IEEE Trans. on Electronic Devices*, vol. 52, no 2, 2005, pp. 183-189.

[1.24] B. E. Shi, "Gabor-type filtering in space and time with cellular neural networks," *IEEE Trans. Circuits Syst. I,* vol. 45, Feb. 1998, pp. 121–132.

[1.25] B. E. Shi, T. Roska, and L. O. Chua, “Design of linear cellular neural networks for motion sensitive filtering,” *IEEE Trans. Circuits Syst. II,* vol. 40, May 1993.

[1.26] E. H. Adelson, and J. R. Bergen, “Spatiotemporal energy models for the perception of motion,” *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 2, pp. 284–299, Feb. 1985, pp. 320–331.

[1.27] J. Barron, D. S. Fleet, S. S. Beauchemin, and T. A. Burkitt, “Performance of optical flow techniques,” in *Proc. IEEE CVPR*, Champaign, IL, 1992, pp. 236–242.

[1.28] K. L. Shepard, “Design methodologies for noise in digital integrated circuits,” *Proc. DAC*, 1998, pp. 94–99.

[1.29] Shaofeng Yu, “Extremely scaled planar bulk CMOS: challenges and options,” *ICSICT*, vol. 1, 2004, pp. 41-46.

[1.30] Yong Chen, Gun-Young Jung, Douglas A A Ohlberg, Xuema Li, Duncan R Stewart, Jan O Jeppesen, Kent A Nielsen, J Fraser Stoddart and R StanleyWilliams, “Nanoscale molecular-switch crossbar circuits,” *Nanotechnology*, 14, 2003, pp. 462-468.

[1.31] Ziegler, M. M. and Stan, M. R. “CMOS/nano co-design for crossbar-based molecular electronic systems,” IEEE Transactions on *Nanotechnology*, vol.2, 2003, pp. 217-230.

[1.32] Myung-Hyun Lee, Young Kwan Kim and Yoon-Hwa Choi, “A defect-tolerant memory architecture for molecular electronics,” *IEEE Transactions on Nanotechnology*, vol. 3, 2004, pp. 152-157.

[1.33] Yi Luo, C. Patrick Collier, Jan O. Jeppesen, Kent A. Nielsen, Erica DeIonno, Greg Ho, Julie Perkins, Hsian-Rong Tseng, Tohru Yamamoto, J. Fraser Stoddart and James R. Heath, “Two-Dimensional Molecular Electronics Circuits,” *ChemPhysChem*, vol. 3, 2002, pp. 519-595.

[1.34] D. B. Strukov and K. K. Likharev, “CMOL FPGA: a reconfigurable architecture for hybrid digital circuits with two-terminal nanodevices,” *Nanotechnology*, vol. 16, 2005, pp. 888-900.

[1.35] D. B. Strukov and K. K. Likharev, “Prospects for terabit-scale nanoelectronic memories,” *Nanotechnology*, vol. 16, 2005, pp. 137-148.

[1.36] A. E. Owen, P. G. Le Comber, W. E. Spear and J. Hajto, “Memory switching in amorphous silicon devices,” *J. Non-Cryst. Solids* 59 & 60, 1983, pp. 1273-1280.

[1.37] M. Jafar and D. Haneman, “Switching in amorphous-silicon devices,” *Phys. Rev. B*, 49, 1994, pp. 13611-13615.

[1.38] P. G. Lecomber et. al., “The switching mechanism in amorphous silicon junctions,” *J. of Non-Cry. Solids 77 & 78*, 1985, pp. 1373-1382.

[1.39] Jerzy Kanicki Ed., “Amorphous & microcrystalline semiconductor devices,” Artech House, Boston, 1992.

[1.40] Sung Hyun Jo and Wei Lu, “Nanovolatile Resistive Switching Devices based on Nanoscale Metal/Amorphous Silicon/Crystalline Silicon Juctions,” *Material Research Society*, 2007.

[1.41] Sung Hyun Jo and Wei Lu, “CMOS Compatible Nanoscale Nonvolatile Resistive Switching Memory”, *Nano letters*, Vol. 8, No. 2, 2008, pp. 392-397.

[1.42] Stephen Y. Chou, Peter R. Krauss and Preston J. Renstrom, “Imprint Lithography with 25-Nanometer Resolution,” *Science 5*, vol. 272, 1996, pp. 85-87.

[1.43] L.E. Brus, Chemistry and Physics of Semiconductor Nanocrystals, 2007.

[1.44] Leon O. Chua and Tomas Roska, Cellular Neural Networks and Visual Computing, Cambridge University Press, 2002.

[1.45] William J. Gallagher and Stuart S. P. Parkin, "Development of the magnetic tunnel junction MRAM at IBM: From first junctions to a 16-Mb MRAM demonstrator chip", IBM, 24 January 2006.

[1.46] Rabaey, J., et al., *Digital Integrated Circuits: A Design Perspective*, 2nd. ed., Prentice Hall Publishers, Inc., 2003.

[1.47] Andre DeHon, “Array-Based Architecture for FET-Based, Nanoscale Electronics”, *IEEE Transactions on Nanotechnology*, vol. 2, 2003, pp. 23-32.

[1.48] Andre DeHon, Seth Copen Goldstein, Philip J. Kuekes and Patric Lincoln, “Nonphotolithographic Nanoscale Memory Density Prospects”, IEEE Transactions on Nanotechnology, vol. 4, 2005, pp. 215-228.

[1.49] Gregory S Snider and R Stanley Williams, “Nano/CMOS architectures using a field-programmable nanowire interconnect”, *Nanotechnology* 18, 2007.

[1.50] G. Snider, “Computing with hysteretic resistor crossbars”, *Applied Physics A*, 80, 2005, pp. 1165-1172.

[1.51] J. M. Cruz and L. O. Chua, “A 16 x 16 Cellular Neural Network Universal Chip: The First Complete Single-Chip Dynamic Computer Array with Distributed Memory”, *Analog Integrated Circuits and Signal Processing*, 15, 1998, pp. 227-237.

[1.52] Intel's 45nm CMOS Technology, Intel Technology Journal, Volume 12 Issue 02 Published June 17, 2008.

[1.53] Woo Hyung Lee and Pinaki Mazumder, “Motion detection by quantum dots based velocity tuned filter,” *IEEE Trans. on Nanotechnology*, vol. 7, 2008.

[1.54] Woo Hyung Lee and Pinaki Mazumder, “Color image processing using multi-peak RTD,” *IEEE conference on Nanotechnology*, 2007.

[1.55] Woo Hyung Lee and Pinaki Mazumder, “New velocity tuned filter using nanoelectronic architecture,” *IEEE conference on Nanotechnology*, 2006.

[1.56] Woo Hyung Lee and Pinaki Mazumder, “New logic circuits consisting of quantum dots and CMOS*”, European conference on circuit theory and design*, Cork, Ireland, 2005.

[2.1] L. O. Chua and L. Yang, “Cellular Neural Networks: Applications,” *IEEE Trans. Circuits and Systems*, vol. 35, Oct. 1988, pp. 1273-1290.

[2.2] J. P. Sun, G. I. Haddad, P. Mazumder and J. N. Schulman, “Resonant Tunneling Diodes: Models and Properties”, *Proceedings of the IEEE*, 1998.

[2.3] L. L. Chang, L. Esaki, and R. Tsu, “Resonant Tunneling Diode in Semiconductor Double Barriers,” *Appl. Phys. Lett.*, vol. 24, Jun. 1974, pp. 593-595.

[2.4] A. Seabaugh and P. Mazumder, “Quanmtum Devices and Their Applications,” *Proceedings of the IEEE*, vol. 7, no. 4, April 1999.

[2.5] G. I. Haddad and P. Mazumder, “Tunneling Devices and Their Applications in High-Functionality/Speed Digital Circuits,” *Journal of Solid State Electronics*, vol. 41, no. 10, Oct. 1997, pp. 1515-1524.

[2.6] K. Karahaliloglu, S. Balkir, S. Pramanik, and S. Bandyopadhyay, “A Quantum Dot Image Processor,” *IEEE Trans. on Electron Devices*, vol. 50, no. 7, July 2003.

[2.7] A. C. Seabaugh, Y. C. Kao and H. T. Yuan, “Nine-state Resonant Tunneling Diode Memory,” *IEEE Electron Device Lett.*, vol. 13, no. 9, 1992, pp. 479-481.

[2.8] http://www.cdrinfo.com/Sections/News/Details.aspx?NewsId=23573

[2.9] http://users.erols.com/chare/video.htm

[2.10] E. Khaliullin et al., "Two-photon excited luminescence from metal nanoparticles: Discrete spectrum and quantum-size effect," APS, 2002.

[2.11] Texas Instruments, TMS320C6711 DSK: http://focus.ti.com/docs/toolsw/folders/print/tmds320006711.html

[2.12] Texas Instruments, TMS320DM642 Product folder: http://focus.ti.com/docs/toolsw/folders/print/tmds320dm642.html

[2.13] Texas Instruments, TMS320DM642 Power Consumption: http://focus.ti.com/lit/an/spra962a/spra962a.pdf

[2.14] Texas Instruments, TMS320C6416 Product folder: http://focus.ti.com/docs/prod/folders/print/tms320c6416.html

[2.15] Texas Instruments, TMS320C6416 Power Consumption: http://focus.ti.com/lit/an/spra811c/spra811c.pdf

[2.16] http://focus.ti.com/dsp/docs/dsphome.tsp?sectionId=46&DCMP=TIHeaderTracking&HQS=Other+OT+hdr_p_dsp

[2.17] W. Pratt, Digital Image Processing, 3rd Edition, Wiley 2001.

[3.1] M Egmont-Petersen, D de Ridder, H Handels, "Image processing with neural networks—a review", *Pattern Recognition*, 2002, pp. 2279-2301.

[3.2] P. Kinget and M. S. Steyaert, "A Programmable Analog Cellular Neural Network CMOS Chip for High Speed Image Processing", *IEEE Journal of Solid-State Circuits*, vol. 30, 1995, pp. 235-243.

[3.3] E. H. Adelson, *Layered representation for image coding*. Technical Report 181, Vision and Modeling Group, The MIT Media Lab, 1991.

[3.4] N. Franceschini, "Early processig of color and motion in a mosaic visual system," *Neuro-science research*, vol. 2, 1985, pp. 17–49.

[3.5] A. B. Torralba and J. Herault, "An efficient neuromorphic analog network for motion estimation," *IEEE Transactions on Circuits and Systems*, vol. 46, no. 2, pp. 269–280, 1999.

[3.6] L. Esaki and R. Tsu, "Superlattice and negative differenctial conductivity in semiconductors," *IBM J. Res. Develop.*, vol. 14, no. 1, 1970, pp. 61–65.

[3.7] L. L. Chang, L. Esaki, and R. Tsu, “Resonant tunneling in semiconductor double barriers,” *Appl. Phys. Lett.*, vol. 24, no. 12, 1974, pp. 593–595.

[3.8] P. Mazumder, S. Kulkarni, M. Bhattacharya, J. P. Sun, and G. I. Haddad, “Digital circuit applications of resonant tunneling device,” *Proc. IEEE*, vol. 86, no. 4, 1998, pp. 664–686.

[3.9] J. N. Schulman, H. J. D. L. Santos, and D. H. Chow, “Physics based rtd current-voltage equation,” *IEEE Electron Device Lett.*, vol. 17, 1996, pp. 220–222.

[3.10] T. P. E. Broekaert *et al.*, “A monolithic 4-bit 2-gsps resonant tunneling analog-to-digital converter,” *IEEE J. Solid-State Circuits*, vol. 33, 1998, pp. 1342–1349.

[3.11] J. P. Sun, G. I. Haddad, P. Mazumder, and J. N. Schulman, “Resonant tunneling diodes: models and properties,” *Proc. IEEE*, vol. 86, 1998, pp. 641– 660.

[3.12] R. D. Kenney, M. J. Schulte, and M. A. Erle, “A high-frequency decimal multiplier,” *IEEE International Conference on Computer Design*, 2004, pp. 26–29.

[3.13] Yang *et al.*, “Unequal packet loss resilience for mpeg-4 video over the internet,” *ISCAS*, vol. 2, 2000, pp. 832–835.

[3.14] A. Seabaugh, Z. Hu, Q. Liu, D. Rink, and J. Wang, “Silicon-based tunneling diodes integrated circuits,” *Proc. Int. Workshop Quantum Functional Devices*, 2000, pp. 5–8.

[3.15] Y. Su, J. Chang, Y. Lu, C. Lin, K. Wu and Z. Wu, “Novel AlInAsSb/InGaAs Double-Barrier Resonant Tunneling Diode with High Peak-to-Valley Current Ratio at Room Temperature,” *IEEE Electron Device Letters*, vol. 21, 2000, pp. 146-148.

[4.1] P. J. Kuekes and R. S. Williams and J. R. Heath, “Molecular Wire Crossbar Memory”, US Patent 6,128,214, 2000.

[4.2] M. R. Stan, F. D. Franzon, S. C. Goldstein and J. C. Lach and M. M. Ziegler, “Molecular Electronics: From Devices and Interconnect to Circuits and Architecture,” *Proceedings of the IEEE*, vol. 91, 2003.

[4.3] P. J. Kuekes and R. S. Williams and J. R. Heath, “Demultiplexer for a Molecular Wire Crossbar Network”, US Patent 6,256,767, 2001.

[4.4] P. J. Kuekes and R. S. Williams and J. R. Heath, “Molecular-wire Crossbar Interconnect for Signal Routing and Communications”, US Patent 6,314,019, 2001.

[4.5] T. Hogg and G. Snider, “Defect-tolerant Logic with Nanoscale Crossbar Circuits”, *Journal of Electronic Testing*, 2007.

[4.6] R J Luyken and F Hofmann, “Concepts for hybrid CMOS-molecular non-volatile memories,” *Nanotechnology*, 14, 2003, pp. 273-276.

[4.7] A. DeHon, P. Lincoln and J. E. Savage, “Stochastic assembly of sublithographic nanoscale interfaces,” *IEEE Transactions on Nanotechnology*, vol. 2, 2003, pp. 165-174.

[4.8] M. Durlam et. al., “A low power 1Mbit MRAM based on 1T1MTJ bit cell integrated with copper interconnect” Motorolar presentation 2002.

[4.9] Saied Tehrani, “Toggle MRAM performance, reliability, and scalability” Freescale presentation 2005.

[4.10] http://www.research.ibm.com/journal/rd/501/gallagher.html.

[4.11] H. Horii et al., *2003 Symposium on VLSI Technology*, pp. 177-178.

[4.12] Y. Chen, G. Y. Jung, D. A. A. Ohlberg, X. M. Li, D. R. Stewart, J.O. Jeppesen, K. A. Nielsen, J. F. Stoddart, and R. S. Williams, “Nanoscale molecular-switch crossbar circuits,” *Nanotechnology*. 14, 462, 2003.

[4.13] Y. Cheng and C. Hu, MOSFET Modeling and BSIM3 User’s Guide, Boston: Kluwer Academic Publishers, 1999.

[4.14] H. Frizsche, “Physics of instabilities in amorphous semiconductors”, *IBM J. Res. Develop.*, vol. 13, Sept. 1969.

[4.15] R. G. Neale and John A. Aseltine, “The Application of Amorphous Materials to Computer Memories”, *IEEE Transactions on Electron Devices*, vol. ED-20, 1973, pp. 195-205.

[4.16] P. G. Le Comber, W. E. Spear and A. Ghaith, “Amorphous-Silicon Field-Effect Device and Possible Application”, *Electronics Letters*, vol. 15, 1979, pp. 179-181.

[4.17] W. Den Boer, *Applied Physics Letters*, 1982, 40, pp. 812.

[4.18] P. G. Le Comber, A. E. Owen, W. E. Spear, J. Hajto, A. J. Snell, W. K. Choi, M. J. Rose, S. Reynolds, *Journal of Non-Crystal Solids*, 1985, 1373, pp. 77-78.

[4.19] A. E. Owen, P. G. Le Comber, J. Hajto, A. J. Snell, *International Journal of Electron*, 1992, 73, pp. 897.

[4.20] M. Jafar, D. Hanemann, *Physical Reiview B*, 1994, 49, 13611.

[4.21] A. Avila, R. Asomoza, *Solid-State Electron*, 2000, 44, 17.

[4.22] J. Hu, H. M. Branz, R. S. Crandall, S. Ward, Q. Wang, *Thin Solid Films*, 2003, 430, pp. 249.

[4.23] Sung Hyun Jo and Wei Lu, unpublished.

[4.24] Y. Shacham, “Filament Formation and the Final Resistance Modeling in Amorphous-Silicon Vertical Programmable Element,” *IEEE Transactions on Electron Devices*, vol. 40, 1993, pp. 1780-1788.

[4.25] R. D. Gould, “The effects of filamentary conduction through uniform and flawed filaments and insulators and semiconductors,” *J. Non-Crystalline Solids*, vol. 55, 1983, pp. 363.

[4.26] W. Lu and C. Lieber, “Nanoelectronics from the bottom up,” *Nature Materials*, 6, 2007, pp. 841-850.

[4.27] Hu, J., Branz, H. M., Crandall, R. S., Ward, S. & Wang, Q., “Switching and filament formation in hot-wire CVD p-type a-Si:H devices,” *Thin Solid Films* **430**, 2003, pp. 249-252.

[4.28] Jafar, M. & Haneman, D., “Switching in amorphous-silicon devices,” *Phys. Rev. B*, 49, 1994, pp. 13611-13615.

[4.29] D. B. Strukov and K. Likharev, “Prospects for terabit-scale nanoelectronic memories,” *Nanotechnology*, 16, 2005, pp. 137-148.

[4.30] D. B. Strukov and K. Likharev, “CMOL FPGA: a reconfigurable architecture for hybrid digital circuits with two-terminal nanodevices,” *Nanotechnology*, 16, 2005, pp. 888-900.

[4.31] http://www.dso.com/indepth/showArticle.jhtml?articleID=190303175.