PROGRAMMING EXPLORATION OF MEMRISTOR CROSSBAR

by

Xiaocong Du

B.S. in Automation, Shandong University, Jinan, China, 2014

Submitted to the Graduate Faculty of

the Swanson School of Engineering in partial fulfillment

of the requirements for the degree of

Master of Science

University of Pittsburgh

2016

UNIVERSITY OF PITTSBURGH

SWANSON SCHOOL OF ENGINEERING

This thesis was presented

by

Xiaocong Du

It was defended on

March 29, 2016

and approved by

Hai Li, Ph.D., Associate Professor, Department of Electrical and Computer Engineering

Yiran Chen, Ph.D., Associate Professor, Department of Electrical and Computer Engineering

Zhihong Mao, Ph.D., Associate Professor,

Department of Electrical and Computer Engineering

Thesis Advisor:

Hai Li, Ph.D., Associate Professor, Department of Electrical and Computer Engineering

PROGRAMMING EXPLORATION OF MEMRISTOR CROSSBAR

Xiaocong Du, M.S.

University of Pittsburgh, 2016

Memristor crossbar is prevailing as one of the most promising candidates to construct the neural network because of their similarity to biological synapses, favorable programmability, simple structure and high performance regarding area efficiency and power consumption. However, the performance of the memristor crossbar is limited by unideal programming and sensing process.

In this thesis, the most preferred cell structure which is known as “one-transistor-one-memristor” is investigated. Different factors that may have impacts on programming, such as the structure, the parameters and the conductance of a crossbar cell are studied using both theoretical analysis and simulation.

Based on previous analysis, the programming process of the memristor crossbar deserves a deep exploration. For programming, the primary objective is to find out the relationship between the programmability of the memristor crossbar and its characteristics, such as the IR-drop and the crossbar size. The results are expected to be useful references for researchers designing the memristor crossbar.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGMENT

At the beginning of my thesis, I would like to thank all those people who made my thesis possible and unforgettable.

First of all, I would like to thank my advisor, Dr. Hai Li of the Swanson School of Engineering at the University of Pittsburgh. The door to Prof. Li's office is always open whenever I run into a trouble spot or had a question about my research. She gave me the precious opportunity to join in Evolutionary Intelligence lab which had enormous significance to me. I would also thank my co-advisor in EI-Lab, Dr. Yiran Chen, for his support, assistance and valuable guidance.

Also, I would extend my sincere thanks to Dr. Zhihong Mao, for his charming personality and professional dedication. He teaches me a lot, from the aspects of both academy and life.

I am also indebted to my friends, Zheng Li, Bonan Yan, Qing Yang, Jianlei Yang, Linghao Song, Chaofei Yang, Yuxu Wu and Jiachuan Chen. Without their passionate assistance, the thesis would not have finished.

Finally, I would give my profound gratitude to my parents for providing me with continuous support and encouragement throughout my years of study. This accomplishment would not have been possible without them.

1.0 INTRODUCTION

Recent years, machine learning technology are widely applied in pattern recognition, text transcription, artificial intelligence, and relevant areas [1]-[3]. The neural network has become popular in the last ten years as a kind of well-known algorithm in the machine learning system. For example, a neural network can be used in a system of pattern recognition which can determine accurately what an input pattern is or model a relationship between a pattern and real objects [4][5]. However, to implement extensive and efficient software neural networks, processing and storage issues need to be considered [6][7]. On the one hand, occupying plenty of data rows when simulating can consume a large amount of memory in the computer. On the other hand, transmission of signals through tons of connections can consume incredible amounts of processing power and time. Schmidhuber in [8] claim that the success of neural networks, especially in image recognition area, is largely dependent on the improvement of hardware, such as *Graphics Processing Unit* (GPU), which can dramatically decrease the training time from months to days.

However, the conventional neural network based on *Complementary Metal-Oxide-Semiconductor Transistors* (CMOS) is facing with several serious problems, such as scale limitations [9]-[11] and the "memory wall" [12][13], i.e. the degraded efficiency of data transportation between CPU and storage system. To break through these limitations, emerging nanoscale resistive devices such as memristor, is prevailing as one of the most promising candidates to construct the neural network thanks to their similarity to biological synapses,

favorable programmability, simple structure and high performance regarding area efficiency and power consumption [8][14][15]. Using memristors as synapses in neuromorphic circuits can offer both high connectivity and high density required for high-speed computing. The conductance of the memristor can be controlled by changing the charge through it, which is similar to biological synapse [14].

Based on nanoscale devices, memristor crossbar array, consisting of termed memristors, have been identified as a leading candidate for memory and computation in the neuromorphic network [16][17][18].

1.1 MOTIVATION

For the "one-transistor-one-memristor" crossbar array used in the neural network, programming/writing is the first and most significant step in training the network. During the programming operation, the voltage on memristor is always less than the applied, because the transistor, as well as line resistance, acts as a voltage divider. Moreover, the divided voltage is affected by various factors, including crossbar cell structure, transistor size, writing voltage, line resistance, memristance and crossbar array scale. Moreover, those factors will be taken into account while designing a memristor crossbar.

In this work, individual influence of each factor on programming operation will be explored step by step. Based on these results, the design margin of the memristor size will be investigated, which is of a considerable guiding significance for memristor crossbar design.

1.2 ROADMAP

This thesis is organized as follows: Chapter 2 presents the design background knowledge of transistor, memristor and crossbar array; Chapter 3 introduces and analyzes “one-transistor-one-memristor” structures and their characteristics; In chapter 4, the programming performance of crossbar cell in terms of the relationship between IR-drop and resistance ratio is simulated and analyzed; Chapter 5 gives the design margin of the crossbar array scale based on previous results; Chapter 6 presents the overall conclusion and future works.

2.0 DESIGN BASICS

This chapter will give the necessary knowledge of NMOS transistor, memristor, "one-transistor-one-memristor" cell structure and crossbar, which lays a foundation for the later work.

2.1 NMOS TRANSISTOR

The *Metal Oxide Semiconductor Field Effect Transistor* (MOSFET) is a commonly-used type of transistor used for switching electronic signals. The structure and symbol of NMOS transistor (enhancement mode field-effect transistor) is shown in Figure 1. The transistor length and width used in the following chapters refer to the length and width of the conducting channel between the source and drain.

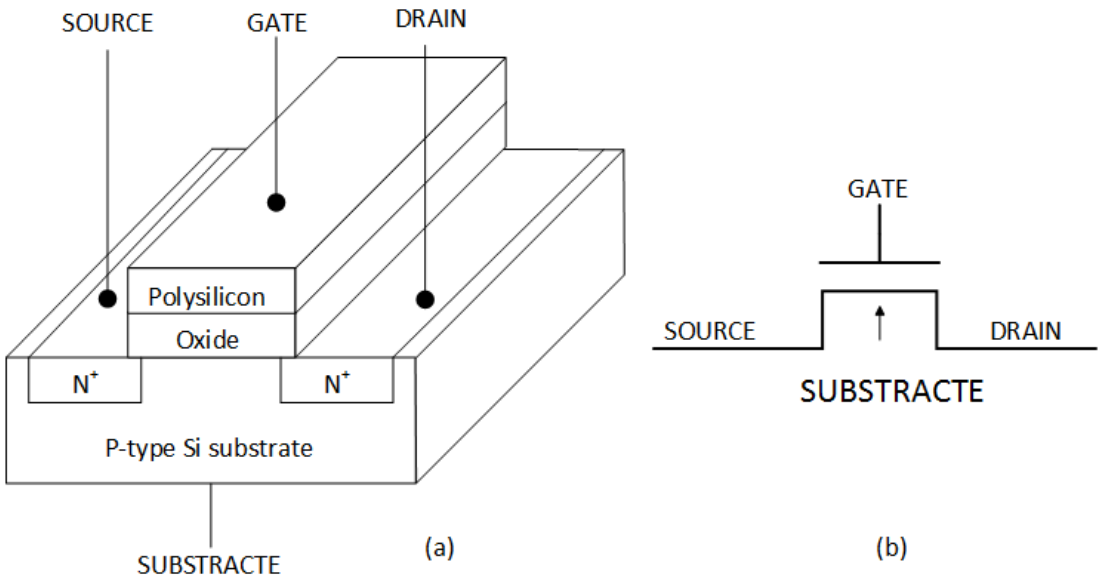

Figure 1. NMOS structure (a) and symbol (b)

Usually, the transistor has three working zones: cut-off, linear and saturation. When the applied voltage V_{gs} (the voltage between gate and source) is lower than V_t (the threshold voltage), the transistor works in the cutoff zone. When $V_{gs} > V_t$ and $V_{ds} < V_{gs} - V_t$, the transistor works in the linear zone. When $V_{ds} > V_{gs} - V_t$, the transistor works in saturation zone. The saturation voltage, V_{dsat}, is defined as $V_{gs} - V_t$.

In this thesis, the transistor I use is "NFET33" from the IBM CMR8SF-RVT library, 130nm technology. By simulation, the relationship between I_{ds} and V_{ds} under different given V_{gs}

is shown in Figure 2. The length here is 400nm and the width 4μm. From the graph, the threshold voltage, saturation voltage and conductance of the NMOS transistor can be estimated.

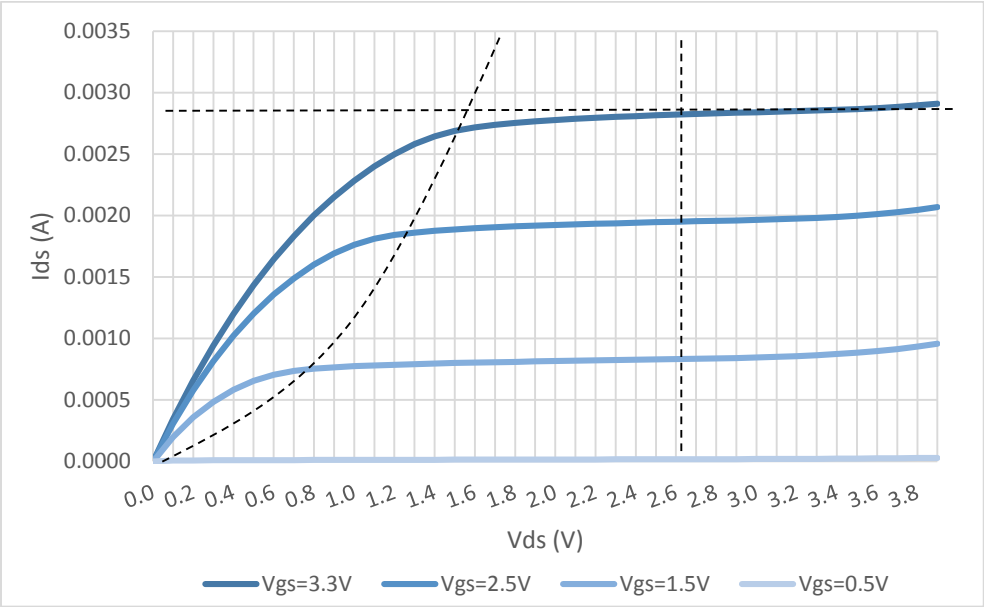

Figure 2. I-V curve of NMOS transistor

When V_{gs} is given to be 0.5V, the transistor is cut off, without current flowing through drain and source. From which, it can be deduced that the threshold voltage is a little higher than 0.5V. In the manual, the threshold voltage is indicated to be 0.58V. Moreover, take V_{gs}=3.3V for example. It can be analyzed that when $V_{ds} = 2.7V$ or so, the transistor steps from linear zone to saturation zone. This voltage will be a reference in the study below.

2.2 MEMRISTOR

The memristor was a nanoscale circuit device that proposed by Leon Chua in 1971. It was originally defined following a non-linear functional relationship between magnetic flux linkage and the amount of electric charge that has flowed and joining the past three classes of the electrical circuit, the resistor, the capacitor and the inductor [8]. Memristance is related to the integral of the input current rather than the instantaneous input value [14].

The definition of the memristor is based on the integrals of current and voltage, which is allowed in the linear time-invariant system. Table 1 shows the comparison of several commonly used circuit elements. The parameter Φ_m refers to the magnetic flux linkage. Thus, the writing voltage and current can change the resistance of memristor when the Φ_m meets the requirements [19]-[21].

Table 1. Comparison of main circuit elements

	Characteristic units	Computation
Resistor	Resistance/Ohm	$R=\frac{V}{I}$
Capacitor	Capacitance/farad	$C=\frac{dQ}{dV}$
Inductor	Inductance/henry	$L=\frac{d\Phi_m}{dI}$
Memristor	Memristance/Ohm	$R_m=\frac{d\Phi_m}{dQ}$

2.3 CROSSBAR

A crossbar is a collection of “something” arranged in a matrix configuration and is often used to connect the system. The “something” can be resistors, switches, transistors plus resistors, and memristors. The memristor crossbar is shown in Figure 3, an N by M crossbar array. It has N inputs and M outputs. Memristor crossbar is nowadays evolving as a promising candidate that can be used in neural network computation.

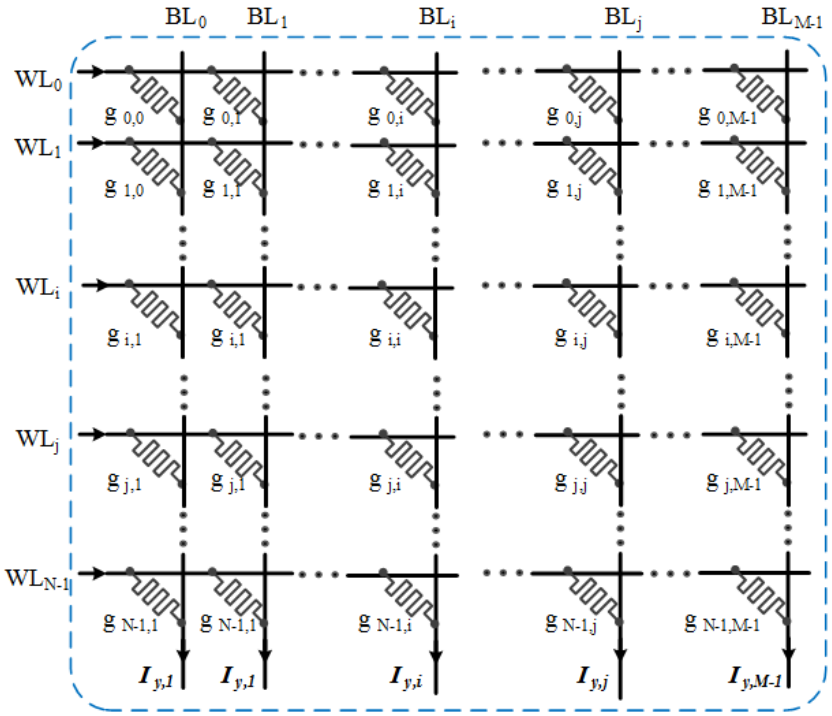

Figure 3. A N×M memristor crossbar array

2.4 “ONE-TRANSISTOR-ONE-MEMRISTOR” STRUCTURE

When building crossbar array, the most famous cell structure is called “one-transistor-one-memristor,” where memristor is connected to an NMOS transistor, as the Figure 4 shows. The word line (WL) is connected with the memristor and the bit line (BL) is connected to the source of NMOS transistor. The selecting line (SL) is connected to the gate of NMOS transistor and responsible for turning on and off the NMOS transistor [16].

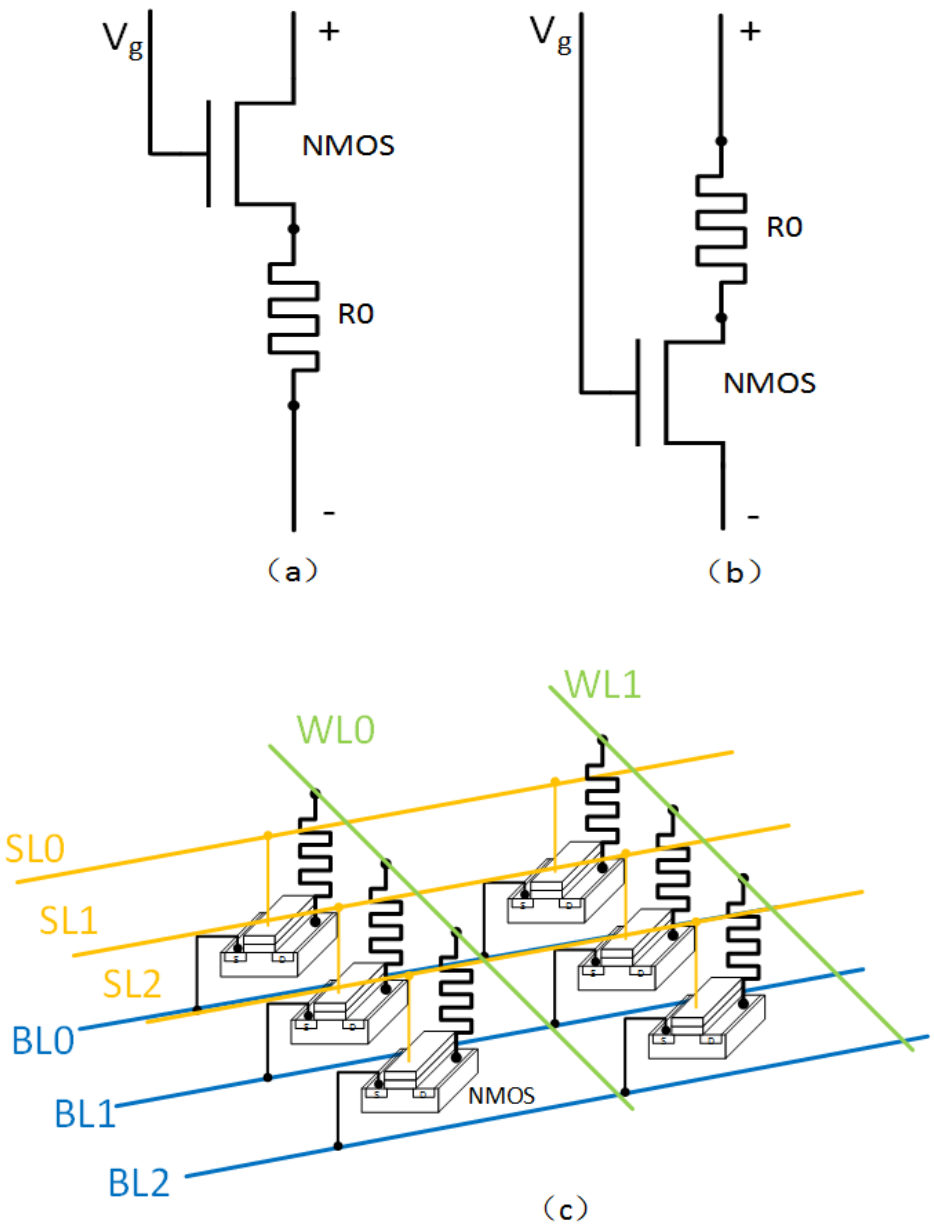

Figure 4. One-transistor-one-memristor symbol (a) (b) and structure (c)

The reason that we use "one-transistor-one-memristor" structure is to avoid sneak path. Figure 5 shows a kind of sneak path issue. When writing the specified memristor, the current flows through the addressed memristor (the green path) but also flows through the neighboring memristor (red path). The sneak path will have a negative impact when reading current since the

current is a sum of this two current. In this case, the transistor can work as a switch to help us determine which path the writing current will flow through.

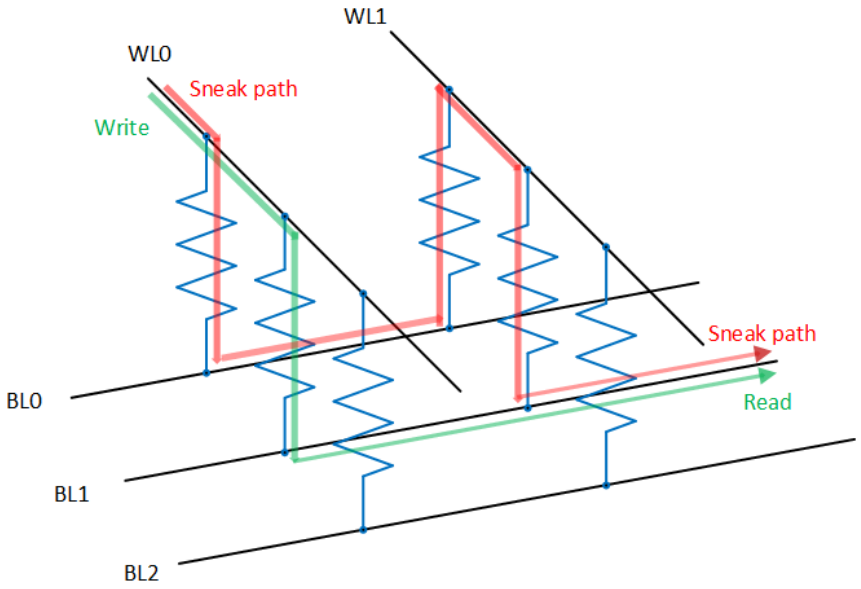

Figure 5. Sneak path issue

3.0 PROPERTIES EXPLORATION OF ONE-TRANSISTOR-ONE-MEMRISTOR STRUCTURE

In this section, the "one-transistor-one-memristor" structure will be studied, from the perspective of structures and parameters. Furthermore, relations between cell performance regarding programmability and parameters were explored based on theoretical analysis and simulation verification.

3.1 STRUCTURE MODELS COMPARISION

For "one-transistor-one-memristor" array, every cell consists of one transistor and one resistor. Also, we should have a resistor on the bit line to help measure the current going through bit line. In this section, we will take one by one crossbar array as an example to explore different structures and their performance. The cell contains voltage source, Word line (WL), Bit line (BL), cell resistance (R_0), cell transistor (T_0), line resistance (R_{line}) and ground (GND).

The first kind of structure shows as Figure 6.

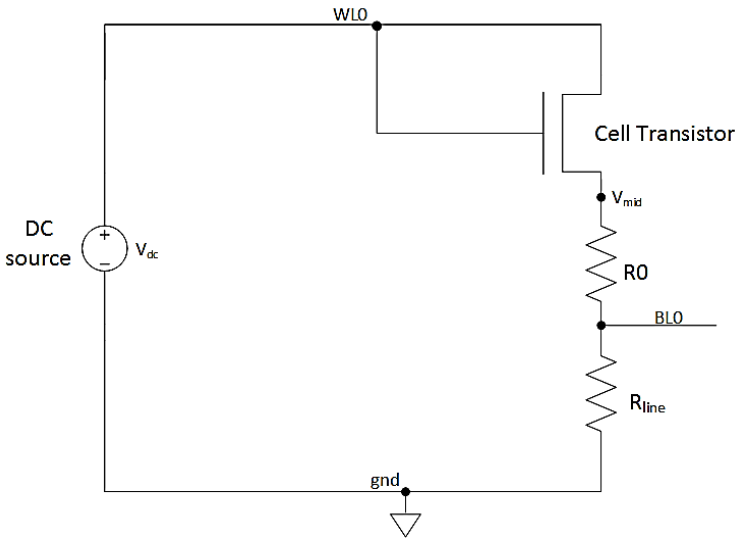

Figure 6. First structure model of 1T1R array

In this type of structure, the gate and drain of the transistor are both connected to the direct current (DC) source. Therefore, V_{gs}=V_{ds}. Here V_{gs} refers to the voltage difference between gate and source, and similarly, V_{ds} means voltage difference between drain and source.

The second structure shows as Figure 7.

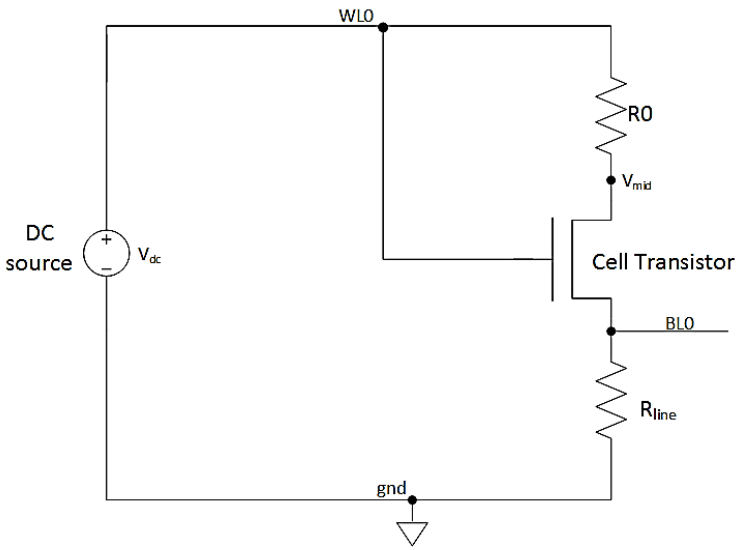

Figure 7. Second structure model of 1T1R array

In this kind of structure, the gate port is connected to DC source while the drain port is connected to the resistor. In this case, $V_{gs}>V_{ds}$.

Now it is a critical crossbar design factor to find out which structure works better as a crossbar array. As discussed before, the function of the transistor is to act as a switch for current. As a result, the voltage that transistor consumes should be as low as possible. Correspondingly, simulations are conducted below to test which model above has better performance.

In Cadence design environment on the LINUX platform, a simulation model is built to test which type mentioned above performs better. The first step is to decide the parameters of the cell transistor, cell resistance, sensing resistance and the DC voltage value.

3.1.1 Components claim

The library used in this work is 130nm CMR8SF-RVT Process while the default length of NMOS (cell nfet33 in the cmrf8sf library) transistor is 400nm. The default width of the NMOS transistor is 500nm. Here, a definition, “feature size” will be used in helping decide the transistor size. The feature size of a transistor is defined as the minimum length of the MOS transistor channel between the drain and the source [22][23]. In this library, the feature size can be 400nm.

In this work, the transistor width will be confined to ten times the feature size to save chip area. So, the upper limit of the transistor width is 4000nm. Here, 4000nm is set as the transistor width.

As for the cell resistance, the value of R_0 is chosen to be 100KOhm, with the width of resistor equals 200nm and the length equals 50μm. This resistance value is large enough to weaken the influence that transistor and sensing resistance bring to the array cell.

The line resistance should be as smaller to reduce the voltage drop in this circuit. By simulation, I find that a 10um length metal layout generates 10Ohm line resistance in the post-simulation. Here, R_{line} is chosen to be 150Ohm, with the width equals 3μm and the length equals 1μm, to mimic the real line resistance.

Simulation is the next step to help decide which model has better performance.

3.1.2 Model comparison

In the Cadence Virtuoso, DC sweep is used to do the simulation. Here DC source varies from 0V to 4V and the voltage of the bit line (V_{BL0}) and the midpoint (the wire connecting transistor and resistance, V_{mid}) are recorded. From the schematic, it can be observed:

For the first structure:

$$V_{T0}=V_{WL0}-V_{mid}$$

$$V_{R0}=V_{mid}-V_{BL0}$$

For the second structure:

$$V_{T0}= V_{mid}-V_{BL0}$$

$$V_{R0}= V_{WL0}-V_{mid}$$

Using these data to calculate the voltage that transistor and resistance consume, we get a graph (Figure 8) and two tables (Table 2 and Table 3)

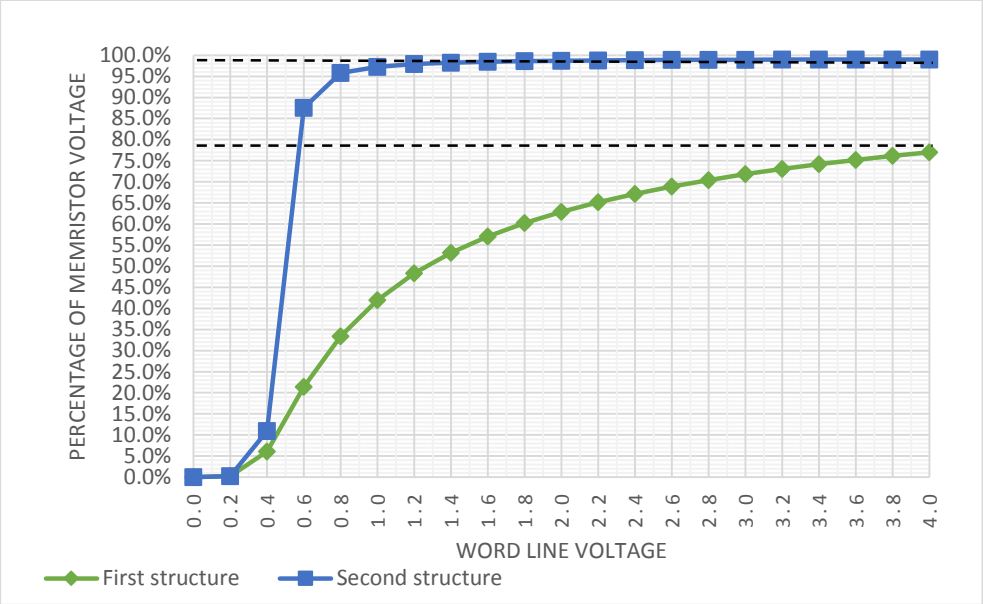

Figure 8. Performance comparison of two structures

Table 2. Performance of the first structure

V_{WL0}/V	V_{mid}/V	V_{BL0}/V	V_{T0}/V	V_{R0}/V	$V_{R0}+V_{T0}$/V	V_{R0}/V_{WL0}
0.0	0.00000	0.00000	0.00000	0.00000	0.00000	0.000%
0.2	0.00046	0.00000	0.19954	0.00046	0.20000	0.231%
0.4	0.02451	0.00004	0.37549	0.02448	0.39996	6.120%
0.6	0.12855	0.00019	0.47145	0.12836	0.59981	21.393%
0.8	0.26733	0.00040	0.53267	0.26693	0.79960	33.366%
1.0	0.42030	0.00063	0.57970	0.41967	0.99937	41.967%
1.2	0.58047	0.00086	0.61953	0.57961	1.19914	48.301%
1.4	0.74577	0.00111	0.65423	0.74466	1.39889	53.190%
1.6	0.91453	0.00136	0.68547	0.91317	1.59864	57.073%
1.8	1.08629	0.00162	0.71371	1.08467	1.79838	60.259%
2.0	1.26034	0.00188	0.73966	1.25846	1.99812	62.923%
2.2	1.43652	0.00214	0.76348	1.43438	2.19786	65.199%
2.4	1.61443	0.00240	0.78557	1.61202	2.39760	67.168%
2.6	1.79400	0.00267	0.80600	1.79133	2.59733	68.897%
2.8	1.97495	0.00294	0.82505	1.97201	2.79706	70.429%
3.0	2.15728	0.00321	0.84272	2.15407	2.99679	71.802%
3.2	2.34077	0.00348	0.85923	2.33729	3.19652	73.040%
3.4	2.52543	0.00376	0.87457	2.52167	3.39624	74.167%
3.6	2.71086	0.00404	0.88914	2.70683	3.59596	75.190%
3.8	2.89758	0.00431	0.90242	2.89326	3.79569	76.139%
4.0	3.08530	0.00459	0.91470	3.08071	3.99541	77.018%

Table 3. Performance of the second structure

V_{WL0}/V	V_{mid}/V	V_{BL0}/V	V_{T0}/V	V_{R0}/V	$V_{R0}+V_{T0}$/V	V_{R0}/V_{WL0}
0.0	0.00000	0.00000	0.00000	0.00000	0.00000	0.000%
0.2	0.19953	0.00000	0.19953	0.00047	0.20000	0.237%
0.4	0.35618	0.00007	0.35611	0.04382	0.39993	10.955%
0.6	0.07462	0.00078	0.07384	0.52538	0.59922	87.563%
0.8	0.03382	0.00114	0.03267	0.76618	0.79886	95.773%
1.0	0.02754	0.00145	0.02609	0.97246	0.99855	97.246%
1.2	0.02529	0.00175	0.02354	1.17471	1.19825	97.893%
1.4	0.02459	0.00205	0.02254	1.37541	1.39795	98.243%
1.6	0.02458	0.00235	0.02223	1.57542	1.59765	98.464%
1.8	0.02500	0.00265	0.02236	1.77500	1.79735	98.611%
2.0	0.02570	0.00294	0.02276	1.97430	1.99706	98.715%
2.2	0.02661	0.00324	0.02337	2.17339	2.19676	98.790%
2.4	0.02769	0.00354	0.02416	2.37231	2.39646	98.846%
2.6	0.02892	0.00383	0.02509	2.57108	2.59617	98.888%
2.8	0.03027	0.00413	0.02614	2.76973	2.79587	98.919%
3.0	0.03173	0.00442	0.02731	2.96827	2.99558	98.942%
3.2	0.03330	0.00472	0.02858	3.16670	3.19528	98.959%
3.4	0.03497	0.00502	0.02995	3.36503	3.39498	98.971%
3.6	0.03673	0.00531	0.03142	3.56327	3.59469	98.980%
3.8	0.03859	0.00561	0.03298	3.76141	3.79439	98.984%
4.0	0.04053	0.00590	0.03463	3.95947	3.99410	98.987%

It is very clear to tell the crossbar's conductance performance from the tables above. When the input voltage is less than 0.3V, the conductance of the two transistors in each structure shows nearly the same and the dramatically low. However, when the input voltage approaching and exceeding 0.6V, the memristor in the second structure can occupy more percentage of voltage than the first structure. Also, it's evident that in the first table, when the voltage is relatively small (such as 1.2V or lower voltage), the voltage of the transistor is even larger than it is on the resistance. This results from non-ideal performance for a crossbar array, such as parasite capacitance and resistance, sneak path and leakage current.

As mentioned above, the voltage of the transistor in a crossbar cell should be as small as possible. Therefore, the second structure will be used in the following investigation due to its better performance.

Moreover, the theoretical explanation will be presented as follows: The relationship of I_{ds} and V_{ds} shows as below (Formula 3.1) [24][25]:

$$I_{ds} = \begin{cases} 0 & V_{gs} < V_t \quad \text{Cutoff} \\ \beta\left(V_{GT} - V_{ds}/2\right)V_{ds} & V_{ds} < V_{\text{dsat}} \quad \text{Linear} \\ \frac{\beta}{2}V_{GT}^2 & V_{ds} > V_{\text{dsat}} \quad \text{Saturation} \end{cases}$$

In the first structure, for the transistor, $V_{gs} = V_{ds}$, $V_{dsat} = V_{gs} - V_t$ (Formula 3.2). Therefore, $V_{ds} > V_{dsat}$, the transistor works in saturation zone. While in the second structure, $V_{ds} < V_{gs}$ and $V_{ds} < V_{gs} - V_t$ since the memristance is very large, the transistor works in linear zone. It is obviously that the conductance of transistors is larger when working in the linear area than that in saturation zone. So, the transistor in the second structure has a higher conductance which leads to a lower voltage.

3.2 OPTIMIZATION OF PARAMETERS

This part shows the progress of optimization of the parameters, especially the transistor width for its dominant influence on sensing current. First, a mathematical method is utilized to give a derivation and find out an optimized width size which is between 400nm and 4000nm. And parallel simulation is conducted to validate the theoretical analysis.

$$I_{ds} = \frac{Q_{channel}}{L/v} = \mu C_{OX}\frac{W}{L}\left(V_{gs} - V_t - \frac{V_{ds}}{2}\right)V_{ds} = \beta(V_{GT} - \frac{V_{ds}}{2})V_{ds} \quad \text{(Formula 3.3)}$$

$$\text{Here, } \beta = \mu C_{OX}\frac{W}{L} \qquad \text{(Formula 3.4)}$$

$$I_{ds} = \beta\left(V_{gs} - V_t - \frac{V_{ds}}{2}\right)V_{ds} = \frac{V_{wl} - V_{ds}}{R_0 + R_{line}}$$

Here, R_{line} is relatively so small that we can ignore it in this derivation. Thus,

$$V_{wl} = V_{gs}$$

$$I_{ds} = \beta\left(V_{wl} - V_t - \frac{V_{ds}}{2}\right)V_{ds} = \frac{V_{wl} - V_{ds}}{R_0}$$

$$\beta R_0\left(V_{wl} - V_t - \frac{V_{ds}}{2}\right)V_{ds} + V_{ds} = V_{wl}$$

$$-\frac{1}{2}\beta R_0 V_{ds}^2 + [(V_{wl} - V_t)\beta R_0 + 1]V_{ds} - V_{wl} = 0$$

$$\text{Substitute } \beta = \mu C_{OX}\frac{W}{L} \text{ into the equation above,}$$

$$-\frac{1}{2}\mu C_{OX}\frac{W}{L}R_0 V_{ds}^2 + \left[(V_{wl} - V_t)\mu C_{OX}\frac{W}{L}R_0 + 1\right]V_{ds} - V_{wl} = 0$$

Here, R_0, V_{wl} and V_t are all constants decided by manufacturing process and technology. So, a graph (Figure 9) of relationship between W and V_{ds} can be plotted as below, when $R_0 = 100K\Omega$, $V_{wl} = 3.3V$ and $V_t = 0.58V$.

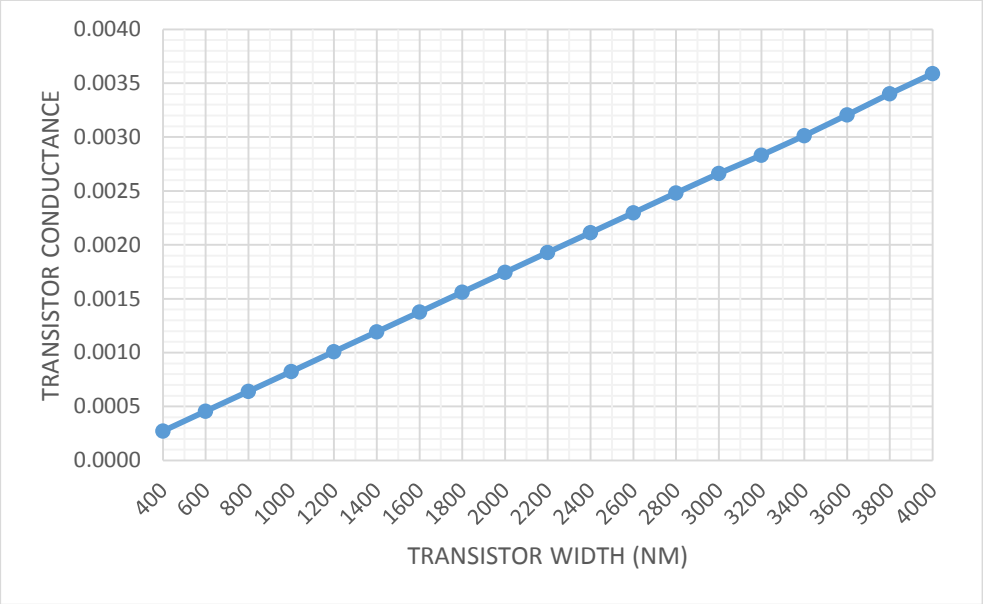

Figure 9. Relationship between Transistor width and transistor conductance

From the derivation and graph about, it can be noticed that the larger transistor width, the better conductance. Thus, the 4000nm transistor width will be used in the models in the following chapters.

3.3 THE CONDUCTANCE OF ONE CROSSBAR CELL

As studied above, the crossbar cell will be utilized in this section is the one that the transistor inside is connected with a memristor and line resistance. The transistor width is chosen to be 4.0μm, as discussed above. Moreover, the memristance is determined to be 100KOhm and 150KOhm separately. By simulation, a line chart can be obtained as below (Figure 10). The DC voltage is swept from 0V to 4V, which is the standard working zone for the transistor in this library.

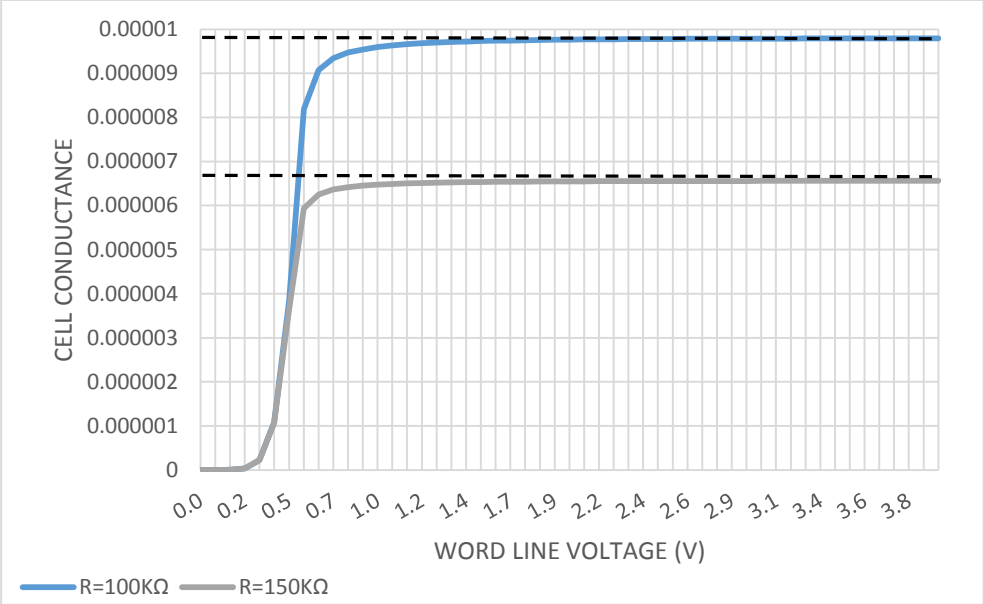

Figure 10. Relationship of input voltage and cell conductance

From the figure above, the conductance property can easily tell. First, when the writing voltage is less than 0.5V or so, the conductance is very low since the transistor is working in the cutoff zone. When the writing voltage reaches and exceeds 0.5V, the conductance improves rapidly since the transistor collaborates in the linear area. After the voltage reaching 1.2V or so, the conductance remains high, with a relatively very small slope, which is because the conductance is decided mainly by memristance. So the writing voltage should better be a relatively high voltage, like 3.3V or above.

4.0 PROGRAMMABILITY STUDY OF MEMRISTOR CROSSBAR CELL

The programmability of the memristor crossbar is closely related to the voltage applied, line resistance and the crossbar scale. In this chapter, the main work is to research the programmability of the memristor crossbar.

4.1 IR-DROP OF ONE CROSSBAR CELL

With the conductance line presented, the next step is to study the programming property of the memristor crossbar. First, a cell will be examined, followed by a column and, at last, an array. Due to the resistance of the WL and the BL, there is a voltage drop across the network, commonly referred to as the IR-drop [26]-[29]. The objective is to collect the data of IR-drop of the crossbar to forecast the programmability of the crossbar array. If the IR-drop is too large, then the programmability will decrease as the scale of the crossbar array increase exponentially the IR-drop impact on both programming and sensing cannot be ignored when the crossbar size increases exponentially [30].

In the beginning, the programmability of a cell, *i.e.* a one by one crossbar array will be studied. From the result, the IR-drop of one cell can be observed, which laid a foundation for the IR-drop study for crossbar column and array. The schematic of one crossbar cell is shown as below

(Figure 11), with the WL resistance R_{wl} and the bit line resistanceR_{bl}, the transistor ($T_{0,0}$) width being 4.0um and length being 400nm.

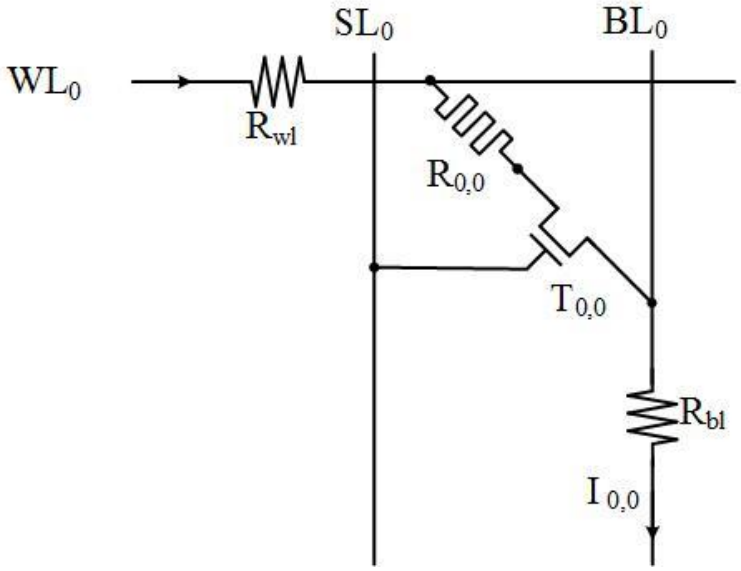

Figure 11. Schematic of one by one crossbar cell with line resistance

4.2 IR-DROP WITH FIXED RESISTANCE RATIO

Here the effective programming voltage ratio $\frac{V_{R_{i,j}}}{V_{wl}}$ which represents the percentage of voltage on memristor is defined to measure the programming efficiency of one cell. In this section, the main objective is to explore the relationship between IR-drop and efficient programming voltage on the condition that ratio between memristance and line resistance k is fixed, but memristance varies. Here, the line resistor ranges from 10Ohm to 200Ohm while the memristance ranges from 500Ohm to 2×10^4Ohm. The effective voltage ratio refers to the ratio between the voltage of memristor and voltage of word line. The schematic is shown in Figure 12. By simulation, a graph can be obtained as below (Figure 13), when k equals 50, 100, 500, 1000, separately.

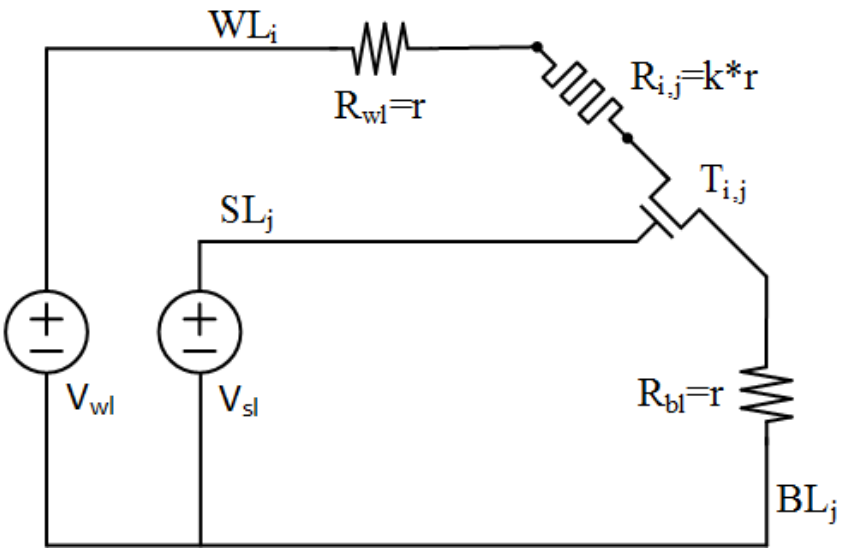

Figure 12. Schematic of programming selected memristor crossbar cell

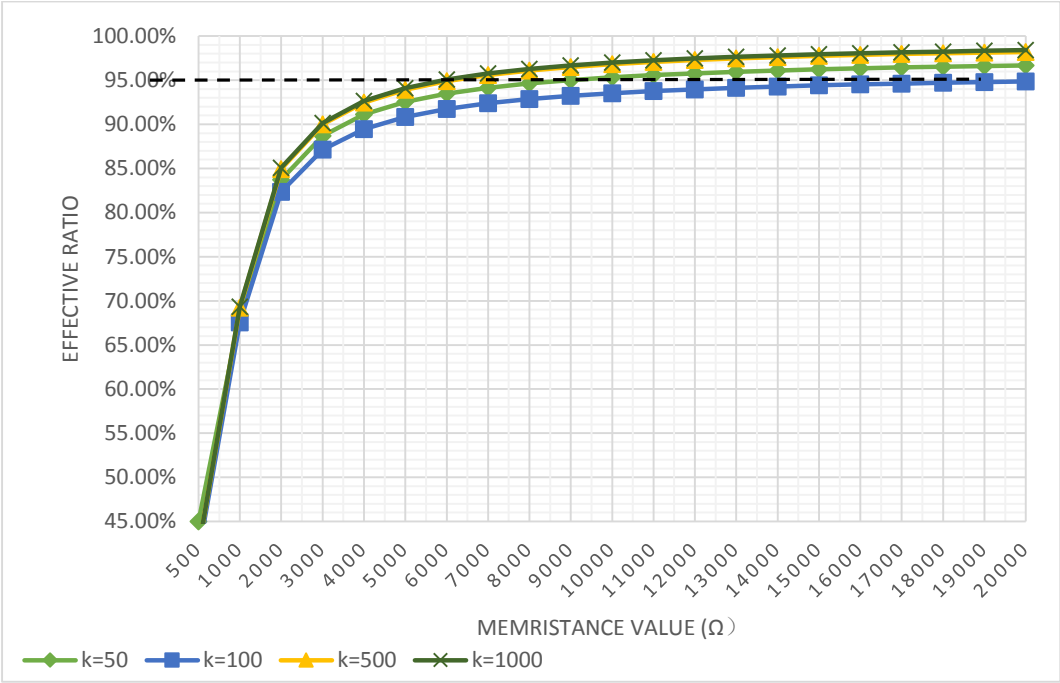

Figure 13. Effective voltage ratio with fixed k, memristor size varying from 500Ohm to 20KOhm

As the figure shows, when the memristance is relatively small, the effective programming voltage ratio is relatively low. As the memristance increases, the programmability increases, first

rapidly and then slowly. This trend is because of the voltage division. When memristor is quite small, the difference between memristor and line resistance is relatively low. In this case, the line resistance will occupy a considerable percentage of the writing voltage. Moreover, when the memristor is large enough, the line resistance can consume little voltage so that the programming is more efficient.

Moreover, for different k value, with the parameter k becomes larger, the memristor is easier to be programmed at the same memristance. The is because smaller k indicates smaller line resistance when the memristor is fixed, and smaller resistance indicates occupying lower voltage ratio.

4.3 IR-DROP WITH DIFFERENT RESISTANCE RATIOS

In this section, the dependency of actual programming voltage ratio and the resistance ratio k will be explored. Based on the previous research work [31]-[33], the proportion of memristance and line resistance varies from 50 to 1000. Thus, the memristance will be fixed while k will be swept from 50 to 1000 in the subsection 4.3.1. The line resistance remains constant when k varied from 50 to 1000 in the subsection 4.3.2.

4.3.1 Fixed memristance

In this subsection, the memristance is fixed at 500Ohm, 1KOhm, 5KOhm, 10KOhm separately, while the 1/k varies from 0.001 to 0.02. Figure 14 depicts the effective voltage ratio as resistance ratio varies which offers a reference for memristor crossbar design.

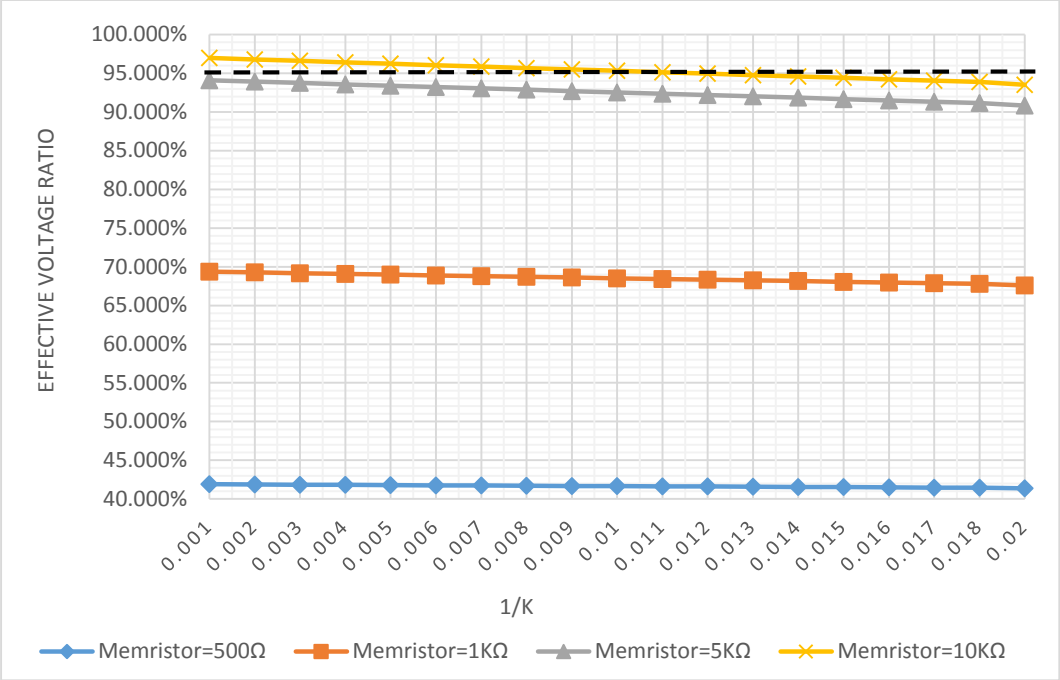

Figure 14. Effective voltage ratio with fixed memristor size, 1/k varying from 0.001 to 0.02

From the line graph above, one can easily tell the memristor is hard to be programmed when the memristor is relatively small, like less than 5KOhm. The reason is the difference between memristor and line resistance is relatively low so that the line resistance can occupy a large ratio of writing voltage. When the memristor is greater than 5KOhm, the effective voltage ratio can reach 95%. The results are mainly because more massive resistance can be a better voltage divider. Thus, the designer should better select memristors with greater resistance for better programming.

4.3.2 Fixed line resistance

In this subsection, the line resistance is fixed when k changes from 50 to 1000. The result is shown in Figure 15.

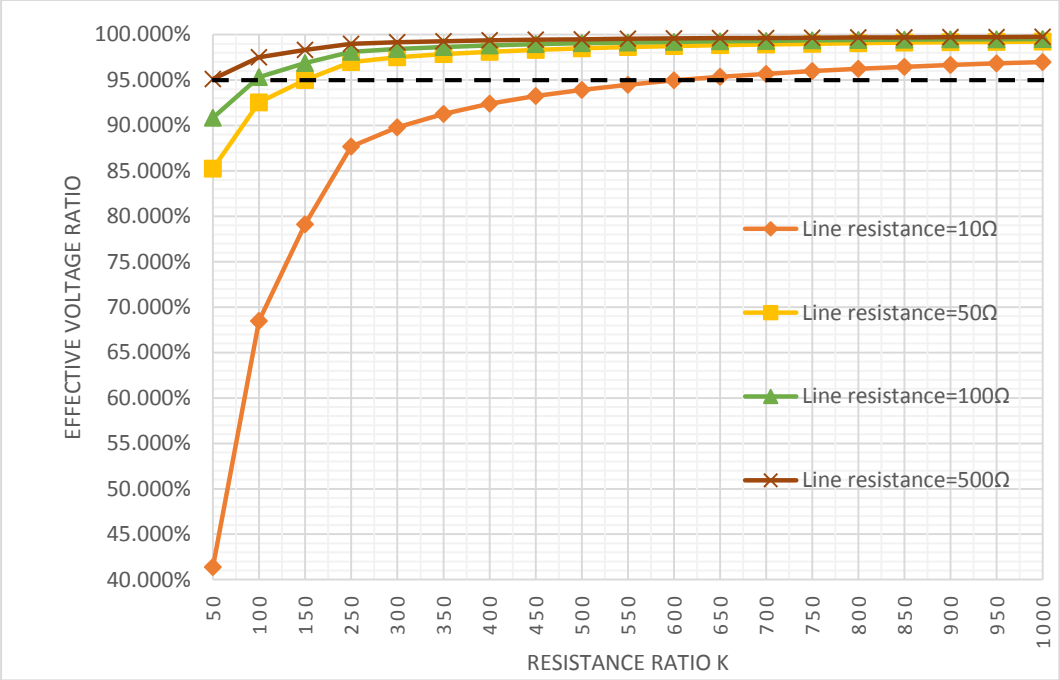

Figure 15. Effective voltage ratio with fixed line resistance, k varying from 50 to 1000

From the graph above, it can be concluded that when the resistance ratio k is relatively high, like above 250, the programming of the memristor is easier to realize. Meanwhile, it is harder to program when the memristance is small than it is large since the transistor has the similar resistance with a memristor of small size, like 500Ohm or so. Thus, designers should choose a larger k and an appropriate line resistance based on their *Process Design Kit* (PDK).

5.0 DESIGN MARGIN OF THE CROSSBAR ARRAY SIZE

With the IR-drop foundation laid, in this section, the available memristor crossbar size will be explored. First, a 32 by 1 crossbar array will be studied, then a 32 by 32 and 64 by 64 crossbar array.

5.1 CROSSBAR COLUMN PROGRAMMING

The schematic is shown as below (Figure 16), with the word line resistance and the bit line resistance considered. If we want to program a memristor, take $R_{1,0}$ for example, the word line should be connected to 3.3V and the other word lines be connected to 0V. The selecting line SL_0 is connected to 3.3V while the bit line BL_0 is connected to 0V. When programming $R_{1,0}$, transistors in other lines are all in cutoff zone so that the current flows only through $R_{1,0}$ and through the bit line to the sensing circuit (Figure 16).

In this subsection, several memristors will be written in different bits to observe the programming performance when the different bit is being written. I will write the memristor $R_{0,0}$,$R_{7,0}$, $R_{15,0}$, $R_{23,0}$ and $R_{31,0}$ and record their effective voltage ratio. The parameters are shown in Table 4.

Table 4. Parameter reference

Parameters	Writing voltage/V	Transistor Width/nm	Transistor Length/nm	Line resistance/Ohm	Memristor Resistance/Ohm
value	3.3	4000	400	100	100K

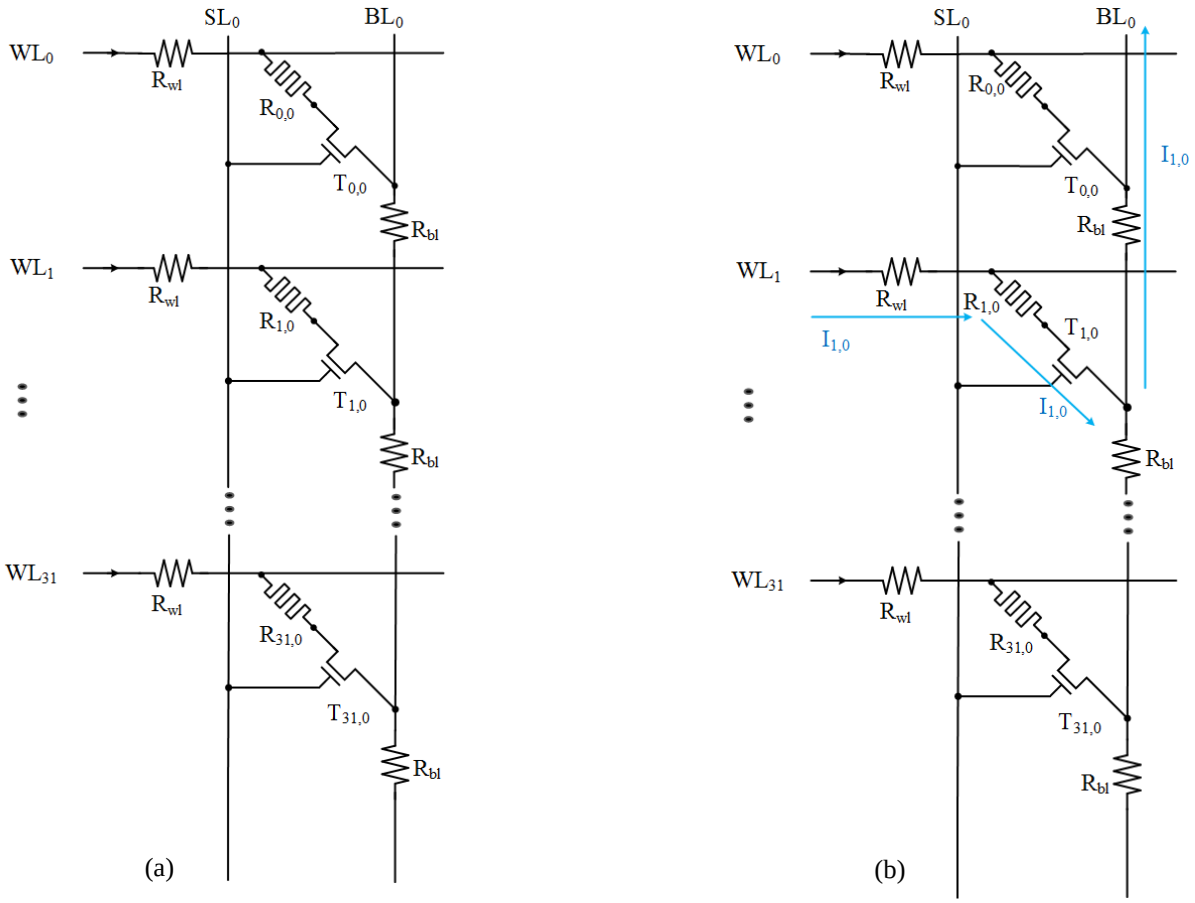

Figure 16. Schematic of a 32 by 1 crossbar column with line resistance (a) and current flow indication when programming the memristor $R_{1,0}$ (b).

Table 5. Performance of programming the 32 by 1 crossbar array

Written bit	Writing voltage/V	$V_{R_{WL}}/V$	V_R/V	V_T/V	Effective voltage ratio
$R_{0,0}$	3.3	0.003210	3.293652	0.003148	99.8076%
$R_{7,0}$	3.3	0.00318	3.292562	0.004258	99.7746%
$R_{15,0}$	3.3	0.003158	3.291217	0.005625	99.7338%
$R_{23,0}$	3.3	0.003231	3.290157	0.050440	99.7017%
$R_{31,0}$	3.3	0.003211	3.288742	0.008047	99.6588%

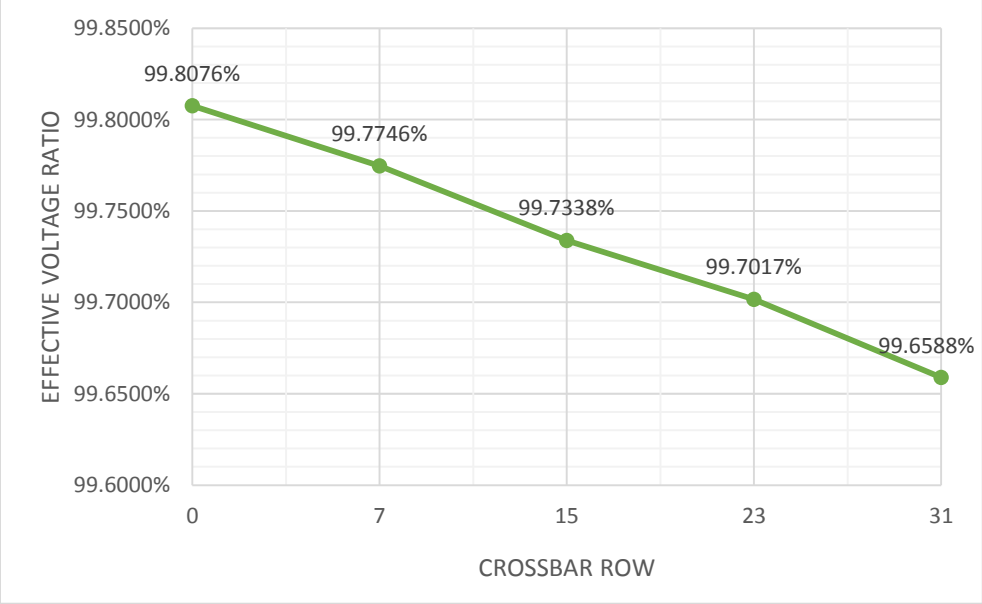

Figure 17. The memristor crossbar column programming performance

As Table 5 and Figure 17 shows, the effective programming voltage ratio decreases as the accumulation of the line resistance. In this situation, the memristor can always be programmed since all the good voltage ratio is larger than 95%. This transistor size, memristance and value k will be used in the following design exploration.

5.2 CROSSBAR ARRAY PROGRAMMING

5.2.1 The 32 by 32 crossbar array

For a 32 by 32 crossbar array (Figure 18), the memristor $R_{0,0}$,$R_{7,7}$, $R_{15,15}$, $R_{23,23}$ and $R_{31,31}$ as well as $R_{31,0}$ and $R_{0,31}$ will be observed. The memristance is chosen to be 100KOhm and line resistance 100Ohm.

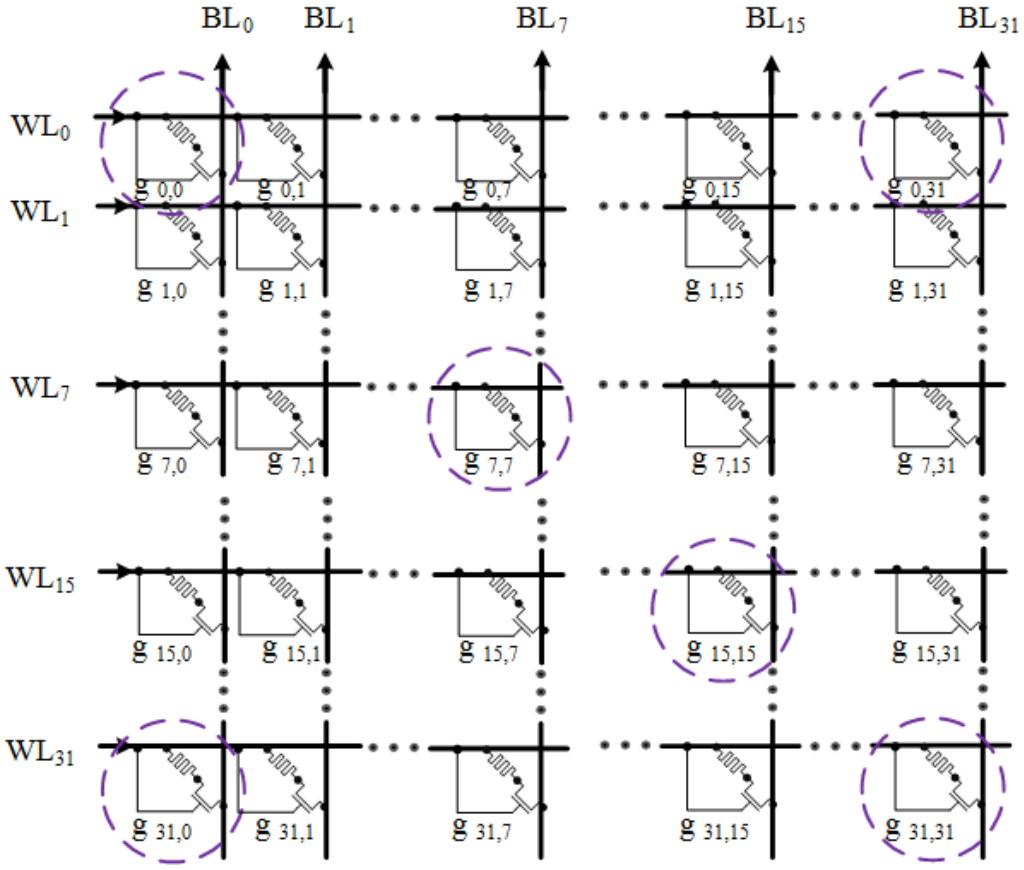

Figure 18. Schematic of 32 by 32 crossbar

When writing memristor $R_{0,0}$, the WL0 is connected to 3.3V and BL0 connected to 0V, and the SL0 is connected to 3.3V. The other selecting lines, word lines and bit lines are all connected to 0V. The performance is shown in Table 6 and Figure 19.

Table 6. Performance of programming the 32 by 32crossbar array

Written bit	Writing voltage/V	$V_{R_{WL}}$/V	V_R/V	V_T/V	Effective voltage ratio
$R_{0,0}$	3.3	0.003285	3.293506	0.009166	99.803%
$R_{7,7}$	3.3	0.003243	3.251996	0.009068	99.228%
$R_{15,15}$	3.3	0.003201	3.20955	0.008966	98.695%
$R_{23,23}$	3.3	0.003159	3.168065	0.008867	98.164%
$R_{31,31}$	3.3	0.003409	3.123051	0.008761	97.490%
$R_{31,0}$	3.3	0.003285	3.293506	0.009166	99.803%
$R_{0,31}$	3.3	0.003243	3.251996	0.009068	99.228%

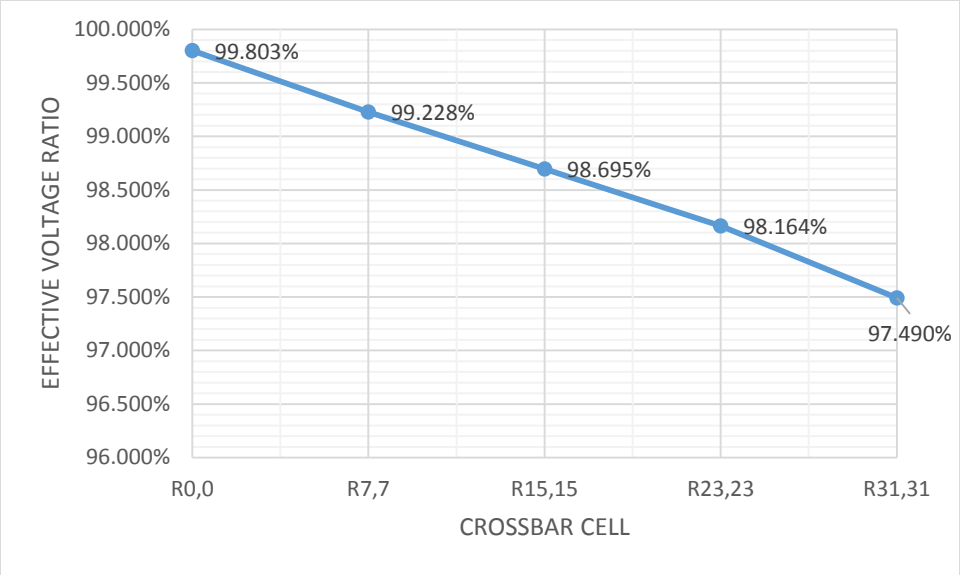

Figure 19. The 32 by 32 memristor crossbar array programming performance

5.2.2 The 64 by 64 crossbar array

For the 64 by 64 crossbar array, nine crossbar cells will be observed as Table 7 and Figure 20.

Table 7. Performance of programming the 64 by 64crossbar array

Written bit	Writing voltage/V	$V_{R_{WL}}$/V	V_R/V	V_T/V	Effective voltage ratio
$R_{0,0}$	3.3	0.003300	3.284345	0.009166	99.526%
$R_{7,7}$	3.3	0.003245	3.244222	0.009071	98.310%
$R_{15,15}$	3.3	0.003207	3.206022	0.008976	97.152%
$R_{23,23}$	3.3	0.003173	3.172549	0.008891	96.138%
$R_{31,31}$	3.3	0.003141	3.141411	0.015687	95.194%
$R_{39,39}$	3.3	0.003111	3.110803	0.008731	94.267%
$R_{47,47}$	3.3	0.003079	3.079059	0.008649	93.305%
$R_{55,55}$	3.3	0.003044	3.044227	0.008562	92.249%
$R_{63,63}$	3.3	0.003004	3.003620	0.008466	91.019%

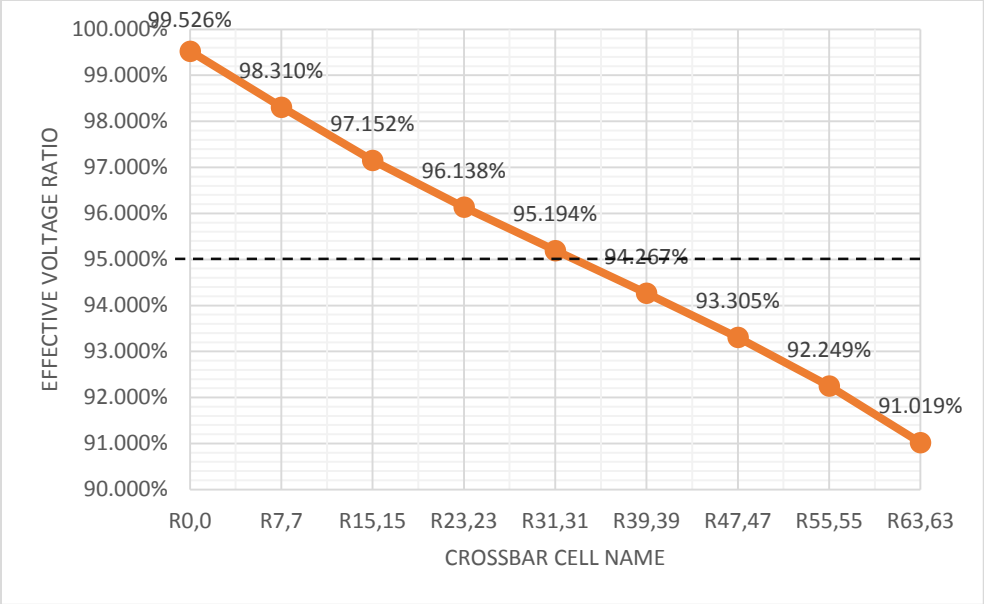

Figure 20. Programming performance of 64 by 64 memristor crossbar

For a 64 by 64 crossbar array, it can be seen that the programming is quite difficult for the corner memristor $R_{63,63}$. The phenomenon is because when programming memristor $R_{63,63}$, the current also flows through 64-word line resistors and 64-bit line resistors. These line resistors are all voltage dividers that worsen the programming progress.

Also, from the 64 by 64 crossbar array, we can deduce that the programming of 128 by 128 crossbar array is hard to realize since half of the cells inside are not able to receive enough writing voltage and current. Thus, designers should take care of the crossbar array scale.

Possible solutions to expand the programmable crossbar size is to eliminate line resistance or to use more efficient NMOS transistors which have higher conductance.

6.0 CONCLUSION AND FUTURE WORK

In this work, I investigate the individual influence that each factor can bring to the programming process of crossbar cell. Based on the analysis, I simulate and analyze the relationship between programming performance of one crossbar cell and some variables like line resistance and resistance ratio, etc. Last but not least, I explored the design margin of memristor crossbar scale to help guide the memristor crossbar design.

Based on all the design exploration above, designers can know how to choose structures, transistor size, memristance and line resistance. Moreover, a crossbar that is larger than 64 by 64 size should be avoid.

In the future, the optimization of programming needs further exploration. Also, the sensing process of the memristor crossbar array deserves a profound study. All of the above are of significance in memristor crossbar array design and can help improve the performance of neuromorphic circuit design.

BIBLIOGRAPHY

[1] M. S. Bartlett, G. Littlewort, M. Frank, C. Lainscsek, I. Fasel, and J. Movellan, "Recognizing facial expression: machine learning and application to spontaneous behavior." pp. 568-573.

[2] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor, "Canonical correlation analysis: An overview with application to learning methods," *Neural computation,* vol. 16, no. 12, pp. 2639-2664, 2004.

[3] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature,* vol. 521, no. 7553, pp. 436-444, 2015.

[4] A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern recognition,* vol. 30, no. 7, pp. 1145-1159, 1997.

[5] S. M. Weiss, and I. Kapouleas, "An empirical comparison of pattern recognition, neural nets and machine learning classification methods," *Readings in machine learning,* pp. 177-183, 1990.

[6] D. C. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Deep, big, simple neural nets for handwritten digit recognition," *Neural computation,* vol. 22, no. 12, pp. 3207-3220, 2010.

[7] C. Yakopcic, and T. M. Taha, "Energy efficient perceptron pattern recognition using segmented memristor crossbar arrays." pp. 1-8.

[8] S. H. Jo, T. Chang, I. Ebong, B. B. Bhadviya, P. Mazumder, and W. Lu, "Nanoscale memristor device as synapse in neuromorphic systems," *Nano letters,* vol. 10, no. 4, pp. 1297-1301, 2010.

[9] K. Itoh, and M. Horiguchi, "Low-voltage scaling limitations for nano-scale CMOS LSIs," *Solid-State Electronics,* vol. 53, no. 4, pp. 402-410, 2009.

[10] J. Kawa, *Design for manufacturability and yield for nano-scale CMOS*: Springer Science & Business Media, 2007.

[11] S.-W. Sun, and P. G. Tsui, “Limitation of CMOS supply-voltage scaling by MOSFET threshold-voltage variation,” *Solid-State Circuits, IEEE Journal of,* vol. 30, no. 8, pp. 947-949, 1995.

[12] P. Machanick, “Approaches to addressing the memory wall,” *School of IT and Electrical Engineering, University of Queensland*, 2002.

[13] S. A. McKee, "Reflections on the memory wall." p. 162.

[14] Y. N. Joglekar, and S. J. Wolf, “The elusive memristor: properties of basic electrical circuits,” *European Journal of Physics,* vol. 30, no. 4, pp. 661, 2009.

[15] S. R. Williams, “How we found the missing memristor,” *Spectrum, IEEE,* vol. 45, no. 12, pp. 28-35, 2008.

[16] D. Chabi, D. Querlioz, W. Zhao, and J.-O. Klein, “Robust learning approach for neuro-inspired nanoscale crossbar architecture,” *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 10, no. 1, pp. 5, 2014.

[17] G. Indiveri, B. Linares-Barranco, R. Legenstein, G. Deligeorgis, and T. Prodromakis, “Integration of nanoscale memristor synapses in neuromorphic computing architectures,” *Nanotechnology*, vol. 24, no. 38, pp. 384010, 2013.

[18] R. Kozma, R. E. Pino, and G. E. Pazienza, *Advances in neuromorphic memristor science and applications*: Springer Science & Business Media, 2012.

[19] N. Duraisamy, N. M. Muhammad, H.-C. Kim, J.-D. Jo, and K.-H. Choi, “Fabrication of TiO 2 thin film memristor device using electrohydrodynamic inkjet printing,” *Thin Solid Films,* vol. 520, no. 15, pp. 5070-5074, 2012.

[20] Y. Ho, G. M. Huang, and P. Li, "Nonvolatile memristor memory: device characteristics and design implications." pp. 485-490.

[21] R. E. Pino, J. W. Bohl, N. McDonald, B. Wysocki, P. Rozwood, K. A. Campbell, A. Oblea, and A. Timilsina, "Compact method for modeling and simulation of memristor devices: ion conductor chalcogenide-based memristor devices." pp. 1-4.

[22] C. A. Mead, “Scaling of MOS technology to submicrometer feature sizes,” *Journal of VLSI signal processing systems for signal, image and video technology,* vol. 8, no. 1, pp. 9-25, 1994.

[23] S. E. Thompson, and S. Parthasarathy, “Moore's law: the future of Si microelectronics,” *materials today*, vol. 9, no. 6, pp. 20-25, 2006.

[24] N. D. Arora, *MOSFET models for VLSI circuit simulation: theory and practice*: Springer Science & Business Media, 2012.

[25] T. Sakurai, and A. R. Newton, “A simple MOSFET model for circuit analysis,” *Electron Devices, IEEE Transactions on,* vol. 38, no. 4, pp. 887-894, 1991.

[26] B. Liu, H. Li, Y. Chen, X. Li, T. Huang, Q. Wu, and M. Barnell, "Reduction and IR-drop compensations techniques for reliable neuromorphic computing systems." pp. 63-70.

[27] S. Nithin, G. Shanmugam, and S. Chandrasekar, "Dynamic voltage (IR) drop analysis and design closure: Issues and challenges." pp. 611-617.

[28] M. Sinha, and W. Burleson, "Current-sensing for crossbars." pp. 25-29.

[29] C. Xu, D. Niu, N. Muralimanohar, R. Balasubramonian, T. Zhang, S. Yu, and Y. Xie, "Overcoming the challenges of crossbar resistive memory architectures." pp. 476-488.

[30] J. Liang, and H. P. Wong, “Cross-point memory array without cell selectors—device characteristics and data storage pattern dependencies,” *Electron Devices, IEEE Transactions on,* vol. 57, no. 10, pp. 2531-2538, 2010.

[31] A. Chen, “A comprehensive crossbar array model with solutions for line resistance and nonlinear device characteristics,” *Electron Devices, IEEE Transactions on,* vol. 60, no. 4, pp. 1318-1326, 2013.

[32] J.-J. Huang, Y.-M. Tseng, C.-W. Hsu, and T.-H. Hou, “Bipolar nonlinear selector for 1S1R crossbar array applications,” *Electron Device Letters, IEEE,* vol. 32, no. 10, pp. 1427-1429, 2011.

[33] M. M. Ziegler, and M. R. Stan, “CMOS/nano co-design for crossbar-based molecular electronic systems,” *Nanotechnology, IEEE Transactions on,* vol. 2, no. 4, pp. 217-230, 2003.