INTELLIGENT ENERGY-EFFICIENT STORAGE SYSTEM FOR BIG-DATA APPLICATIONS

A Dissertation
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Yifu Gong

In Partial Fulfillment of the Requirements
for the Degree of
DOCTOR OF PHILOSOPHY

Major Department:
Electrical and Computer Engineering

January 2020

Fargo, North Dakota

North Dakota State University
Graduate School

Title

INTELLIGENT ENERGY-EFFICIENT STORAGE SYSTEM FOR BIG-DATA APPLICATIONS

By

Yifu Gong

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

DOCTOR OF PHILOSOPHY

SUPERVISORY COMMITTEE:

Dr. Benjamin Braaten
Chair

Dr. Sudarshan Srinivasan

Dr. Jacob Glower

Dr. Mark McCourt

Approved:

January 23, 2020
Date

Benjamin Braaten
Department Chair

ABSTRACT

Static Random Access Memory (SRAM) is a critical component in mobile video processing systems. Because of the large video data size, the memory is frequently accessed, which dominates the power consumption and limits battery life. In energy-efficient SRAM design, a substantial amount of research is presented to discuss the mechanisms of approximate storage, but the content and environment adaptations were never a part of the consideration in memory design. This dissertation focuses on optimization methods for the SRAM system, specifically addressing three areas of Intelligent Energy-Efficient Storage system design. First, the SRAM stability is discussed. The relationships among supply voltage, SRAM transistor sizes, and SRAM failure rate are derived in this section. The result of this study is applied to all of the later work. Second, intelligent voltage scaling techniques are detailed. This method utilizes the conventional voltage scaling technique by integrating self-correction and sizing techniques. Third, intelligent bit-truncation techniques are developed. Viewing environment and video content characteristics are considered in the memory design. The performance of all designed SRAMs are compared to published literature and are proven to have improvement.

ACKNOWLEDGMENTS

I am deeply grateful to Dr. Benjamin D. Braaten and Dr. Sudarshan Srinivasan for their invaluable supports and editorial advice that have helped complete this work. I would like to thank Dr. Jacob Glower and Dr. Mark McCourt for serving on my graduate committee and taking the time to review and comment on this dissertation. I would like to thank my colleagues and fellow students Dr. Jonathon Edstrom, Dongliang Chen, and Hritom Das for their contributions to this study. For those who were with me through this memorable time in my life, thank you for the company. I am thankful to NDSU Electrical and Computer Engineering department for the financial support that made the presented research possible. Finally, I would like to thank my family for their patience and support throughout my graduate career.

DEDICATION

To my family.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

SRAM .....Static Random Access Memory

CMOS .....Complementary Metal-Oxide-Semiconductor

PSNR.....Peak Signal-to-Noise Ratio

MSE .....Mean Squared Error

MSB .....Most-Significant-Bit

LSB .....Least-Significant-Bit

ECC.....Error-Correction-Code

BIST .....Built-In Self-Test

SNM.....Static Noise Margin

DPSR.....Data Pattern Self-Recovery

ANN.....Artificial Neural Network

VCAS.....Viewing Context-Aware SRAM

DPSR.....Data Pattern Self-Recovery

1. INTRODUCTION

1.1. Background

The modern world is all about video streaming. According to the recent Cisco Visual Networking Index, 79% of total mobile data in 2019 used for Mobile video traffic [1]. Figure 1 shows the predicted growth chart. With the continuous evolution of mobile networks, it is expected to increase 9-fold between 2016 and 2021 [1].

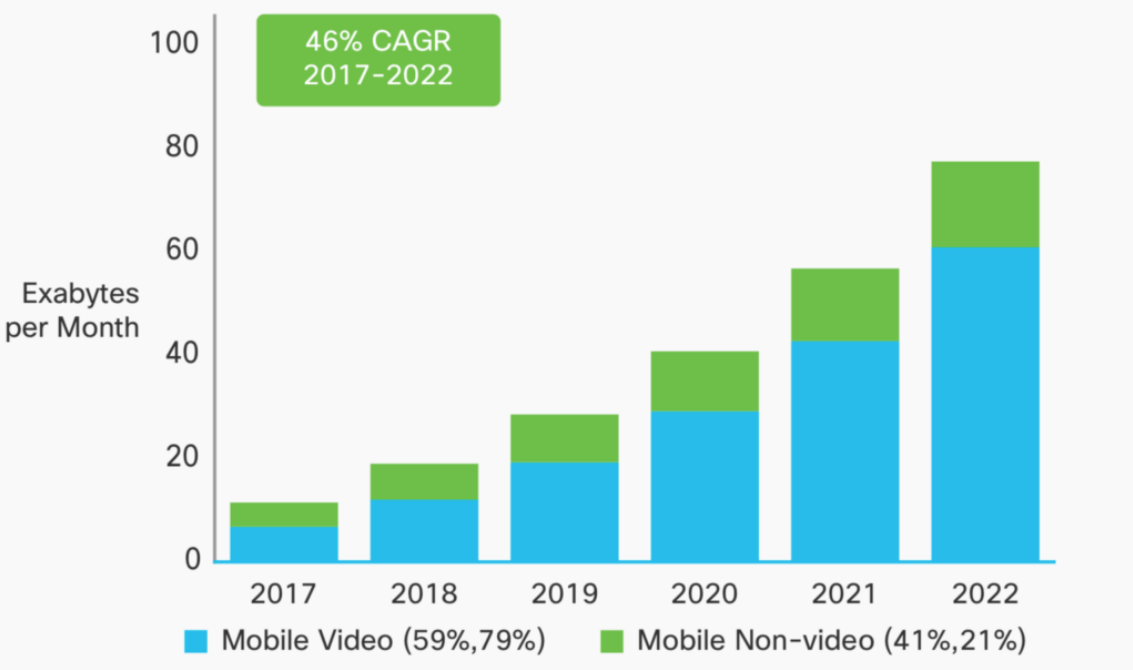


Figure 1. Mobile Data Traffic by 2022 [1].

Video streaming is, therefore, becoming the most energy-consuming applications on mobile devices. During the mobile video steaming process, over 92% of the motion compensation energy [2] and 50% of the video decoding energy consumption [3] comes with frequent memory access, and energy consumption due to video streaming will only continue to increase with the emerging of Ultra-High-Definition (UHD) videos [4]. Accordingly, techniques for enhancing the energy efficiency of video memories are key to the advancement of mobile devices. The study of energy-efficient memory application-specific designs has recently become

a focus of research. Memories that are capable of providing high-quality output while saving power consumption are in demand.

We conducted simulations to calculate important performance parameters, including power efficiency, video quality, and area overhead, in order to test the effectiveness of designed memories. With the variable-controlling approach, the improvement of our work against the state-of-the-art is shown in this dissertation.

1.2. Research Challenges

This research focuses on the design optimization of Static Random-Access Memory (SRAM), a CMOS semiconductor memory, to reduce the effect of failed SRAM cells on output result in approximate computing application. Traditional hardware-level optimization techniques usually come with significant implementation costs to solve the memory failure problem. Two of the most common examples of the costs are the silicon area overhead and performance penalty.

Therefore, the first goal is to come up with a novel energy-efficient SRAM circuit design with simplified additional logic in order to minimize the area overhead. The second goal is to perform simulations to make sure the circuit speed meets the application requirement.

1.3. Statement and Contributions

In this dissertation, design methods of SRAM are investigated and optimized to adapt to different approximate computing applications. The main focus of this dissertation is on reducing the power consumption for approximate computing applications by proposing novel SRAM designs. Rule mining techniques have been used to perform statistical pattern analysis. The discovered data pattern combined with a Data-driven hardware design technique enables an intelligent memory with a better tradeoff between energy efficiency, cost, and accuracy.

The principal contributions of this dissertation are:

- A complete flow is designed to analyze SRAM cell stability. The variables used for analyzing this process include SRAM cell schematics, transistor sizes, and operating voltages. The relation found between the parameters and stability becomes the foundation of this research, which is used for supporting the SRAM design under near-threshold voltage.
- To correct memory faults under low operation voltage with high precision, a self-recovery SRAM design is presented. By comparing the correlation percentages, found by a rule mining technique, the optimized rules can be used to predict the value of adjacent faulty data in memory.
- A novel viewer-aware bit-truncation technique is presented, which enables better visual experience while maintaining similar power efficiency. Based on the developed models and viewer-aware bit-truncation technique, a content-adaptive video memory design with dynamic energy-quality trade-off is implemented. The designed methods are able to adapt to the environment in real-time with minimal quality loss.

1.4. Organization

This dissertation is organized into 10 chapters. Chapter 2 presents previous work and contains an in-depth discussion of fundamental terminology and concepts for low-power SRAM design, which will provide the foundation for the rest of the dissertation. Research on two lo low-power design directions, SRAM cell design and peripheral circuit design are introduced and followed by Application-Specific memory designs. Techniques introduced are adapted, improved, and compared in this work.

Chapter 3 introduces a complete flow to simulate SRAM stability. Based on the physical characteristics for transistors, a method for measuring SRAM cell stability is implemented using a netlist (i.e. a textual description of a circuit made of components). Meaningful data is then extracted from the simulation output using a python script. Finally, by feeding the extracted data into a Matlab script, the failure rate of designed SRAM is acquired. The result of this study was applied to all of the SRAM designs in this dissertation.

Chapters 4 and 5 present the self-recovery techniques. Recently, a new branch of low-voltage embedded memory techniques have been developed to embrace the memory faults, instead of avoiding the faults (assistance techniques or more than 6T cells) or correcting the faults (e.g. ECC). Those techniques aim to mitigate the impact of memory faults by minimizing the magnitude of the error due to a faulty cell, based on the determined memory fault positions from run-time testing (e.g. BIST). By applying a data correlation enabled self-recovery method, self-recovery low-power video memory and an intelligent efficient deep learning synaptic memory are designed and tested for effectiveness. Self-recovery techniques use another bit from the same data or from adjacent data to correct a detected faulty bit.

In Chapters 6 and 7, novel bit-truncation techniques are discussed. These bit-truncation techniques are combined with viewer awareness and a Peak signal-to-noise ratio (*PSNR*) improvement mathematic model. Two low-power video memories are detailed: viewer-aware intelligent video memory and content-adaptive video memory.

Chapter 8 introduces an MTJ based non-volatile SRAM utilizing spin torque transfer magnetization switching in 45 nm technology. The designed non-volatile SRAM is enabled by a sequence of peripheral signals to avoid data loss at power down.

In Chapter 9, machine intelligence is used to enhance visual inspection in welding quality control by developing a low-cost and reliable portable embedded device with advanced machine learning techniques. Our developed device significantly enhances the effectiveness of the visual inspection, which will further enable rapid and cost-effective decision making for welding quality control.

Chapter 10 concludes the contributions of this dissertation and offers a direction for future research.

2. PREVIOUS WORK

To achieve low-power memory design, two directions of approaches are studied. One is adopting more than 6T (6 transistors), a conventional SRAM cell design, to improve the read and write stabilities under low voltage operation such as column-decoupled 8T cells [9], asymmetric 7T cell [10], bit-interleaving 12T cells [11], and read-disturb-free 9T [12]. And the other one is utilizing memory bit-line peripheral circuit schematic to reduce the power consumption such as boosted word-line (WL) voltage [5], adjustment of cell voltage [6], read-modify-write or write-back schemes [7], and dual-rail supply schemes [8]. However, the improvements in the memory power efficiency of those general-purpose design techniques are often achieved with significant design complexity, increased silicon area, and power penalty for voltage regulators or boosting circuits to solve the memory failure issue.

Many recent studies have explored the low-power mobile video memories with application-specific designs. For example, Sinangil et al. [13] present an SRAM that reduces read power consumption by lowering the bit-line switching for Most-Significant-Bits (MSBs). A hybrid 6T+8T SRAM design to achieve quality-power optimization is detailed in Chang et al. [14]. To reduce the conventional 6T bit-cell failure probability, a heterogeneous sizing scheme is shown in [15]. In [16], video memory is presented that uses the Lease-Significant-Bits (LSB) of video data to store error-correction-code (ECC).

2.1. Previous Work on SRAM Cell Design

In 2011, Joshi introduced a novel 8T-CDC column-decoupled SRAM design [9]. In comparison to standard 6T techniques, this design enables enhanced voltage scaling capabilities and 30%–40% power reduction. Monte-Carlo statistical simulation methodology is applied in this paper to study the read and write stability. Monte-Carlo simulation methodology is applied

in this dissertation to analyze the SRAM failure rate. In 2006, a read-static-noise-margin-free SRAM cell is designed to help improve the speed of conventional SRAMs by Takeda [10]. It consists of seven transistors, an additional transistor for loop-cutting is added to a 6T-cell. The area of the proposed SRAM is 23% smaller than that of a conventional SRAM at the same operation speed. In [11], a new bit-interleaving 12T subthreshold SRAM cell is presented by Chiu. With Data-Aware Power-Cutoff (DAPC) Write-assist designed SRAM cell has an improved Write-ability to mitigate device variations at low supply voltage. The measured results show that under the worst-case bit-line data patterns, Data can be successfully read and written at 350 mV (100 mV lower than the threshold voltage).

The solutions listed above are on the SRAM cell itself. Some other work focused on SRAM peripheral circuits that have focused on improving SRAM stability by optimizing control signal circuits will be discussed in the next section.

2.2. Previous Work on SRAM Peripheral Circuits Design

The variation tolerant assist circuits against process variation were proposed in 2008 by Nii [6]. This work lowers the WL voltage to improve the SRAM readability. But the downside is not only the slower speed but the stability of the writing operation. A read assists circuit and a write assist circuit is designed to improve the read and write ability at the same time with less than 10% area overhead. By introducing the read assist circuit, the static noise margin is increased by about 100 mV at 1.0V supply voltage. And compared with the case without assist circuits, the write margin was improved by about 35 mV. Similarly, in 2009, a 512Kb dual-power-supply SRAM is designed by Hirabayashi [5]. This work focused on improving SRAM stability with minimized SRAM size by using two voltage supplies. The cell failure rate is

improved more than three orders of magnitude by using an adaptive WL-level programming scheme and dynamic-array-supply control scheme to increase Static Noise Margin.

In 2009, Kushida proposed a self-write-back sense amplifier in order to improve cell stability [7]. A pair of capacitance separators are inserted between SRAM cells and a sense amplifier to immunize the disturbing effect. The proposed sense amplifier is able to improve the cell failure rate two orders of magnitude at 0.6 V. In 2013, he also presented an SRAM circuit technique to reduce active and standby power consumption at room temperature [8]. In the active mode, the cell supply voltage (V_{CS}) is adaptively controlled by a bit-line power calculator. And a retention circuit regulates V_{CS} in the standby mode. Compared with the conventional scheme, the power consumption in the active and standby modes at room temperature is reduced by 27% and 85%, respectively.

2.3. Previous Work on Application-Specific Memory Design

Application-specific SRAM is customized for a particular use, rather than intended for general-purpose. As a result, for different applications, the best solution can be various. Some of the design thoughts are introduced as follows.

Sinangil proposed a prediction-based reduced bit-line switching scheme to reduce switching activity on the bit-lines [13]. And a sense-amplifier is designed to reduce the energy consumption of the sensing network. Compared with an 8T SRAM, proposed techniques provide up to 1.9× lower energy consumption. A low-power two-port real-time video processing SRAM is proposed by Fujiwara [14]. This SRAM is designed for real-time image processing in which data have a statistical correlation. With propose design, 53% power reduction can be achieved on the bit-lines, and it saves 43% of the total read power. The idea of using correlated data to improve circuit design is applied to our recent researches. The speed and area overheads are 4%

and 11%, respectively. Kwon proposes a heterogeneous SRAM cell sizing architecture in 2012 [15]. SRAM cell size is a key factor to SRAM failures, the failure rate is in an inverse ratio to SRAM size. In this work, SRAM cells with different sizes are applied to store data based on the significance of each data bit. The proposed SRAM design technique achieves 4.49 dB *PSNR* improvement compared to the SRAM with the same cell size at 900-mV supply voltage. SRAM size is an important factor in SRAM stability. The SRAM failure rates and SRAM sizes are mapped in this dissertation, which can be used to establish a mathematical model for optimizing SRAM cell design.

Previous works have shown that the study of energy-efficient storage design has many areas of research. But for video SRAM design, no one has considered subjective perception with the video quality. Video applications have been shown to have a certain degree of resistance or tolerance to errors [38]. This error resistance allows a hardware redesign to enable power saving using approximate computing methods. Most of the memory designs detailed in this dissertation are specifically for video applications, but the design methods can also be applied for other types of applications.

3. SRAM STABILITY

The relative strength of a transistor could be changed by the process parameter variations that break the balance in the symmetric circuit [17]. Reduce the supply voltage (voltage scaling) has the potential to achieve significant power savings. However, voltage scaling aggravates the effect of parameter variations, which could cause memory failures such as read access, read disturb, or write failures [18].

3.1. SRAM Mechanism

Among various SRAM bitcells, 6T and 8T are the two most widely used architectures. 6T achieves optimized silicon area cost, while 8T reduces memory failures due to the decoupled read and write paths. These two types of designs are used in applications with different focuses.

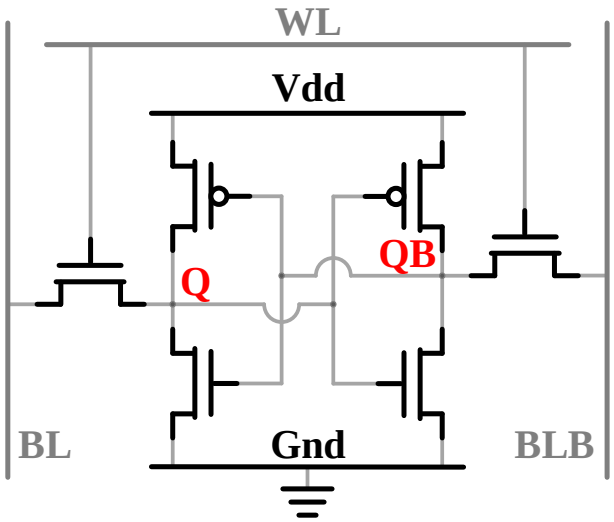


Figure 2. 6T SRAM circuit schematic.

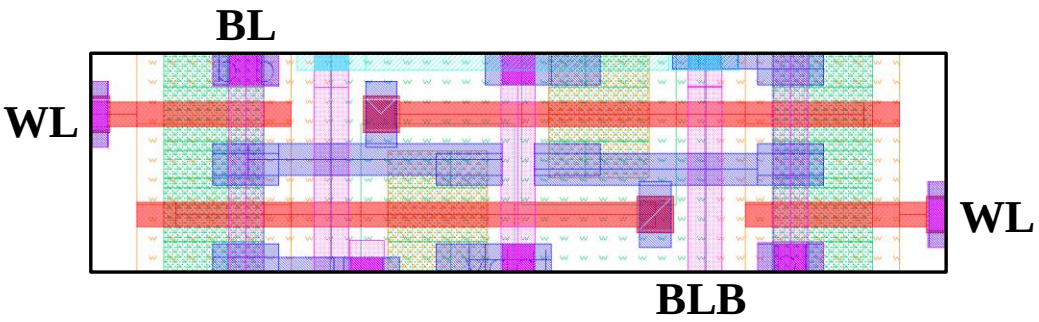


Figure 3. 6T SRAM layout.

The 6T SRAM cell schematic and layout are shown in Figures 2 and 3, respectively. The conventional circuit of the 6T SRAM cell consists of 6 transistors, the circuit is made of two back to back connected inverters along with two NMOS transistors on the sides to access the stored data. Thus, the SRAM bit cell not only stores the data bit (Q) but also the complement (QB). Note that because the adjacent two SRAM cells share the same WL contact in the layout, some of the contacts are not fully displayed.

The reading operation is the state when data is fetched from the memory cell. In order to read data, both bit-line (BL) and bit-line-bar (BLB) are pre-charged to Vdd, and the word line (WL) is low. Then, when the WL is enabled to logic 1 (Vdd), the access transistors on the sides are turned on. As a result, BL and BLB are connected to Q and QB, respectively. Please note that the BL and BLB are floating, which means they are not connected to Vdd anymore but are still holding the charges. A sense amplifier is connected at the end of the bit-lines to produce the output by comparing the voltages in the bit-lines (not shown in the figures).

The writing operation is the state when data is written into the cell. To write data into a cell, BL is pulled to the value of the given data, and BLB takes the complementary value. For example, if the input data is 0 then BL = 0 and BLB = 1 (Vdd); whereas, if data is 1 then BL = 1 and BLB = 0. Q and QB will be updated according to the bit-lines.

The 8T SRAM cell consists of 2 additional transistors connecting to QB, compared with 6T SRAM cell, to create a separate reading path as shown in Figure 4. The layout of the 8T SRAM is shown in Figure 5.

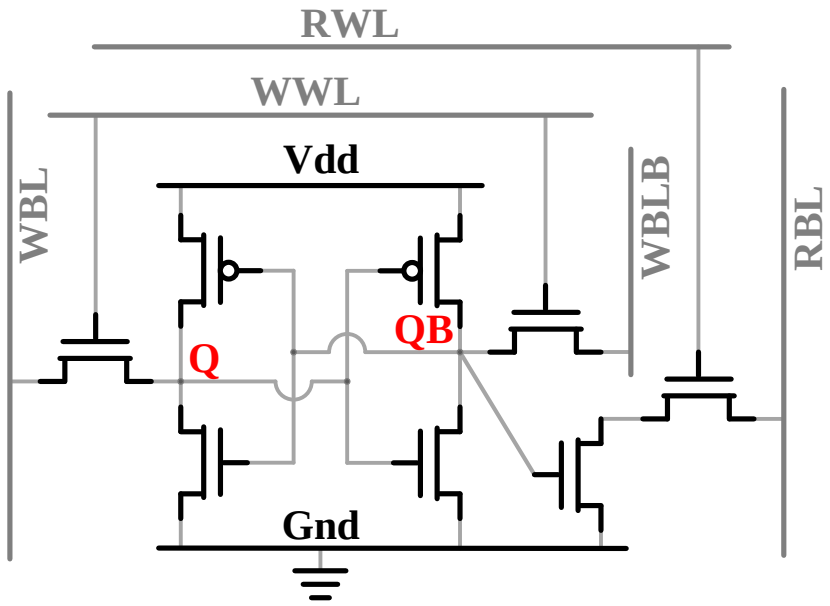


Figure 4. 8T SRAM circuit schematic.

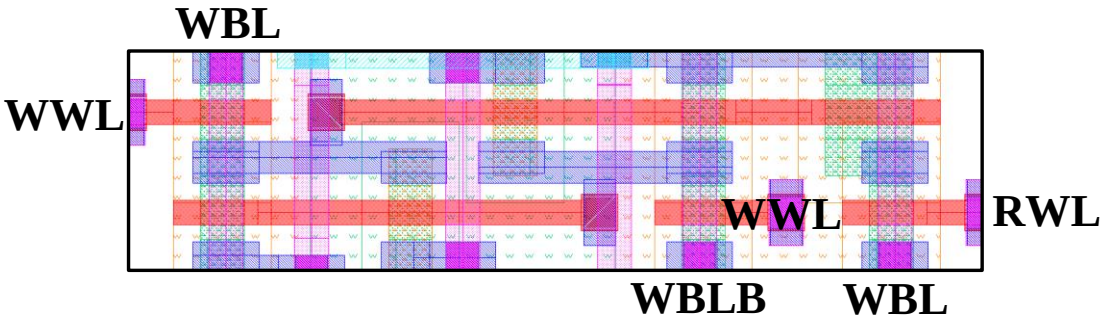


Figure 5. 8T SRAM Layout.

Compared to the 6T SRAM cell, the 8T SRAM has a separate reading path. Two additional lines, read word-line (RWL) and read bit-line (RBL), are added into the circuit to enable the reading operation. During the reading operation, RBL is pre-charged to Vdd. Then, RWL is enabled to logic 1. If the value stored in QB is one, RBL will be discharged to Gnd, otherwise, RBL stays at logic 1.

The writing operation for 8T is similar to 6T. To write data into a cell, write bit-line (WBL) is pulled to the value of the given data, and write bit-line-bar (WBLB) takes the complementary value. Then the write word-line (WWL) is enabled to Vdd. Q and QB will be updated according to the bit-lines.

3.2. SRAM Static Noise Margin

Static Noise Margin (SNM) is a stability measurement of an SRAM cell based on the Voltage Transfer Characteristics (VTC) of the cross-coupled inverters of the SRAM [36]. Figure 6 illustrates a pair of cross-coupled inverters and two equal static noise sources, the schematic of an SRAM cell for simulating the SNM. The SNM of the SRAM cell is defined as the maximum value of V_n that can be allowed before the SRAM cell changing state [36].

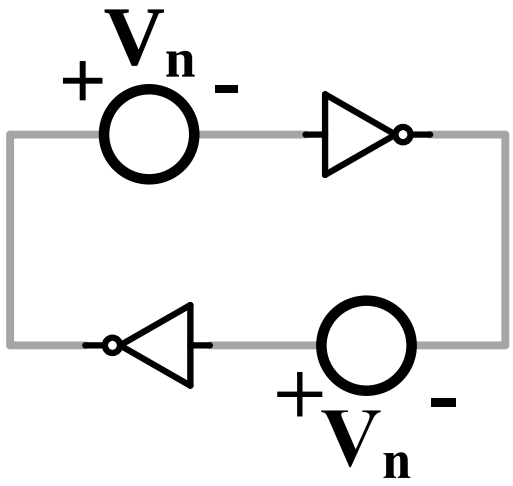


Figure 6. Cross-coupled inverters and two noise sources, Vn.

SRAM cell is reconstructed for testing to form the “butterfly curve”, used for measuring the Read Static Noise Margin (RSNM). The reconstructed 6T read circuit is shown in Figure 7. Because of the process variations, the SRAM cell is not symmetric. Thus, the SRAM cell circuit is divided into two portions and simulated separately. During the simulation, a DC voltage source is sweeping at node V1 from 0v to Vdd, and the voltage at node V2 is measured to obtain the VTC.

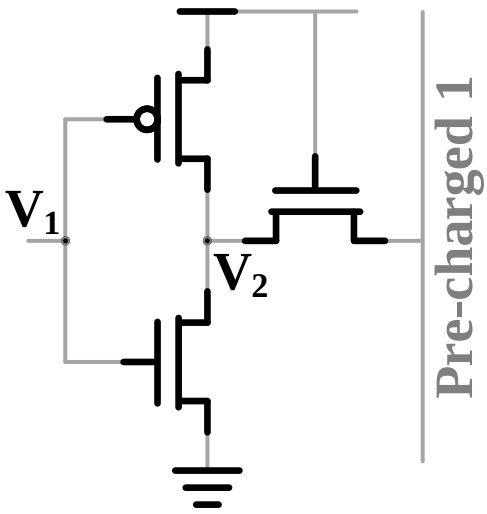


Figure 7. Circuit to measure RSNM for 6T SRAM.

As for the writing operation, since a pair of complementary data are stored in the cross-coupled inverters, different circuits are applied for VTCs to obtain the Write Static Noise Margin (WSNM). The circuit schematics are shown in Figure 8. The measurement of the circuit on the left of Figure 8 is similar to the RSNM. The difference is on the right. It represents the stored data is '1' and the accessed bit-line is grounded to simulate the write '0'operation. A DC voltage source is sweeping at node V1 from 0v to Vdd, and the voltage at node V2 is measured.

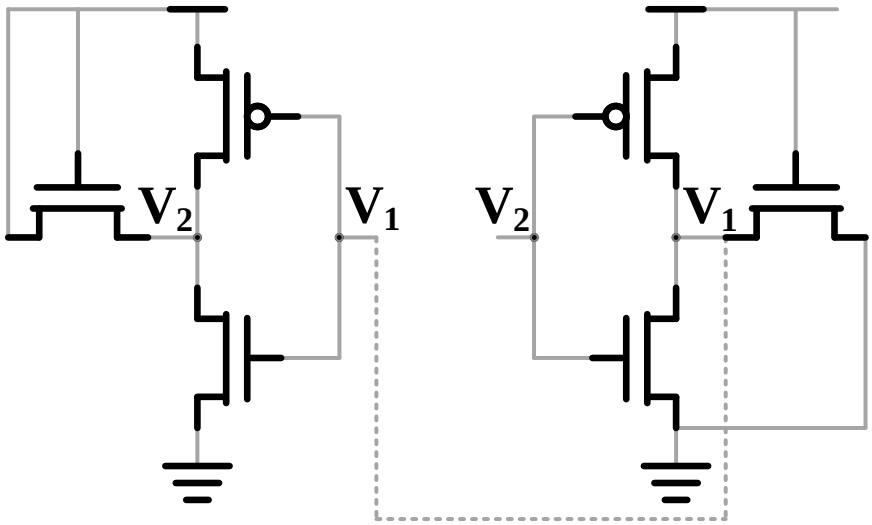


Figure 8. Circuits to measure WSNM for 6T SRAM.

Figure 9 shows examples of an RSNM and a WSNM of 6T. The RSNM can be determined by the size of the largest square inscribed in the area between two curves. Similarly,

the size of the smallest embedded square fits between the lower part of the curves represents the WSNM.

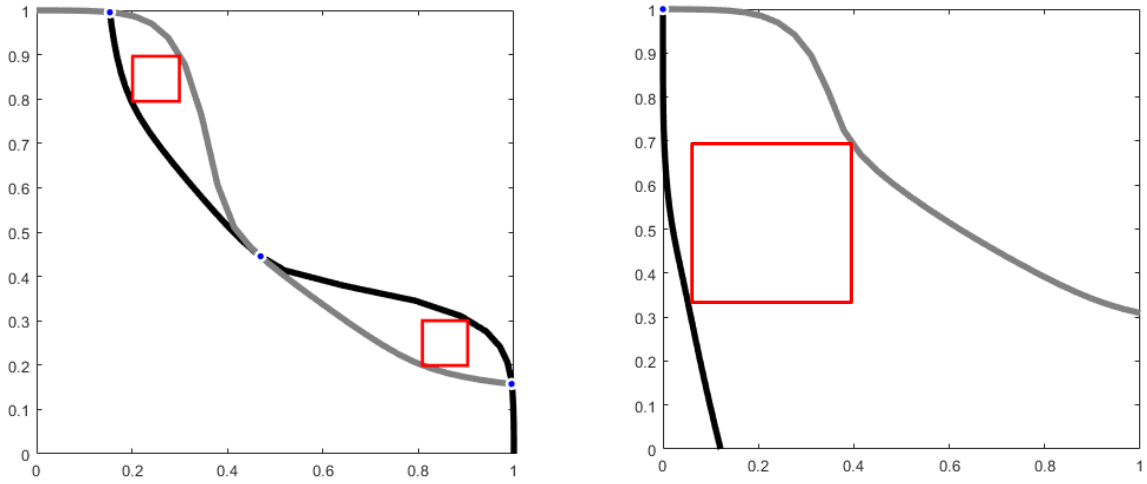


Figure 9. Left: RSNM of 6T SRAM. Right: WSNM of 6T SRAM.

As shown in Figure 10, the separated read path makes the circuit to measure 8T RSNM different from 6T. Data stored at V_2 node only applied to the gate of the read transistor, which has much less current flow compared with the 6T read mechanism. That allows the reading operation for 8T SRAM has more tolerance to the noise.

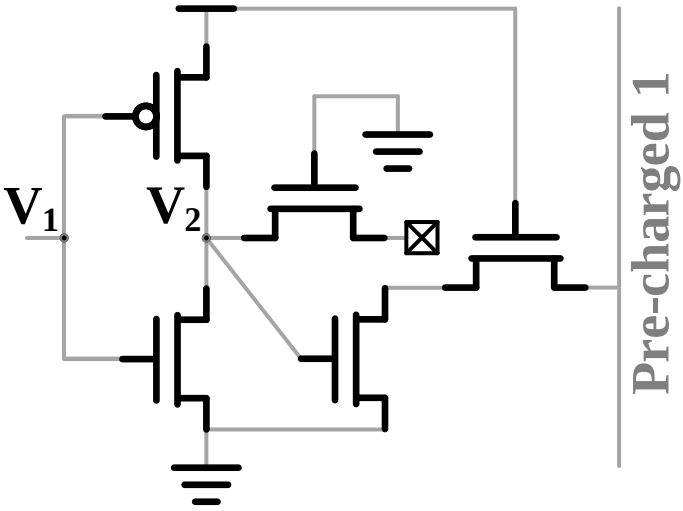


Figure 10. Circuit to measure Read Static Noise Margin for 8T SRAM.

Because of the writing operation of 8T SRAM is similar to 6T SRAM, the circuit to measure WSNM of 8T SRAM is the same as 6T only with different names of the wires.

SRNM and WSNM for 8T SRAM are shown in figure 11. It is worth noting that the RSNM is significantly improved because of the separated reading path. Thus, 8T SRAM is much more stable than the conventional 6T SRAM.

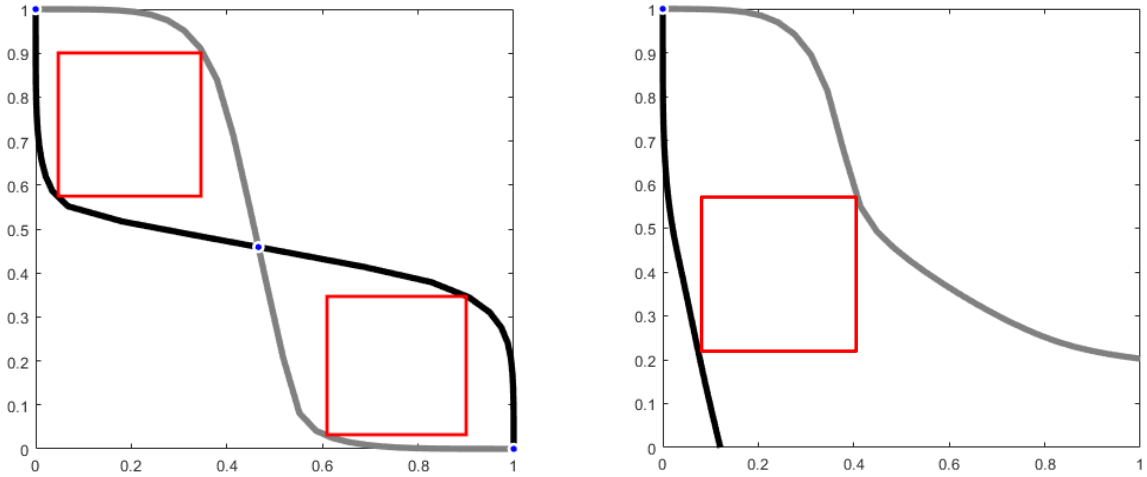


Figure 11. Left: RSNM of 8T SRAM. Right: WSNM of 8T SRAM.

To determine SNM values a mathematical model is established that calculates the values for the diagonals of the squares [36]. The (x, y) coordinate system is rotated 45° to a (u, v) coordinate system, and by taking the subtraction of two VTCs, the diagonal length is obtained. To transfer VTCs into the (u, v) system, the coordinate system is rotated as follow;

$$\vec{u} = \begin{pmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{pmatrix} \vec{x} \tag{1}$$

where x and $\bar{u}$ are the vectors (x, y) and (u, v) respectively. And the functions for u and v can be derived as follow;

$$\begin{aligned} u &= \frac{1}{\sqrt{2}}x - \frac{1}{\sqrt{2}}y \\ v &= \frac{1}{\sqrt{2}}x + \frac{1}{\sqrt{2}}y \end{aligned} \tag{2}$$

Figure 12 shows an SNM butterfly curve with a rotated (u, v) coordinate system. Two VTCs are rotated and shown in the upper part of Figure 12. In the ideal situation, the largest

squares fit in the two wings have the same diagonal length. But in the real world, there is always offset, so the smaller diagonal length is considered the value of SNM.

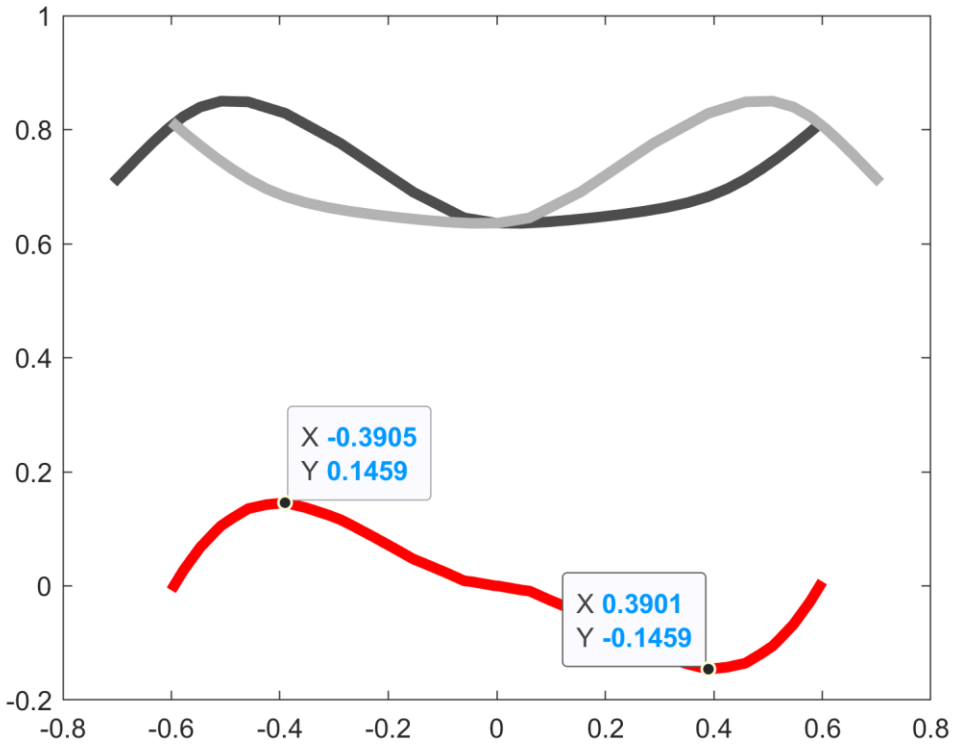


Figure 12. SNM estimation in a rotated coordinate system.

3.3. SRAM Failures

SRAM cell is a very busy circuit, surge currents flowing during reading and writing operations with magnetic field coupling, electric field coupling, and Gnd-Vdd upsets. Memory stability is being tested during operations. Memory failures can generally be divided into two categories: read failure and write failure. Read failure happens when the reading operation accidentally flips the states of a bitcell; write failure happens if a wrong date is stored to bitcell.

The primary cause of memory failures is process variations, in particular, threshold voltage variations (σV_{th}), which can be expressed as:

$$\sigma V_{th} = \frac{A_{VT}}{\sqrt{WL}} \tag{3}$$

where A_{VT} is a technology-dependent constant, W and L are the width and length of the transistor [19]. In the 45nm predictive technology σV_{th} for an NMOS and PMOS transistor with W equal to the minimum L_{EFF} (effective length) is 46.9mV and 41.8mV, respectively. Equation (3) clearly shows that σV_{th} is inversely proportional to $\sqrt{WL}$, which means as the W and L increase, the deviation of the threshold voltage is reduced.

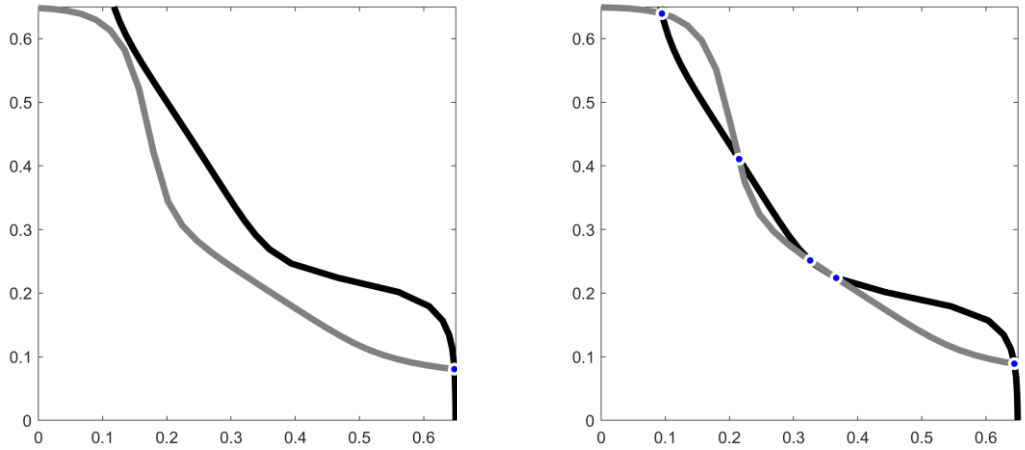


Figure 13. Examples of common read failures.

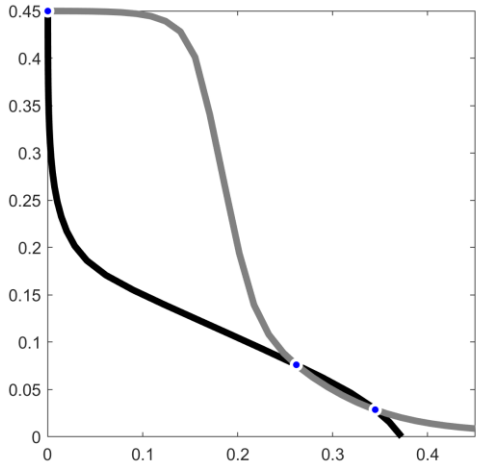


Figure 14. Example of write failure.

After injecting the variations into the SRAM circuit netlist, each transistor will be assigned a random variation. This then resulted in an imbalanced SRAM cell which directly

leads to memory failure. Figure 13 shows the two most common read failures; a write failure is shown in figure 14.

3.4. SRAM Failure Rate Simulation

In semiconductor manufacturing, process corners represent the extremes of fabrication parameter variations. An automatic complete flow is designed for SRAM failure rate simulation using the Monte-Carlo method. The inputs of this program include SRAM type (6T or 8T), SRAM size, supply voltage, Monte-Carlo simulation sample numbers, technology name (i.e. 45nm technology), and process corners. In this research, 1 million times of simulation are executed for each experiment.

Based on the physical characteristics for transistors, the above-mentioned circuits of measuring SRAM cell stability is implemented using netlist (a textual description of a circuit made of components). Then the meaningful data is extracted from the simulation output using a python script. Finally, by feeding the extracted data to a Matlab script, the failure rate of designed SRAM is acquired. The result of this study is then applied to all of the SRAM designs in this dissertation.

Figure 15 shows the read and write SNM curves for 6T respectively. While the voltage source is sweeping at node V_1 the pre-charged Vdd on the bit-line has a negative effect on V_2, and that leads to the V_2 does not drop to 0 when the voltage source is 1. This phenomenon significantly lowered the RSNM.

Five different process corner combinations are simulated to estimate the read and write failure rates, including “SS” (slow NMOS and slow PMOS), “SF” (slow NMOS and fast PMOS), “FS” (fast NMOS and slow PMOS), “FF” (fast NMOS and fast PMOS), and “TT” (typical NMOS typical PMOS). The simulation is performed on 6T bitcell and 8T bitcell with

3λ, λ is typically half of the minimum gate length. The designed program can automatically calculate the read and write failure rates using MATLAB. This program also automatically updates the voltage (Vdd) in the netlist after every 1 million trails, so we can get failure rates under different supply voltages.

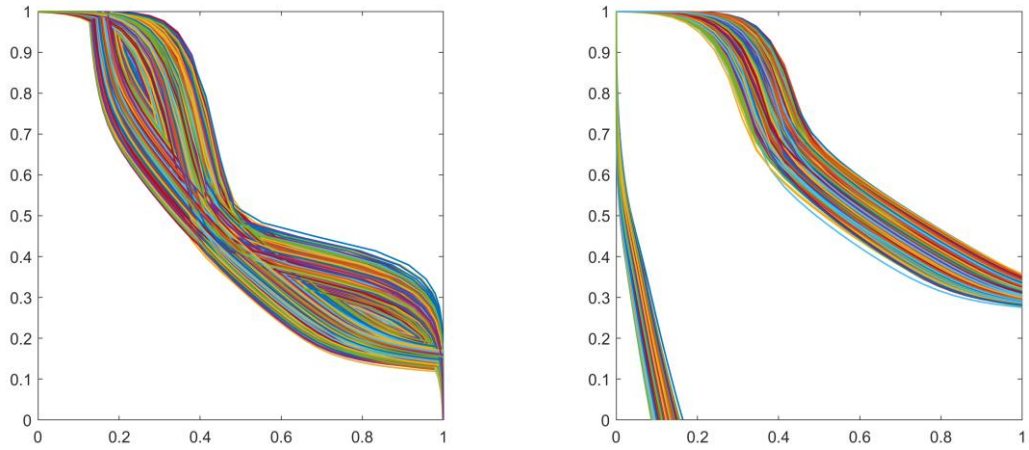


Figure 15. Read and write SNM curves for 6T.

The speed of transistors is another aspect needs to be considered, and that is where the process corner comes in. The faster NMOS is, the stronger pull-down networks are; the faster the PMOS is, the stronger pull-up networks are. To apply this principle to the SRAM design, faster NMOS speeds up the pull-down process, and during the reading operation, failures like the first graph shown in Figure 13 will happen. On the other hand, faster PMOS slows down the pull-down process, and during the writing operation, failures shown in Figure 14 will happen. The analyzed simulation result is shown below.

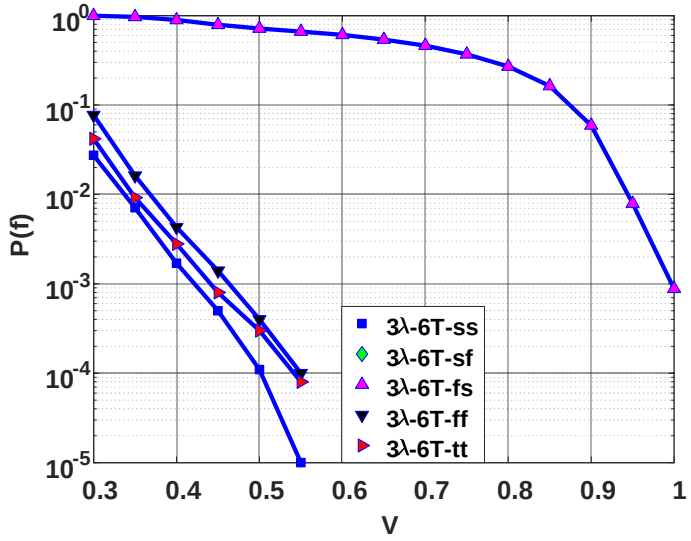


Figure 16. Minimal size 6T SRAM failure rates in different corner combinations.

As shown in Figure 16, 6T SRAM has a significantly worse failure rate at FS corner than the other corners. And as expected, the majority of the fails come from the reading operation. These failures, in the reading operation, are because both the bit-lines are pre-charged to Vdd before the access transistors are turned on, and if the two inverters in the SRAM cell are not "robust" enough, the charges in the bit-lines could disturb the stored data in *Q* and *QB*. Since the 8T SRAM has a separate reading path that significantly reduces the chance to flip the stored data during the reading operation, write failure dominates the overall total failures.

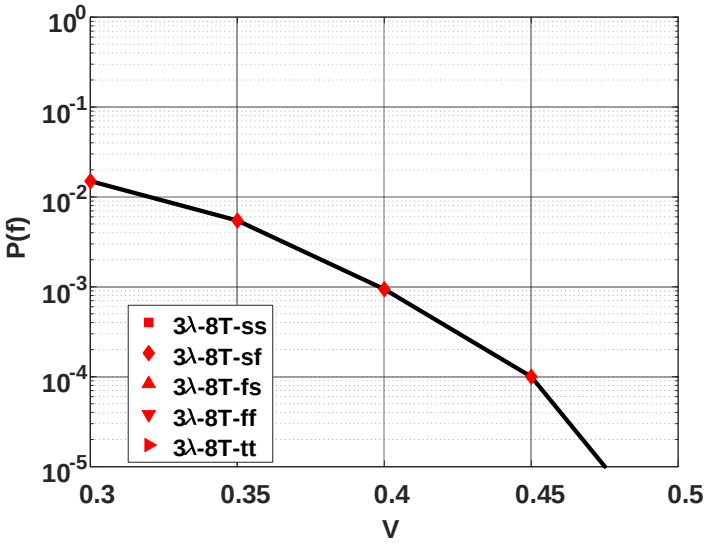


Figure 17. Minimal size 8T SRAM failure rates in different corner combinations.

Only the simulated result for the SF corner of 8T is shown in Figure 17 because no failure appeared at other corner combinations during the simulation. This result proves that the 8T SRAM cell structure is more robust than the 6T SRAM cell structure.

3.5. Conclusion

At this point, a method to automatically analyze the SRAM failure rate has been described. These studies on SRAM stability laid a solid foundation for the memory design. In particular, the failure rate associated with SRAM size and type can be plugged into the simulation to determine which SRAM cell should be used in different applications.

4. DATA PATTERN ENABLED SELF-RECOVERY VIDEO SRAM [46] [1]

Introduced previous works have shown low-voltage memory designs that enable power consumption reduction at the price of huge area overhead and design complexity. This work details a self-recovery video storage system created by mining correlated data patterns within the mobile video data. Both vertical and horizontal patterns are investigated by applying the data mining technique. a novel Data Pattern enabled Self-Recovery (DPSR) SRAM is designed using the discovered optimal patterns. With the implemented SRAM, we are able to deliver good output video quality at near-threshold voltage (0.5 V) operation with negligible area overhead (3.97%).

4.1. Near-Threshold Voltage Memory Failure Analysis

Recent manufacturing technologies indicate the failure probability of an SRAM cell to be between 0.1% and 1%, based on the bit-cell area in [39, 40]. To achieve the 0.1% failure rate, the size of the SRAM cell will have 58% area overhead [40]. Both the SRAM cells with minimum-size (failure rate 10^{-2}) and upsize (failure rate 10^{-3}) are analyzed in this work. In Table 1, the probabilities of multiple faults that occur within the same word-line are listed. It is clear that there are only very few bits will fail at the same time in a word-line. This result supports the idea of using other bits in the same word to recover the faulty bit.

[1] The material in this chapter was co-authored by Yifu Gong, Dongliang Chen, and Jonathon Edstrom. Yifu Gong and Dongliang Chen held primary responsibility for SRAM hardware design and verifiecation. The data association and correlation were investigated by Jonathon Edstrom.

Table 1. Fault Probability in a 32-bit SRAM Word

Number of faults per word-line	SRAM failure rate: 10^{-3} (0.001)	SRAM failure rate: 10^{-2} (0.01)
0	96.8523477%	72.7279953%
1	3.0992274%	23.2812509%
2	0.0479198%	3.6012385%
3	0.0005023%	0.3611914%
4	0.0000028%	0.0267011%
5	0%	0.0015432%
6	0%	0.0000756%
7	0%	0.000004%

4.2. Self-Recovery Data Pattern Investigation

This section introduced the methodology for finding the hidden data-patterns in the video data to allow effective fault recovery. In particular, a new two-dimensional data pattern approach for self-correction techniques is proposed to explore both horizontal and vertical data characteristics.

4.2.1. Horizontal Association Rule Mining

YUV format is a typical format to store and process mobile video data. The YUV format includes one *luma* (*Y*) component and two *chroma* components. Brightness information of the image is stored in *luma*, and *chroma* contains the blue-difference (*Cb*) and red-difference (*Cr*) color information. An example of how YUV 4:2:0 video data stored in the memory is shown in Figure 18. 8-bits of luma data and 8-bits of subsampled chroma data are assigned to each pixel.

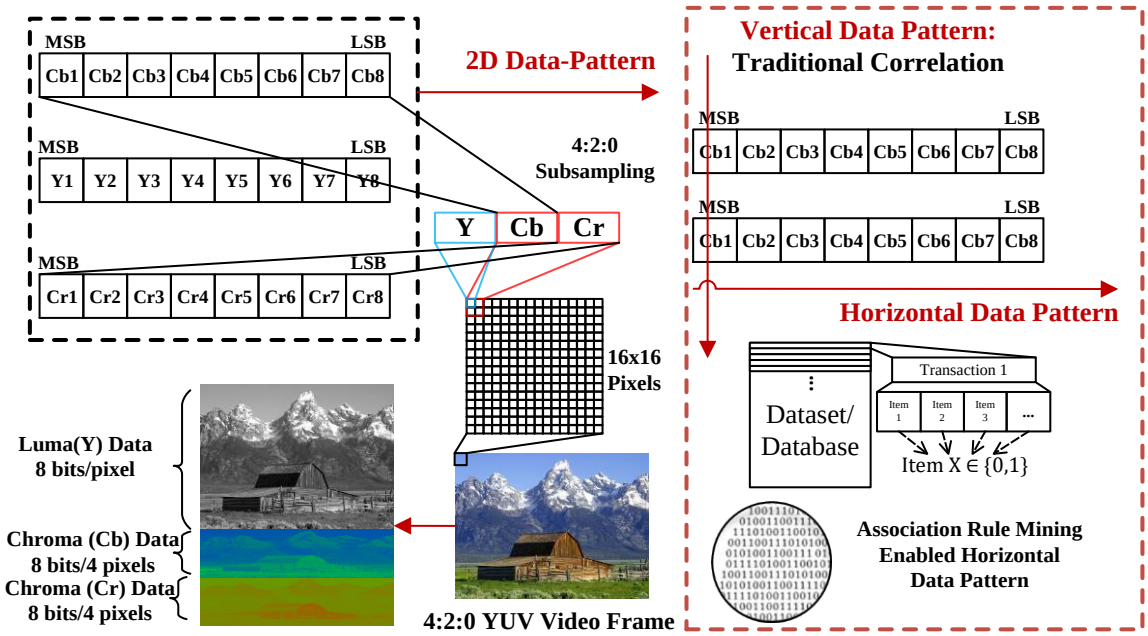


Figure 18. 2D data-pattern enabled self-correction.

4.2.2. Vertical Correlation Rule Mining

The MSBs of pixel data are strongly correlated with the adjacent pixels and have a very low probability of switching. The MSB vertical association probability in adjacent pixels reaches 93%, while the LSB has a decline to 53%.

To select the optimized data-pattern from the explored horizontal associations and vertical correlations, Weighted Confidence is defined as follows:

$$\begin{aligned} \textit{Weighted Confidence} = {} & \textit{Confidence(Rule)} \times \textit{Support(Rule)} \\ & + \textit{Confidence(Complement Rule)} \times \textit{Support(Complement Rule)} \end{aligned} \tag{4}$$

Then this value is compared to the sum of the correlation values for 0 and 1, which we define the correlation value as follows:

$$\begin{aligned} \textit{Correlation} = {} & \textit{Confidence}(Bit_{previous} = 0 \rightarrow Bit_{current} = 0) \\ & + \textit{Confidence}(Bit_{previous} = 1 \rightarrow Bit_{current} = 1) \end{aligned} \tag{5}$$

By applying the weighted confidence and correlation calculation above, we obtain the optimized data patterns with the highest prediction probability to achieve self-recovery, as shown in Table 2

for *Luma* and Table 3 for *Chroma*. The result of our analysis shows that *luma* data has less association within the same pixel and the optimal data patterns are all from correlation.

Table 2. Optimal Luma Data Patterns

Y bits	Optimal Data Patterns	Correct Prediction (%)
Y1	Correlation ($Y1_{previous}$)	91.5290
Y2	Correlation ($Y2_{previous}$)	82.6719
Y3	Correlation ($Y3_{previous}$)	76.2655
Y4	Correlation ($Y4_{previous}$)	67.6406
Y5	Correlation ($Y5_{previous}$)	59.2428
Y6	Correlation ($Y6_{previous}$)	51.7514
Y7	Correlation ($Y7_{previous}$)	44.4694
Y8	Correlation ($Y8_{previous}$)	38.4120

Table 3. Optimal Chroma Data Patterns

Cb bits	Optimal Data Patterns	Correct Prediction (%)	Cr bits	Optimal Data Patterns	Correct Prediction (%)
Cb1	Association $(\overline{Cb2} \rightarrow Cb1)$	98.5965	Cr1	Association $(\overline{Cr2} \rightarrow Cr1)$	96.7237
Cb2	Association $(\overline{Cb1} \rightarrow Cb2)$	99.7935	Cr2	Association $(\overline{Cr1} \rightarrow Cr2)$	97.7735
Cb3	Correlation ($Cb3_{previous}$)	88.4593	Cr3	Association $(\overline{Cr1} \rightarrow Cr2)$	93.8576
Cb4	Correlation ($Cb4_{previous}$)	84.3113	Cr4	Correlation ($Cr4_{previous}$)	83.6360
Cb5	Correlation ($Cb5_{previous}$)	78.5307	Cr5	Correlation ($Cr5_{previous}$)	78.3486
Cb6	Correlation ($Cb6_{previous}$)	69.3991	Cr6	Correlation ($Cr6_{previous}$)	68.8025
Cb7	Correlation ($Cb7_{previous}$)	59.3976	Cr7	Correlation ($Cr7_{previous}$)	59.7336
Cb8	Correlation ($Cb8_{previous}$)	51.1264	Cr8	Correlation ($Cr8_{previous}$)	52.9571

4.3. DPSR Hardware Implementation

Figure 19 shows the schematic of the proposed DPSR, where four 256 words × 32 bits blocks form a 32 kbits array. Both *luma* and *chroma* data are stored in different blocks of the SRAM in this design as shown in Figure 20.

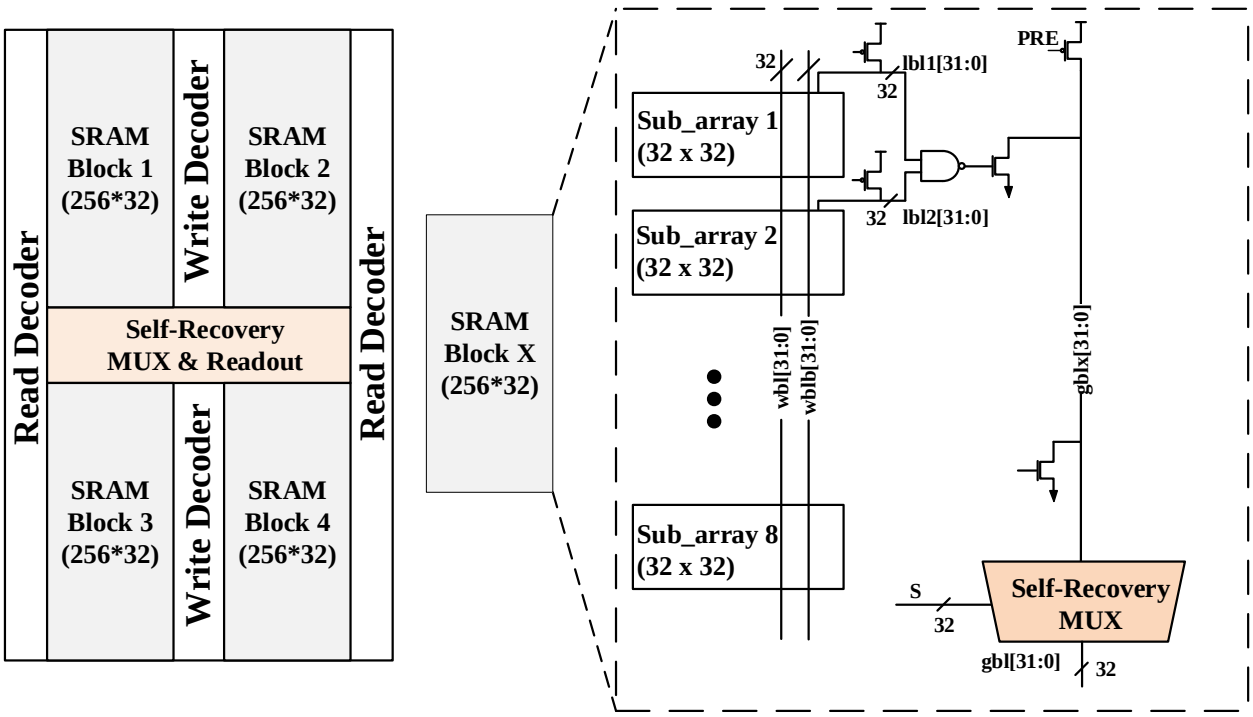


Figure 19. Proposed DPSR.

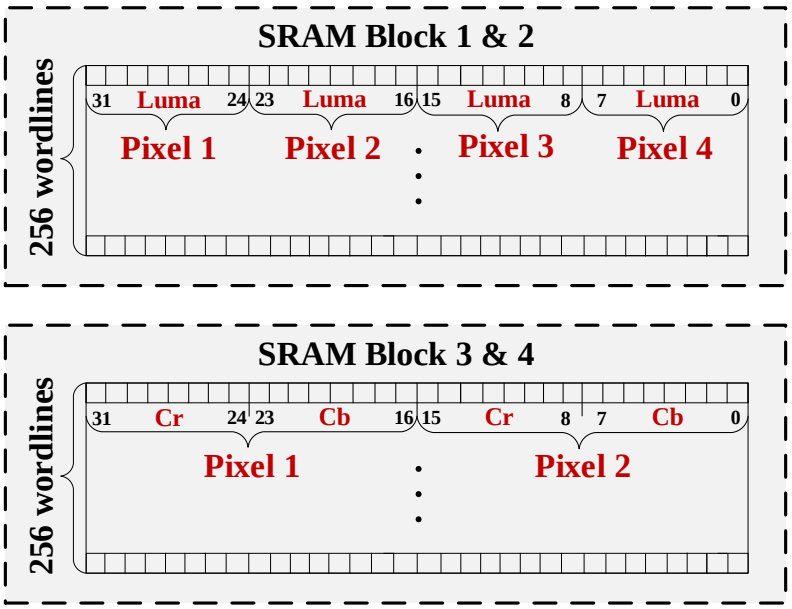


Figure 20. Luma and Chroma data distribution.

To reduce the access time, a hierarchical readout bit-line scheme (local RBL and global RBL) is applied. Multiplexers (MUX) are connected at the global bit-lines (*gbl*) of conventional SRAM to implement the self-recovery logic of DPSR. Figure 21 gives an idea of how each global bit-line (*gbl*) is connected to a multiplexer controlled by the received fault positions. When a fault is detected, the select signal (*S*) for according bit-line will be enabled to correct the faulty bit.

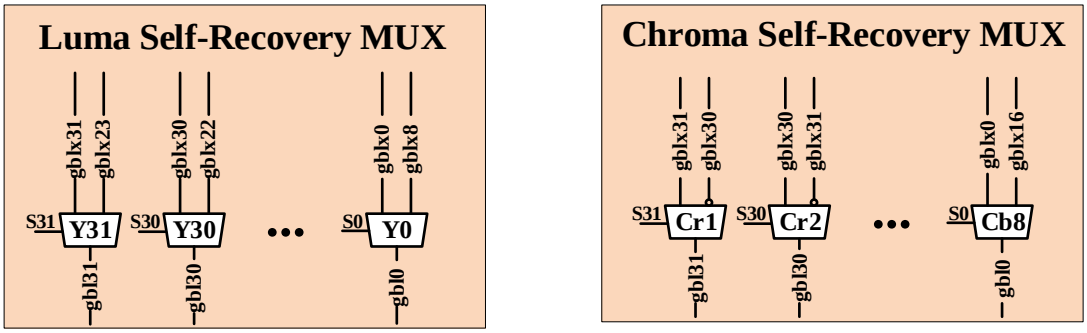


Figure 21. Self-Recovery MUX connection.

Based on the optimal luma patterns and luma data storage of a word, luma data of adjacent pixels will be used to recover the data of the current pixel. For example, Y1 of pixel 1 (Luma[31]) will use Y1 of pixel 2 (Luma[23]) for self-recovery. Chroma data self-recovery is realized for SRAM block 3 and block 4 by applying the optimal chroma patterns.

4.4. Evaluation Methodology and Results

A 32 kb SRAM is implemented using a high-performance 45-nm FreePDK CMOS process to evaluate the effectiveness of the proposed technique.

4.4.1. Performance

The performance of the proposed DPSR is simulated. Because of the inserted multiplexers, DPSR has an increased access time at 0.31 ns compare to the conventional SRAM at 0.27 ns but still will be fast enough to stream high-quality video format.

4.4.2. Layout

As discussed in the introduction, a large portion of the area in a video chip is occupied by embedded SRAMs, thus the SRAM area is an important design concern. The layout of DPSR is shown in Figure. 22. The size of each added self-recovery logic (MUX) is 18.79 µm × 43.47 µm, which leads to a 7.94% overall area overhead.

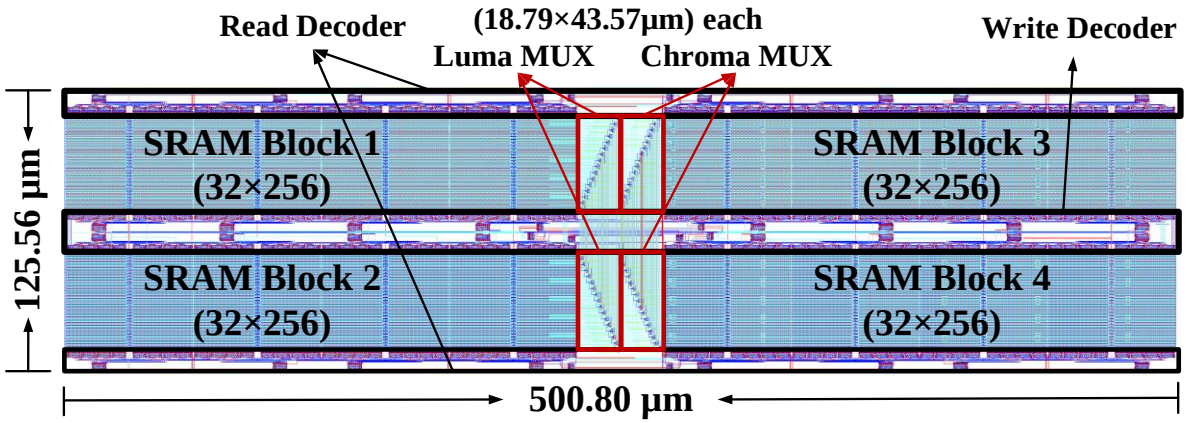


Figure 22. Proposed DPSR.

4.4.3. Video Output Quality Analysis

The PSNR metric is adopted to measure the video quality, which is defined as:

$$PSNR = 10 \cdot log_{10}\left(\frac{255^2}{MSE}\right) \tag{6}$$

where MSE is the mean square error between the original videos (Org) and the degraded videos (Deg), expressed as:

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[Org(i,j) - Deg(i,j)]^2 \tag{7}$$

Researchers have shown that when the PSNR is higher than 30 dB, the video quality would be acceptable [39]. Table 4 compares PSNR values using conventional and proposed design as P_{fail} are 10^{-2} and 10^{-3}. Ten videos from the 25 videos used for verification are measured

and presented in Table 4. It can be seen that the PSNR is over 35 dB even for the highest SRAM failure rate, which proves the DPSR has efficient recovery precision.

Table 4. Video PSNR Metric Comparison

Video Label	**conventional (P_{fail} = 0.001)**	**DPSR (P_{fail} = 0.001)**	**conventional (P_{fail} = 0.01)**	**DPSR (P_{fail} = 0.01)**
Running	34.843802	**47.751093**	27.751663	**37.896356**
Concert	34.843123	**50.617823**	24.745933	**39.835772**
Music Video	34.842942	**48.993861**	24.765553	**37.908828**
Festival	34.843240	**45.838237**	24.892104	**35.958557**
Game	34.843259	**49.286247**	24.759353	**39.699233**
Electric Guitar	34.843014	**51.566521**	24.752845	**42.584377**
Snow	34.844445	**50.725480**	24.761392	**40.861991**
Flute	34.842227	**53.769387**	24.755972	**44.158630**
Vehicle	34.843032	**50.015065**	24.741031	**42.251862**
Planet	34.843295	**53.306924**	24.760113	**44.022668**

The performance of DPSR is compared with the state-of-the-art in Table 5. By applying the data-pattern enabled self-recovery technique, the proposed DPSR achieves reliable operation at near-threshold voltage to enable energy saving with low area overhead (7.94%). The data-shifting technique detailed in [41] has slightly better PSNR measurement but it is realized with large area cost (~14%). A squeezing technique is presented in [39] to compress zeros and store them in less memory space, thereby avoiding the presented memory failures at low voltage. But it was achieved with an extra clock cycle. Thus, DPSR delivers the best video quality for the minimum area overhead.

Table 5. Comparison with Prior Work

	TCASI'12 [15]	*DAC'15 [41]*	*TC'16 [39]*	***This Work***
fault-position awareness	No	Yes	Yes	**Yes**
Low-power techniques	bitcell sizing	data-shifting	data-squeezing	**data-pattern enabled self-recovery**
bitcell modified	Yes	No	No	**No**
near-threshold operation	No (0.9V)	Yes (-)	Yes (0.5 V)	**Yes (0.5V)**
additional logic needed	No	LUTs and shifter	Rearrangement logic and tag array, comparator, Mux	**MUX**
performance overhead	-	-	extra clock (for decompression)	**0.04 ns**
video quality	acceptable	good	-	**good**
area overhead	11-65%	14%	6.3%	**7.94%**

4.5. Conclusion

A data-pattern enabled self-recovery video SRAM is presented in this section. By using data-mining techniques, data association within a pixel and correlation with adjacent pixels are investigated. DPSR SRAM is designed to enable the found data-pattern with low area overhead (7.94%). The proposed design reduces 81.52% dynamic power consumption and 82.45% leakage power consumption as compared to nominal voltage operations. The simulation shows our design is able to deliver good video quality for minimized SRAM at near-threshold voltage.

5. NEURAL NETWORK SYNAPTIC STORAGE DESIGN [45] [2]

The neural network has broad application prospects such as image recognition, pattern discovery, and autonomous control. In this subsection, the effect of memory failures for Artificial Neural Network (ANN) is analyzed, and a neural network synaptic storage is design using obtained data.

ANN consists of many intricately connected bionic pathways (synapses) that connect computational units in between an input and output layer. Each computational unit, or neuron, can pass a signal from the input layer, through multiple hidden layers, and eventually to the output layer. The signals at each edge (the connection between artificial neurons) are adjusted by the weights. The weights of neural networks access memory repeatedly throughout the training process.

In order to acquire the largest failure rate that the memory could endure for ~1% degradation, a wide range of bitcell failure rates are applied for testing on MNIST [20], a widely used digit recognition dataset. We found that ***to allow less than 1% accuracy degradation, a maximum memory failure rate of 10-5 is required***.

5.1. SRAM Bitcell Design

As we know, the 8T SRAM cell is more reliable but bigger than 6T SRAM, and another fact is that the failure rate reduces as the SRAM size increases. The minimal 3λ-8T SRAM bitcell occupies a similar area to 9λ-6T SRAM bitcell. The layout of upsized 9λ-6T bitcell and

[2] The material in this chapter was co-authored by Yifu Gong, Dongliang Chen, and Jonathon Edstrom. Yifu Gong held the primary responsibilities of circuit design and hardware implementation. Dongliang was in charge of hardware simulation and verification. Jonathon Edstrom performed the software simulation.

3λ-8T bitcell are shown in Figure 23. Two bitcells are designed to have the same height (0.4465 μm) and a similar width (about 1.75 μm).

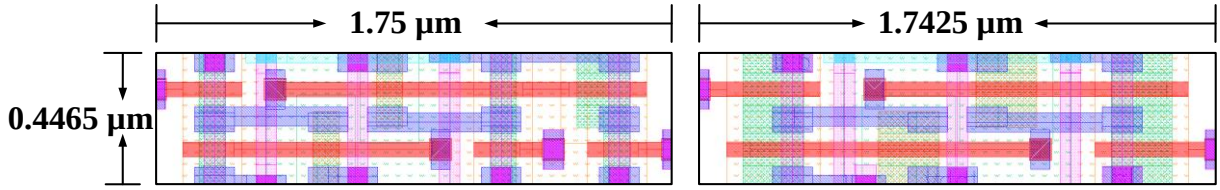


Figure 23. Left: Upsized 9λ-6T bitcell; right: 3λ-8T bitcell.

To decide the size and type of SRAM design, failure rate simulation is performed using the program mentioned in section 2.1. 6T SRAM with variant sizes and an 8T SRAM with minimal size (all in worst corner combination) are simulated, and the obtained failure rates are analyzed and shown in Figure 5. According to Figure 5, to achieve a maximum memory failure rate of 10-5 at 1.0V, 4λ-6T sizing is selected as baseline synaptic memory. Figure 24 also shows that with a similar layout area, the failure rate of 3λ-8T bitcell is significantly lower than 9λ-6T bitcell at the same voltage.

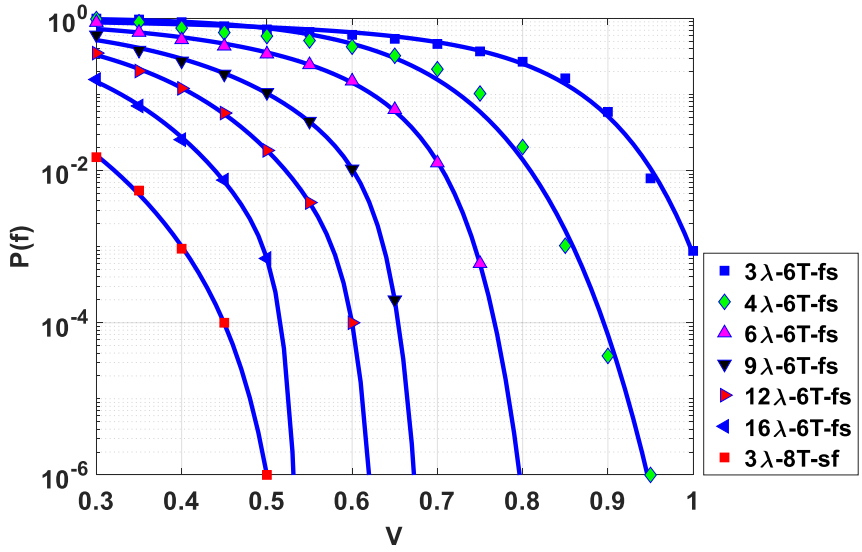


Figure 24. 45nm upsized 6T and 8T SRAM bitcell failure rates in worst corner combination based on Vdd voltage scaling.

5.2. Implementation

During the analysis of data characteristics, we find that the 8 MSBs have a significant impact on the test accuracy in all weight layers. And by introducing offline association rule mining techniques, we further analyze the data association/correlation relationship between the first 8 MSBs.

Association rules	Conf.	Supp.
w30 = 0 → w29 = 1	99.95%	99.95%
w30 = 0 → w28 = 1	99.95%	99.95%
w30 = 0 → w27 = 1	99.89%	99.89%
w30 = 0 → w26 = 1	86.45%	86.45%

Association rules: the percentage numbers indicate the probability of next bit

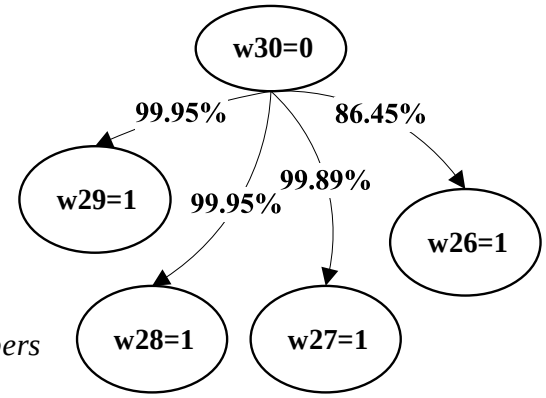


Figure 25. Offline data-mining data relationships.

Figure 25 shows that if the value of W_{30} is *0*, W_{29}-W_{26} bits have a much higher chance to be *1*. Thus, if W_{30} is stored in reliable memory bitcells, when W_{29}-W_{26} bits fail, SRAM may achieve self-recovery based on obtained data association/correlation rules. While the worst-case precision accuracy loss is at 1%, the supply voltage can be scaled down to 0.825V with the proposed self-recovery technique.

Similar to the technique introduced in Chapter 4, the architecture of the proposed synaptic memory with 512 words × 128 bits is shown in Figure 26. To reduce the access time, a hierarchical bit-line schematic (local RBL and global RBL) is designed.

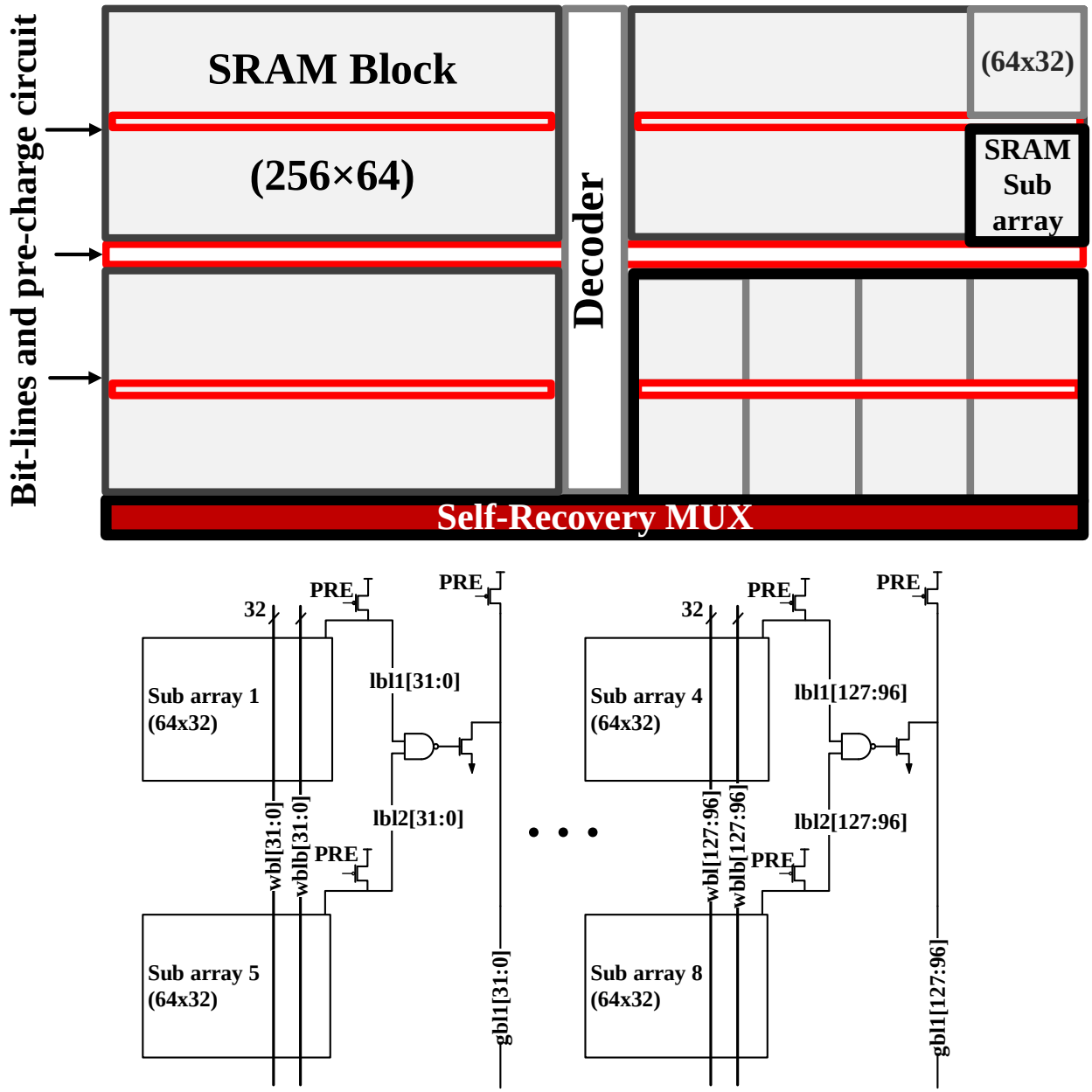


Figure 26. Data-driven efficient synaptic storage.

As shown in Figure 27, two 8T bitcells are applied to store the sign bit (W_{31}) and W_{30} in each 32-bit synaptic weight because of the importance of these bits. A multiplexer-based schematic is adapted in our design to implement a self-recovery synaptic memory. With the obtained data association/correlation relationship, the memory global bit-lines are connected to a self-recovery multiplexer (MUX) accordingly as shown in Figure 28. Pre-detected faulty bit

locations, identified either during post-fabrication testing or during power-on self-test (POST), generate select signals to control the MUX.

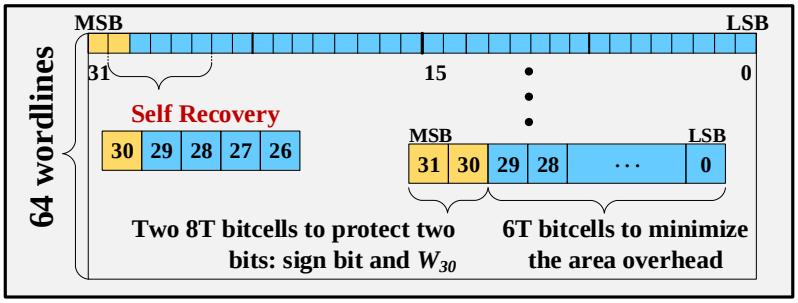


Figure 27. 6T and 8T bit-cell arrangement.

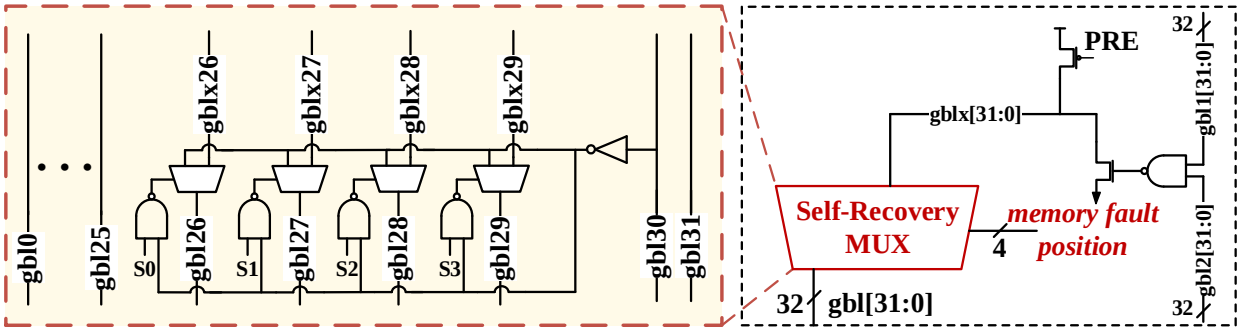


Figure 28. Self-recovery MUX connections.

To assess the performance of the proposed Data-driven storage design, a synaptic memory is implemented based on the schematic shown in Figure 26. As mentioned earlier, a large portion of the silicon area on a deep learning chip is occupied by SRAM, so reducing the area cost of embedded SRAM is a major design problem. Figure 29 shows the layout design. With careful design, the added self-recovery circuits (MUX) only introduces 3.17% area overhead. The parasitic parameters are extracted and included in the power and timing simulation.

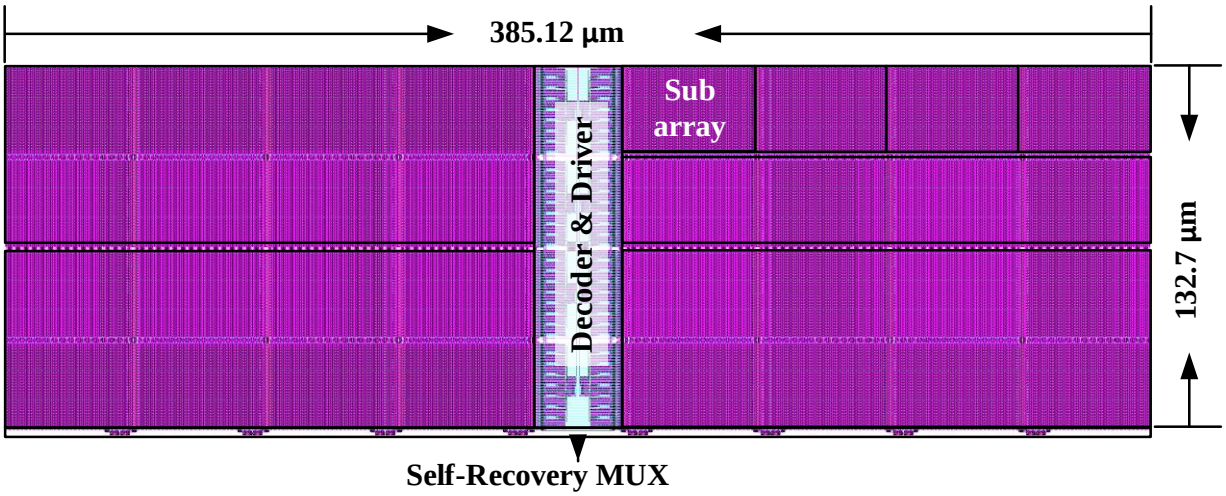


Figure 29. Layout of the proposed memory in a 45 nm technology.

5.3. Evaluation Methodology and Results

As shown in Table 6, the read access time of proposed memory (with MUX) increases from 1.154 ns to 1.415 ns because of the reduced supply voltage. Based on the read access time, the maximum frequency of our proposed design is 706.7 MHz, which meets the requirement of neural network weight update speed.

Table 6. Read and Write Delay Times

	1.0V		*0.825V*	
Scheme	*Write (ns)*	*Read (ns)*	*Write (ns)*	*Read (ns)*
All 6T	0.532	0.779	0.576	0.974
Hybrid w/o MUX	0.532	0.941	0.576	1.121
Hybrid w/ MUX	0.532	1.154	0.576	1.415

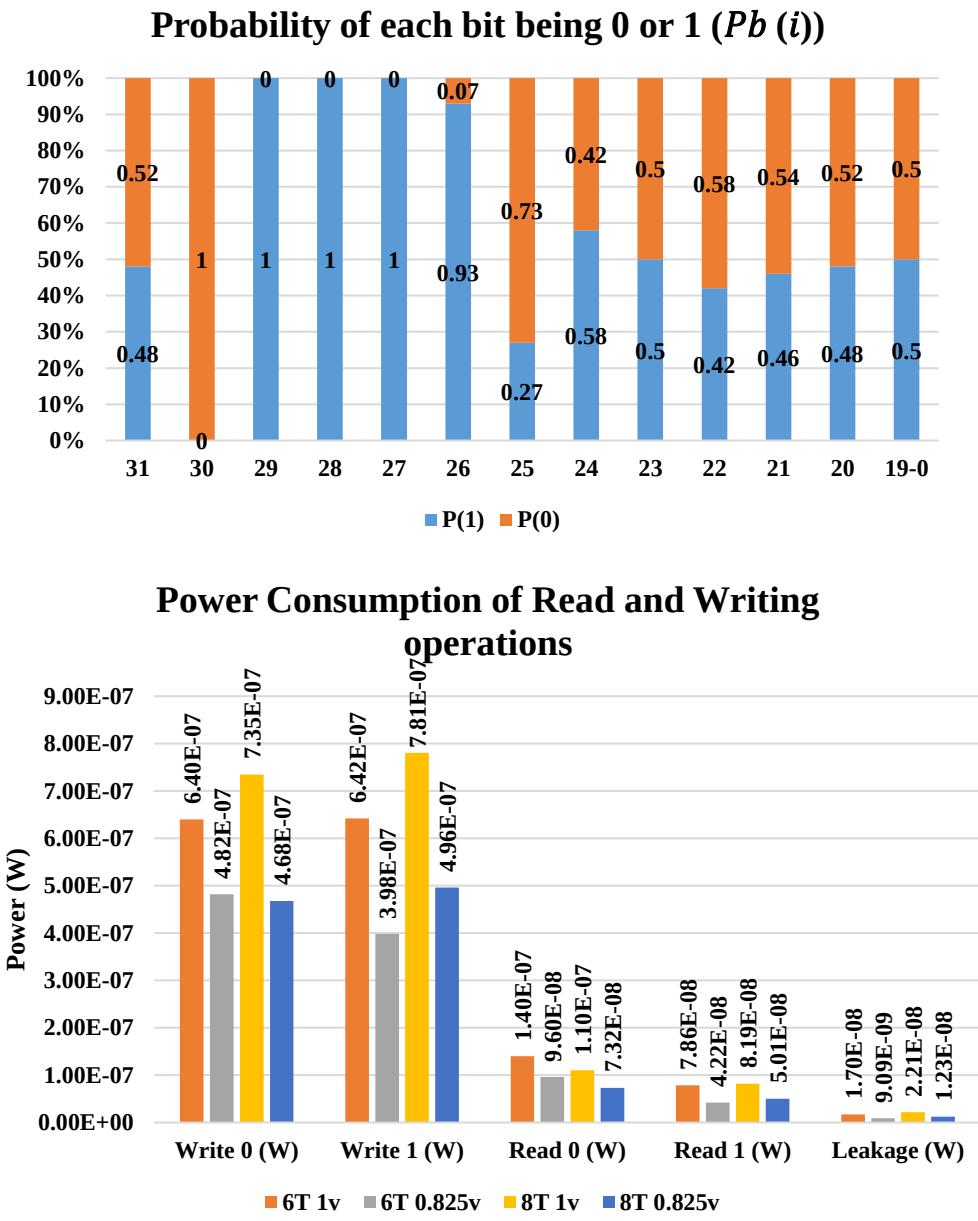


Figure 30. Bits probability and operation power consumptions.

To get a more realistic power assumption, the data switching probability of each bit is considered and power consumptions of reading, writing, and leakage are calculated, and values can be seen in Figure 30. We model the read bit-line (RBL) power consumption of each word in the memory as:

$$P_{Active} = \frac{\sum_{b=0}^{31} \sum_{i=0,1} \left[P_b(i) \cdot \left(R(i) + W(i) \right) \right]}{2}$$

$$P_{Leak} = \sum_{d=0}^{255} \sum_{j=0}^{31} L(j) \tag{8}$$

where P_{total} is the power consumption of both the read and the writing operations; *b* is the bit number; *i* is the value stored in SRAM; *P(i)* is the probability of a particular bit to be 0 or 1, which is shown in Table V; *R(i)*, *W(i)*, and *L(j)* are the read, write, and leakage power consumption, respectively, based on the value *i* and *j*. The bit value probabilities are extracted based on a 2 hidden layer MNIST neural network with 100 hidden nodes per hidden layer.

Based on Equation (8), the conventional SRAM consumes 154.5 μW active power and 138.9 μW leakage power at 1V, while our proposed design at 0.825V consumes 106.1 μW active power and 75.83 μW leakage power, enabling 45.6% and 83.1% savings in active power and leakage power, respectively.

Table 7. Comparison of Techniques.

Memory Techniques		*Avg. Accuracy*	*Average Loss*	*Area Overhead*
Traditional @1V	All 6T	96.121%	0%	0%
Traditional @0.825V	All 6T	9.8%	86.321%	0%
DATE'16 [22] @0.825V	2 MSBs 8T	92.993%	3.128%	1.606%
	3 MSBs 8T	93.120%	3.001%	2.409%
	4 MSBs 8T	94.369%	1.752%	3.212%
	5 MSBs 8T	94.436 %	1.685%	4.015%
This Work @0.825V	***2 MSBs 8T + correction***	***95.401%***	***0.72%***	***3.171%***

The proposed Data-driven memory is compared with traditional memory and the recently developed 8T-6T hybrid synaptic memory [22]. In order to make a valid comparison, all memories are simulated at the same voltages (1V and 0.825V). Classification accuracy for varying numbers of 8Ts in [22] and proposed Data-driven technique is evaluated based on 30

independent trials using the MNIST benchmark. The results are compared against the ideal, fault-free system, which is listed in Table 1. As shown in Table 7, the conventional 6T SRAM has a significant classification accuracy degradation (86.321% loss) at 0.825V. In terms of [22], the more 8T cells are inserted, the network is more accurate. With 5 MSBs stored in 8T, for instance, the average loss is reduced to 1.685% with 4.015% area overhead. It can be observed, our proposed Data-driven technique introduces a lower implementation cost (3.171% area overhead) with a better classification accuracy (95.401%) at 0.825 V.

5.4. Conclusion

In this section, a Data-driven self-correction design for neural network synaptic memory is presented. The proposed memory, as compared to traditional memory, enables 45.6% and 83.1% in active power and leakage power savings, respectively; it also achieves less than 1% degradation in classification accuracy with only 3.17% area overhead.

6. VIEWING CONTEXT-AWARE SRAM DESIGN [44] [3]

In today's mobile video system, embedded memory is the critical component increasingly leading power consumption of mobile devices. This chapter presents a Viewing Context-Aware SRAM (VCAS) that enables power saving with different viewing surroundings. The brighter the surrounding is, the less sensitive human eyes are on detecting video quality changes. This is because the contrast resolution of human eyes is limited [23]–[26]. Thus, when eyes adapt to a brighter environment, they will lose contrast resolution.

To improve memory power efficiency while considering the viewing context of the user, a VCAS is designed and simulated. The proposed VCAS is implemented as a reference framebuffer for H.264, a popular video codec standard in mobile multimedia communications. Within a frame, each pixel has 8-bit *luma* data and 4-bit Chroma data. And for each frame, the memory consecutively stores all the *luma* (Y) data and followed by the Chroma data – Cb (U) and Cr (V). The decoding process requires frequent write and reading operations in memory, which consumes significant power. Thus, reducing power consumption in mobile video memory is worthy of study.

6.1. Enabling VCAS by Introducing Hardware Noise

To decide which low-power memory design should be used (voltage-scaling or bit-truncation), a simple experiment is performed. The output video quality of those two techniques is compared in Figure 31. using Akiyo, a common testing video used in video processing papers, as an example. As shown, the bit-truncation introduces some blur in the video with the *PSNR*

[3] Yifu Gong held the primary responsibilities of circuit design and hardware implementation. This experiment is designed and tested by Yifu Gong and Peng Gao. Dongliang was in charge of hardware simulation and verification. Jonathon Edstrom performed the software simulation. Power saving is calculated by Dongliang Chen.

value at 31.67. As a comparison, the voltage scaling technique causes noise points in the video with the PSNR value at 32.61. Two output results achieve similar *PSNR* values, but the video quality loss with the bit-truncation approach is inconspicuous compared to the voltage scaling approach. Thus, the bit-truncation technique is employed as the approach to enable power-quality tradeoff in different viewing contexts.

(a) (b)

Figure 31. Video output with bit-truncation and voltage scaling. (a) Original video PSNR = 38.83. (b) Bit-truncation PSNR = 31.67.

6.2. VCAS Design Using Bit-truncation Technique

In order to determine the number of bits to truncate, user experience is evaluated in different viewing contexts. 50 people participated in our video tests. During the test, participants watch the original video followed by a random truncated video that has 1 to 7 bits LSBs truncated in three viewing contexts. The process repeats until 7 trials are completed.

Table 8. VCAS Bit-truncation Implementation.

context	***In dark***	***In overcast***	***In sunlight***
Luminance (lux)	0-1000	1000-10000	10000+
Video data bits[1]	xxxxxxxx	xxxxx000	xxxx0000

[1]*x* means original video data and *0* means truncated bit

The decision is shown in Table 8. In this table, x means the original video data, and 0 means using bit-truncation. As shown in overcast, when the VCAS stores data with 3 LSBs truncated, there is no significant degradation of video quality; in sunlight, 4 LSBs truncated can

produce nearly the same perceived quality as the original video due to the strong interference effect of high-ambient luminance. It should be pointed out that the average scores of videos with 4 LSBs truncated in overcast and 5 LSB truncated in sunlight are both below 3.0, indicating that even with ambient luminance interference, the significant degradation of video quality can be easily noticed.

6.3. Hardware Design

To store Luma and Chroma data separately, two SRAMs are used. VCAS is implemented storing Luma data. Chroma data is stored in conventional SRAM. The information on the ambient luminance level can be captured from the light sensor embedded in mobile phones and sent to the VCAS for adaption.

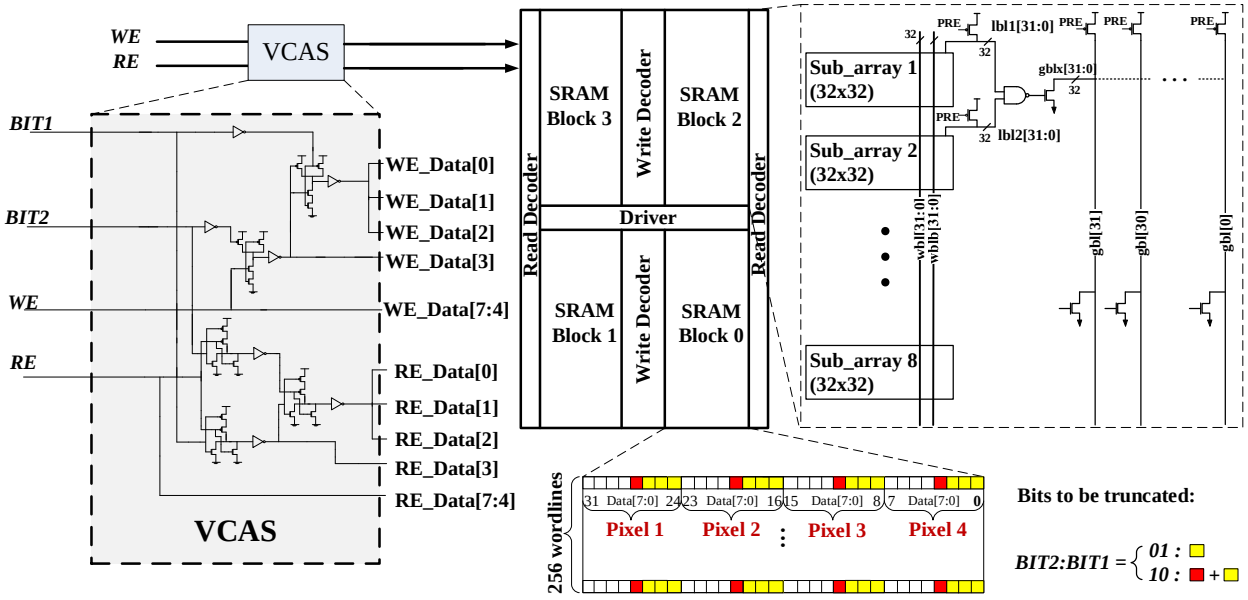
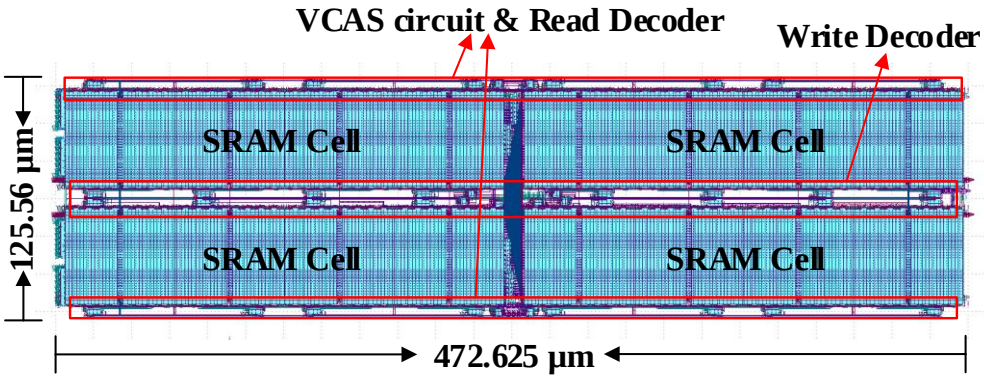


Figure 32. SRAM with VCAS control circuit.

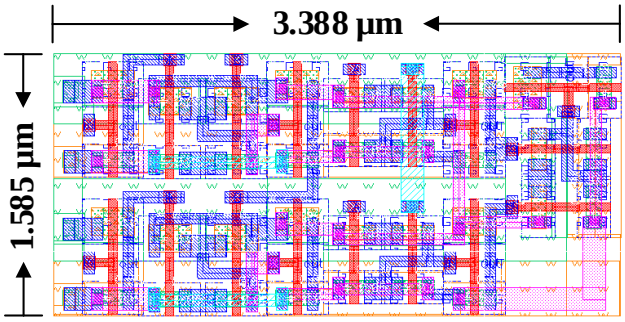
The proposed VCAS circuit is shown in Figure 10. The total size of the memory array is 32 kbit and there are four blocks with 32 bits of 256 words. To reduce access time, a hierarchical bit-line scheme (local RBL and global RBL) is applied. The VCAS control unit consists of a Write Enable (*WE*) control circuit and Read Enable (*RE*) control circuit. Figure 32. Shows the

detailed control circuits for *WE* and *RE*. The memory disables the *WE* and *RE* signals of *0*, *3* or *4* LSBs using the control unit according to the context information bits detected by the mobile sensor. The two-bit control signal (*BIT2* and *BIT1*) for the *WE* (or *RE*) control circuit will enable bit-truncation in the following ways:

{BIT2 BIT1} =
- 00: in dark, use 8-bit original video data
- 01: in overcast, set 3 LSB data WE (or RE) signal to 0s
- 10: in sunlight, set 4 LSB data WE (or RE) to 0s



(a) Layout of VCAS SRAM (472.625 µm × 125.56 µm)



(b) Layout of VCAS control circuit

Figure 33. Proposed layout.

Figure 33 (a) shows the layout design of the VCAS and its control circuit. It is designed with the FreePDK45 library package. The layout passes the Layout Versus Schematic (LVS) and the Design Rule Check (DRC) to prove it is realizable. It can be seen that, since only several gates are added into the conventional SRAM memory, as shown in Figure 33 (b), the area overhead is negligible (<0.01%).

6.4. Simulation Results

First, the performance is simulated to make ensure that the typical mobile videos are supported on VCAS. The simulation results in sunlight, overcast, and dark contexts are shown in Figure. 4. The data (*0x97, 0xf3, 0xc6, 0x0e*) is written to the address (*0x0a, 0x1a, 0x25, 0x3b*) and then read out from the same address. Figure 34 also shows the output data for dark condition, (*0x97, 0xf3, 0xc6, 0x0e*), with no truncation applied to original data; in overcast condition, the output is original data with 3 LSBs truncated, (*0x90, 0xf0, 0xc0, 0x08*); in dark conditions, the output is input data with 4 LSBs truncated, (*0x90, 0xf0, 0xc0, 0x00*). In terms of speed, the proposed technique (125 MHz) is fast enough to deliver the typical mobile video sequences (11MHz for CIF/QCIF and 72MHz for HD720 [27, 28]).

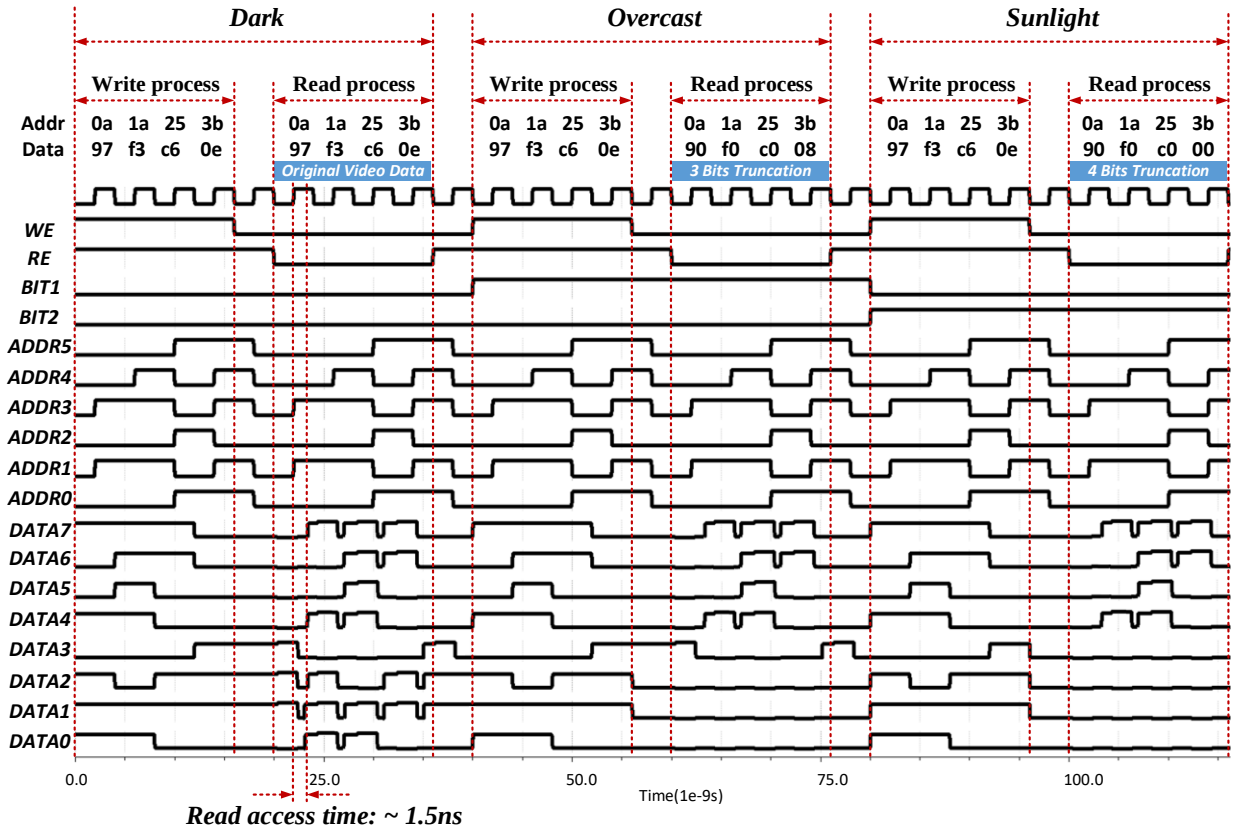


Figure 34. Timing diagram of VCAS in sunlight and in dark.

Multiple videos with varying texture/motion features, recommended by the Joint Collaborative Team on Video Coding (JCT-VC), are used to conduct comprehensive real-world tests. 15 participants were invited to watch videos under three different contexts on an iPhone 6 device. Participants were asked to identify any quality degradation after watching both the original quality and truncated videos under a given luminance context. Approximately 86.7% of the data showed that the participants could not tell the difference between the two videos. For the videos that indicated with a difference, a slight difference in brightness or shading contrast are often to be the cause.

Finally, the power efficiency of VCAS is calculated. Table 9 lists the VCAS power savings. In the overcast and sunlight, VCAS results in 44.9% and 57.2% power savings.

Table 9. Power Savings of VCAS in Different Contexts.

context	***In dark***	***In overcast***	***In sunlight***
Video data	xxxxxxxx	xxxxx000	xxxx0000
Write power	3.50E-07	1.10E-07	6.96E-08
Read power	1.11E-06	6.94E-07	5.55E-07
Power savings	0%	44.9%	57.2%

6.5. Conclusion

Table 10 shows the comparison between the VCAS performance with the state-of-the-art. VCAS presents the negligible implementation cost with an adaptive power-quality tradeoff. Two designs, [27] and [29], give higher power savings than this design. However, these designs are implemented with considerable area overhead (52% and 14.4% respectively). And more notably, this design maximizes the power saving by investigating the relationship between the viewer experience and surroundings.

Table 10. Comparison with Prior Art on Low-Power Mobile Video SRAM.

	TVLSI'08 [14]	***TCASVT'11 [30]***	***TCASII'12 [27]***	***ISVLSI'07 [29]***	**This work**		
					In dark	**In overcast**	**In sunlight**
video specific characteristics	correlation of MSB	contribution of MSB and LSB	different contribution of MSB and LSB	reconstructed image memory	ambient luminance awareness		
dynamic adaption	No	No	No	No	**Yes**		
low-power technique	data flipping	6T+8T bitcells	8T+10T bitcells	10T non-precharge	Bit-truncation for run-time adaption		
bitcell array modification	Yes	Yes	additional word line	No	No		
additional hardware needed	majority logic and data flipping block	single-ended 6T, peripheral circuitries	No	10T SRAM cells	VCAS control circuit		
power penalty for extra bits	Yes	No	No	No	No	No	No
readout power	-14%	-32%	-95%	-74%	**0%**	**-44.9%**	**-57.2%**
video quality[1]	good	acceptable	acceptable	good	**good**	**acceptable**	**acceptable**
area overhead	+14%	+11.64%	+52%	+14.4%	<0.01%		
technology	90nm	90nm	45nm	90nm	45nm		

[1] good: without any quality loss detected; acceptable: without significant quality loss

7. CONTENT-ADAPTIVE MEMORY FOR VIEWER-AWARE SYSTEM [48] [4]

With the success of the paper introduced above, we further looked into the contents of the video to adjust the energy-quality trade-off according to the viewer's experience. In this work, we aim to find a better way to analyze videos in a quantitative way that hardware researchers will also find useful.

7.1. Introduction on Influence of Video Content

Recently developed video macroblock (MB) characterization by analyzing the pixel luminance values' variance [31] is adapted for this work. The analysis of MB variance is typically conducted during the pre-processing stage of video encoding [32, 33]. Plain and textured MBs are defined in Equation (9):

$$V_{MB} = \sum_{i=0}^{15}\sum_{i=0}^{15}(P(i,j) - \rho_{MB})^2 \gg 8$$
$$MB = \begin{cases} Plain & if(V_{MB} \leq Th_{Low}) \\ Textured & Else \end{cases} \quad (9)$$

where ρ_{MB} and V_{MB} are the average luminance and variance of luminance values in a given MB, respectively.

In our analysis, we use their defined calculation to determine whether a given MB is considered either plain or textured, which prevents introducing significant computational overhead. This calculation is based on the variance of pixel luminance values of a certain MB. Figure 35 shows two video samples with similar *PSNR* values, but with varying percentages of

[4] The material in this chapter was co-authored by Yifu Gong and Jonathon Edstrom. Yifu Gong held primary responsibility for SRAM hardware design, verifiecation, and power analysis. The software simulation was investigated by Jonathon Edstrom. Yifu Gong and Jonathon Edstrom designed experiment, collected and analyzed the data.

plain MBs. An important observation is that the banding distortion and plain MBs have a noticeable relationship; videos with larger amounts of plain MBs, especially where plain MBs are compact, tend to decrease the visual experience to the viewers. We, therefore, use this relationship to develop a content-adaptive model to predict the number of LSBs truncated for various videos.

PSNR: 48.1 dB
Min. Plain MB %: 18.37%
Max. Plain MB %: 20.22%
Median Plain MB %: 19.35%
Mean Plain MB %: 19.34%

PSNR: 47.8 dB
Min. Plain MB %: 53.26%
Max. Plain MB %: 54.56%
Median Plain MB %:53.69%
Mean Plain MB %: 53.71%

Figure 35. Plain MBs visualization and video output comparison of two videos with varying plain MB % (with 2 LSBs truncated). White: plain MBs.

7.2. Methodology

In order to determine the acceptable number of LSBs for different videos, subjective video testing is carried out and two models are developed using decision tree and logistic regression methods based on the data collected. The decision tree model is shown in Figure 36. By going through from the root node to the leaves, we can have the number of truncated LSBs. Unlike the decision tree model, the logistic regression model only has one threshold value, which is 28.504%. Accordingly, if the MB percentage is greater than 28.504%, 1 LSB is truncated; otherwise, 2 LSBs would be truncated.

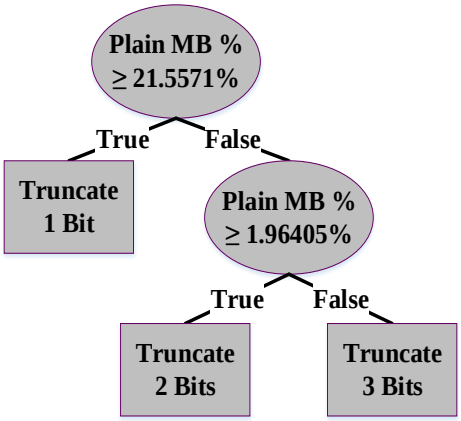


Figure 36. Developed decision tree model for bit-truncation.

A novel truncating methodology is applied for this work. For the bit-truncation technique, a crucial question is what value should be set for those truncated bits. In the previous researches, these truncated LSBs are set to 0s, however, the PSNR equation gives a different answer. Based on Equation 6, we know that to the smaller MSE, the higher PSNR. Assume that the true value of each bit in a pixel is evenly distributed, so Equation 10 can be derived from Equation 7:

$$MSE \propto (Org - Deg)^2 = \frac{1}{2^n}\sum_{x=0}^{2^n-1}(x - x_0)^2 \tag{10}$$

where n represents the number of the truncated bits. x is the true decimal value for the truncated LSBs, and x_0 is the given value for the truncated LSBs. By applying integration, we have:

$$\begin{aligned}\frac{1}{2^n}\sum_{x=0}^{2^n-1}(x - x_0)^2 &= \frac{1}{2^n}\int_0^{2^n-1}(x - x_0)^2\,dx \\ &= \frac{3(2^n-1)}{2^n}[3x_0^2 - 3x_0(2^n-1) + (2^n-1)^2]\end{aligned} \tag{11}$$

Now we assume Equation 11 is a function of x_0. Since the coefficient of ${x_0}^2$ is greater than 0, the smallest MSE can be obtained at where the derivative of Equation 11 is 0, expressed as:

$$f'(x_0) = \frac{3(2^n-1)}{2^n}[6x_0 - 3(2^n-1)] \tag{12}$$

When $x_0 = (2^n - 1)/2$ the derivative of $f(x_0)$ equals 0. However, x_0 is the number of truncated LSBs, which has to be an integer. Thus, $x_0 = 2^{n-1} - 1$ or $x_0 = 2^{n-1}$ are the best solutions to increase PSNR.

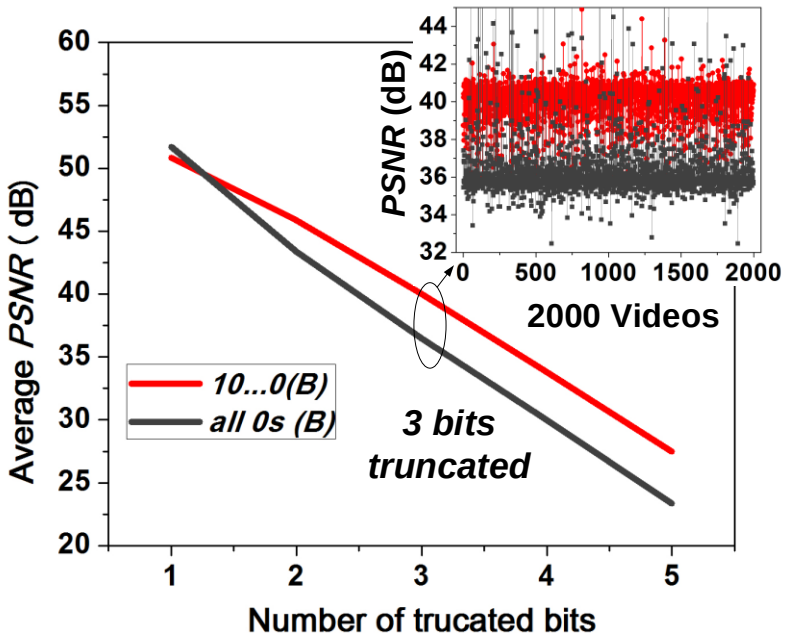


Figure 37. Average PSNR values of 2,000 YouTube-8M videos using two different truncation techniques.

To prove this proposition is true not only in mathematics justification but also in real video examples, 2,000 random videos are selected from YouTube-8M for the experiment. As illustrated in Figure 37, by setting the truncated bits to be 2^{n-1} ($10\cdots0$ with $n-1$ zeros), the PSNR values have significant increases, therefore providing a better viewing experience for the same videos.

7.3. Hardware Design

The proposed memory is implemented using a 45 nm CMOS technology [34]. The architecture of the proposed viewer-aware dynamic bit-truncation memory is shown in Figure 38, which contains 4 blocks of 256×32 6T SRAM bit-cells.

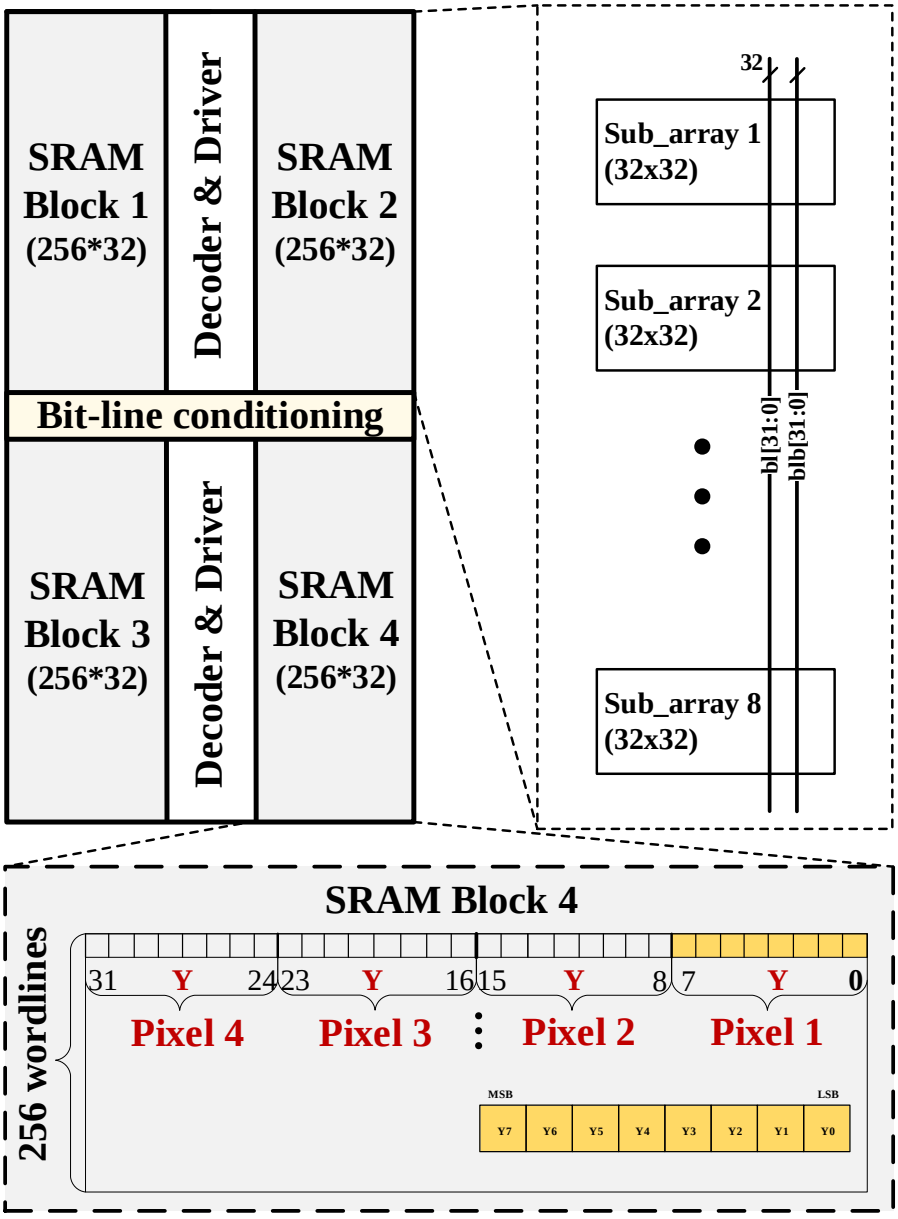


Figure 38. Content-adaptive video memory structure.

Figure 39 shows the truncation controller. Since the truncating signal has two bits (B<0>, B<1>), we are able to truncate 4 LSBs without additional area overhead. Even though 4 LSB truncation did not appear in the test bench, it may happen to some extrema cases such as white noise screen. Then the models will be updated, and our designed memory can easily adapt to it. Two different bit-line conditioning circuits are applied to the memory to enable viewer-aware bit-truncation for LSBs. A pre-charge unit, write driver, and sense amplifier composed the

normal bit-line conditioning circuits, and 4 most significant bits (MSBs) in a byte are connected to it; extra components are added to the remaining bit-lines conditioning circuits to enable bit-truncation, and they are applied to the 4 LSBs in a byte as shown in Figure 39.

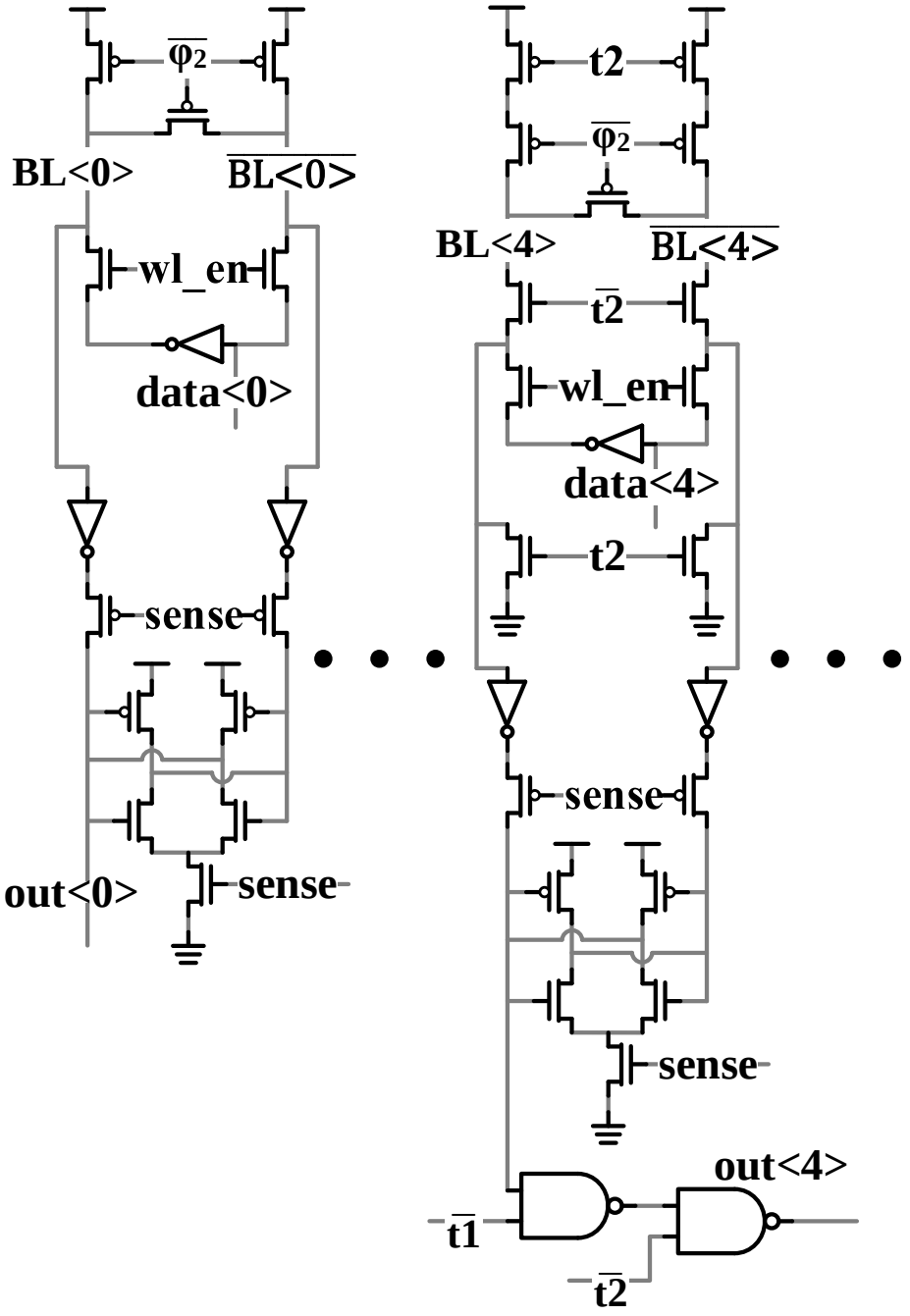


Figure 39. Content-adaptive video memory bit-line conditioning circuits.

φ1 and *φ2* are clock-based signals generated from peripheral circuits, which can be seen in Figure 40. *φ1* enables reading and writing operations; *φ2* controls the pre-charging circuit of the memory. At the end of the reading operation, the *sense* signal turns on for a very short time to reduce the power consumption during the reading operation. The timing diagram in Figure 40 gives an idea of the relationship between these peripheral signals.

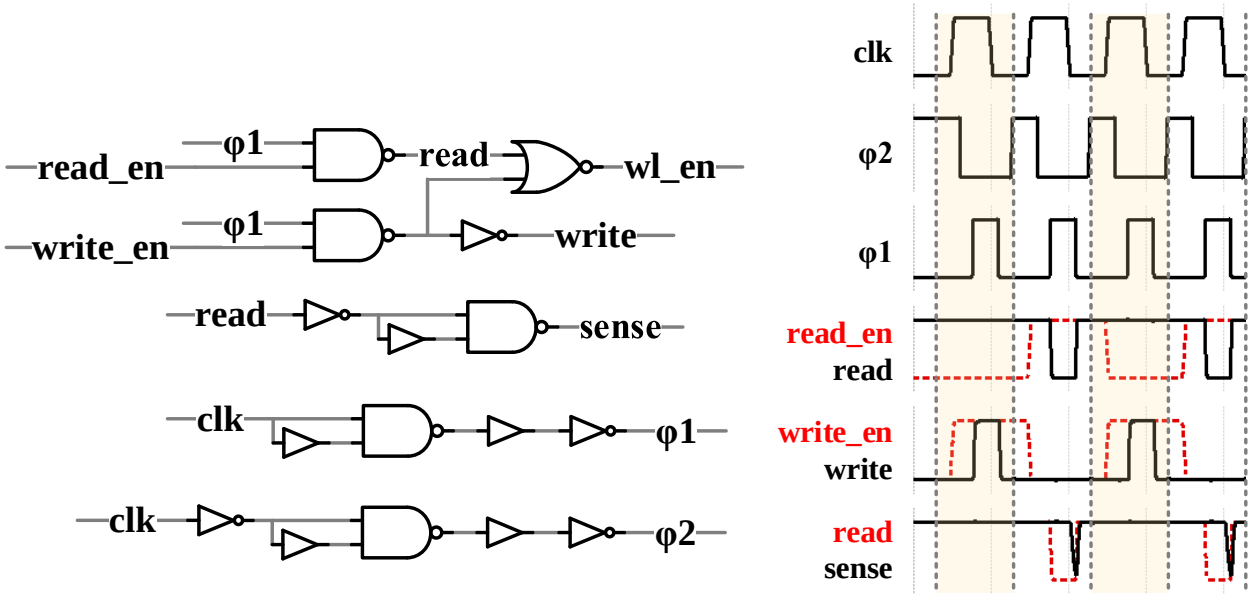


Figure 40. Peripheral circuits with timing diagram.

As shown in Figure 41, three external signals control the truncation process. *trunc_en* controls whether the truncation function is on, and the number of bits to truncate is determined by the other two signals, *B<0>* and *B<1>*. t_1 and t_2 are generated from *B<0>* and *B<1>* signals through two different decoders. A normal 2-to-4 decoder is applied for enabling t_1. The decoder for generating t_2 is a special 2-to-4 truncation control decoder. The truth tables for the decoders are shown in Table 11. Whereas t_1 and t_2 are both **0s**, normal operations are applied; when t_1 is **1**, the pre-charging, write, and reading operations are suspended; on top of t_1 being **1**, if t_2 is **1**, then the output is **0**, otherwise, the output is **1**; the data pattern **01** for t_1 and t_2 never appears.

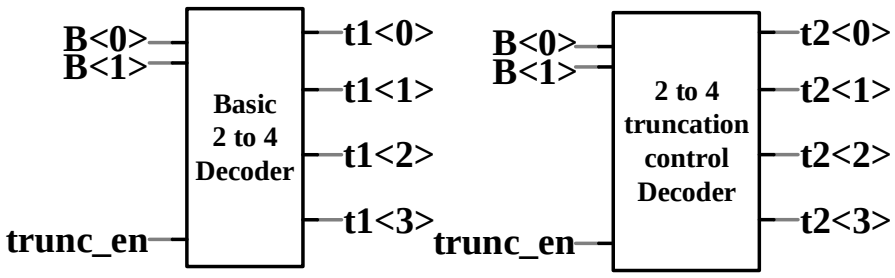


Figure 41. 2 to 4 decoder and truncation control decoder.

Table 11. Truncation Control Decoder Truth Table

Inputs			Outputs			
trunc_en	**B<0>**	**B<1>**	**t2<0>**	**t2<1>**	**t2<2>**	**t2<3>**
0	x	x	0	0	0	0
1	0	0	1	0	0	0
1	0	1	1	1	0	0
1	1	0	1	1	1	0
1	1	1	1	1	1	1

Figure 42 shows the layout design for 512 words × 64 bits viewer-aware bit-truncation SRAM. As designed in the schematic, few gates are added to the bit-line conditioning circuit to enable the truncation function. After careful design, the truncation control decoders can also fit into the free space in the original layout without additional overhead. Compared to traditional SRAM, which is negligible, the proposed memory only consumes 0.32% more silicon area.

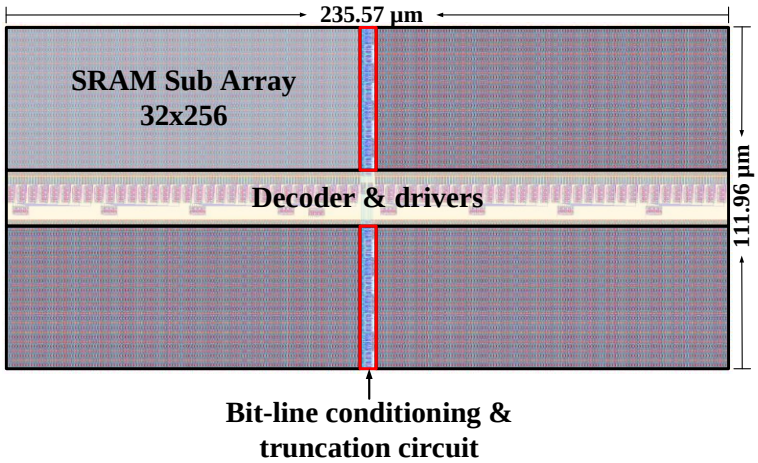


Figure 42. Physical layout design.

7.4. Simulation Results

Figure 43 shows the timing diagram for the proposed memory. To test the functionality of the memory, the data: *0xe9*, *0xce*, *0x62*, and *0x71*, are written to the addresses: *0x55*, *0xb9*, *0xce*, and *0x15*, respectively, and then read out from the same addresses. For example, during a 3 bit-truncation operation, the values read out are *0xec*, *0xcc*, *0x64*, and *0x74*, where the last 3 LSBs for these values are $\mathbf{100}(B)$. The access delay of the reading operation is about 0.5 ns, which is fast enough to deliver the typical mobile video sequences (11MHz for CIF/QCIF and 72MHz for HD720 [29]).

Memory input patterns that cover all possibilities for data switching were tested. These input patterns are used to simulate normal operation and 1 to 4 LSB truncations, and Figure 44 shows the power consumption for each scenario. Respectively, the average power consumption saving for 1 to 4 LSB truncations are 13.54%, 20.10%, 26.83%, and 33.31%, compared to normal operation.

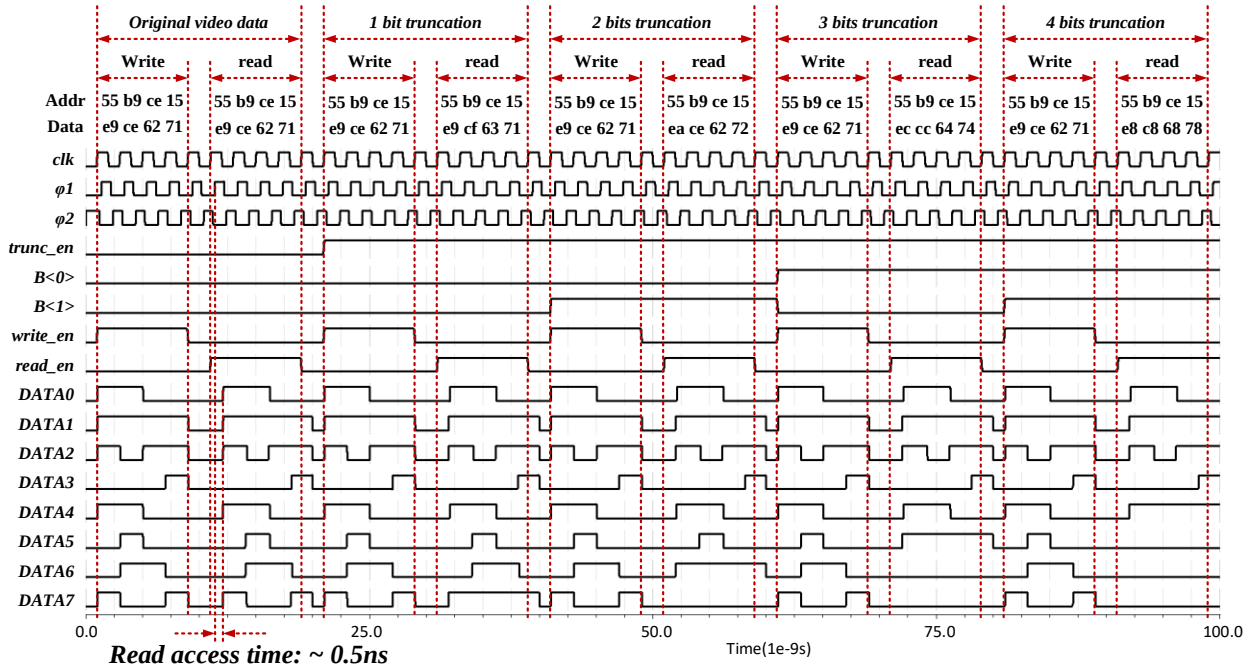


Figure 43. Timing diagram. DATA7: MSB; DATA0: LSB.

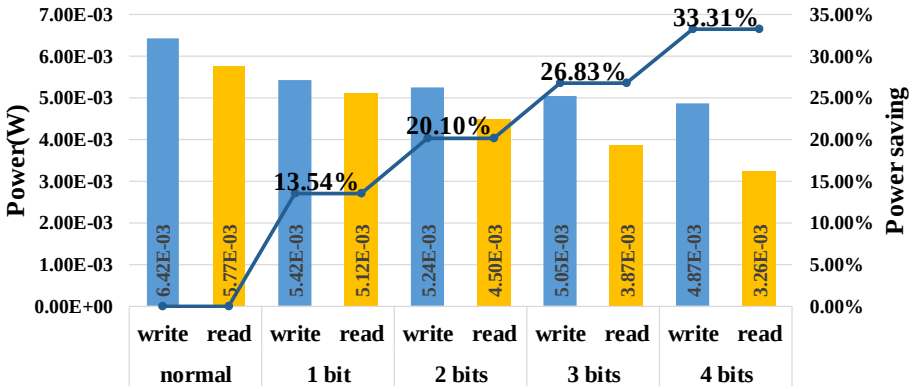


Figure 44. Power savings.

Finally, we conduct psychological experiments at the North Dakota State University Center for Visual and Cognitive Neuroscience to verify the effectiveness of our technique on the viewer's experience. The result of psychological experiments can be seen in Figure 45.

For almost all videos, the developed decision tree model works well. There was only one video out of 20 videos, with tag *wF6lvdXXwc4*, which the vast majority of participants considered unacceptable. A frame from the mentioned video is shown in Figure 46 (top). Severe banding distortion, caused by bit-truncation, appears on the reporter’s face, which can draw attention to the fact that the video quality is changed. That leads viewers to a negative decision on the video. We further process the videos using the ordinal logistic regression model, and it turns out that the ordinal logistic regression model is more conservative that can avoid the worst case of video quality degradation, but some videos may lose opportunities for energy optimization. The same video frame with the number of bits being truncated decided by the ordinal logistic regression model is shown in Figure 46 (bottom).

Figure 45. Video quality testing results using the decision tree model.

Figure 46. Output quality of the video (tag wF6lvdXXwc4): (top) with 3 LSBs truncated using decision tree model and (bottom) with 2 LSBs truncated using the developed ordinal logistic regression model.

7.5. Conclusion

In this paper, we have presented an energy-quality tradeoff of video context-aware memory technique with viewer perspectives. We develop two models to allow hardware adjustment based on the influence of video content to improve the viewer's experience. A novel viewer-aware bit-truncation technique is also implemented to perform energy-quality adaption to the video storage. The designed SRAM can enable up to 33.31% of power saving while providing quality output.

8. MTJ BASED NON-VOLATILE SRAM [50]

8.1. Introduction

Power consumption and performance are the two main concerns for battery-powered portable devices. There is leakage power even the SRAM is not executing any operation. Intuitively, powering off the device enables the most power-saving, but SRAM is a volatile device which means it loses data after powering down. As a result, accessing the external off-chip memory is necessary when powering down the system. Because SRAM read/writing operations require a long store/restore time, up to 50% of total energy is consumed for accessing the external off-chip memory [13]. And reloading data from off-chip memory will produce unavoidable delays and degrade the performance of portable devices. Therefore, a Non-Volatile SRAM is designed to combine high performance and low-power consumption.

There are many researchers focusing on Non-Volatile SRAM design, such as CMOS Technology Compatible Non-Volatile SRAM, Phase Change Memory (PCM) Based Non-Volatile SRAM, and Domain Wall Memory [13] [42] [43]. And one of the most popular ones is Spin Torque Transfer (STT) - RAM with Magnetic Tunnel Junction (MTJ). Each MTJ is made up of 3 layers, two are the ferromagnetic layers (pined-layer and free layer) separated by a tunneling oxide layer. The magnetization of the pined layers is fixed, the other one can be changed by the direction of current flows. When the current flows from the pined-layer to the free layer, the magnetizations of two ferromagnetic layers are parallel, and the MTJ has low resistance; otherwise, magnetizations of two ferromagnetic layers are antiparallel, and that makes the MTJ has high resistance. This feature of MTJ enables the store/restore operations.

This section presents an MTJ based non-volatile SRAM utilizing spin torque transfer magnetization switching in 45 nm technology. The designed non-volatile SRAM circuit is

programed by the peripheral signals to avoid data loss at power down. Data and peripheral signals are store and restore operations are applied before and after power down.

8.2. Methodology

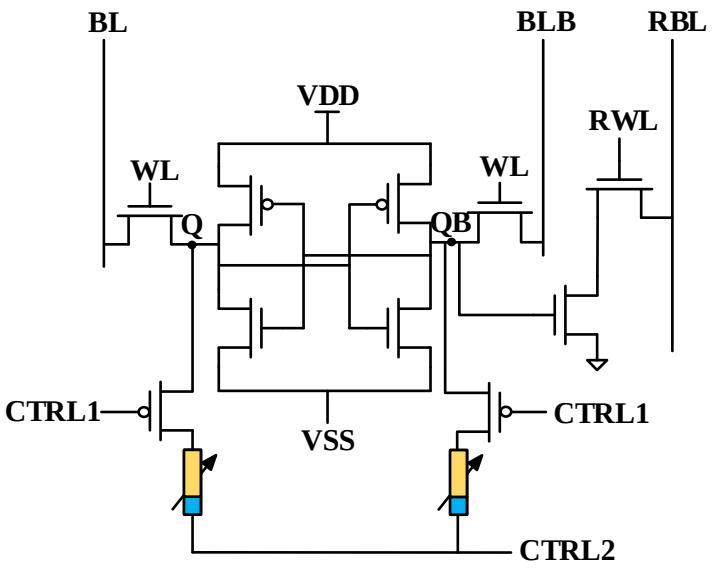


Figure 47. 6T SRAM circuit schematic.

To store data from the SRAM cell (*Q*/*QB*), we need to find a way to transfer the voltage levels to MTJ resistance. Two MTJs with two control PMOS transistors are connected to *Q* and *QB* as shown in Figure 47. The signal values for corresponding operations are listed in Table 12.

Table 12. Control Signals in Different STT-RAM Operation Phases

	VDD	VSS	CTRL1	CTRL2
Normal Operation (NO)	1	0	1	0
Reset (RES)	1	0	0	1.5
Store (STO)	1	0	0	0
Power Down (PD)	1	1	1	1
Restore (RESTO)	1	0	0	1

8.2.1. Normal Operation

During normal operation, the CTRL1 is ‘1’, as a result, MTJs are isolated from SRAM. Since MTJs have no effects on normal SRAM operation, the normal operation is the same as 8T SRAM.

8.2.2. Reset

MTJs must be reset to high resistance before storing data. The most effective way of doing that is to apply a control voltage higher than VDD to CTRL2, so no matter what the values in SRAM cells are, the current will flow from the free layer to the pined-layer through MTJ. In this work, a 1.5×VDD is applied to CTRL2 during reset operation.

8.2.3. Store

A pair of MTJs are used in an STT-RAM cell to store *Q* and *QB*. At this point, both CTRL1 and CTRL2 are ‘0’s. Assume *Q* stores the value ‘1’, which is higher than CTRL2, the current flows from the pined-layer to the free layer through the connected MTJ. And the value stored in *QB* is ‘0’, the same as CTRL2, so there is no current flow. After store operation, one MTJ is at high resistance while the other one stays at low resistance.

8.2.4. Power Down

Instead of putting all the input signals to ‘0’, the power down operation in this design is rising the VSS to ‘1’ to minimize the leakage power through transistors. That is because, during the restore operation, there is a possibility the MTJs get reset and cause an incorrect result.

8.2.5. Restore

The state of SRAM cell is restored during power on. The control signals CTRL1 and CTRL2 are ‘0’ and ‘1’ respectively. High voltage passes through the MTJ with a low resistance to either *Q* or *QB* and forces the other one to be ‘0’.

8.3. Implementation

A 64×64 bits STT-RAM with a peripheral circuit based on 45nm technology is implemented to verify the effectiveness of the proposed design. The circuit schematic can be seen in Figure 48. All the components are Non-Volatile, which means that input and output signals can be restored. STT-RAM array is divided into 8 subarrays in order to store 8 different bits. Storage components are asserted between signals and circuits.

clk signal is for pre-charging the bit-line; *cwddr* and *crddr* signals are the selecting signals for the writing column address and the reading column address respectively; *waddr* and *raddr* signals are the selecting signals for the writing row address and the reading row address respectively; *din* is the data inputs, and *gbl* is data outputs. Note that CTRL1, CTRL2, and STTEN are not listed here because those are global control signals.

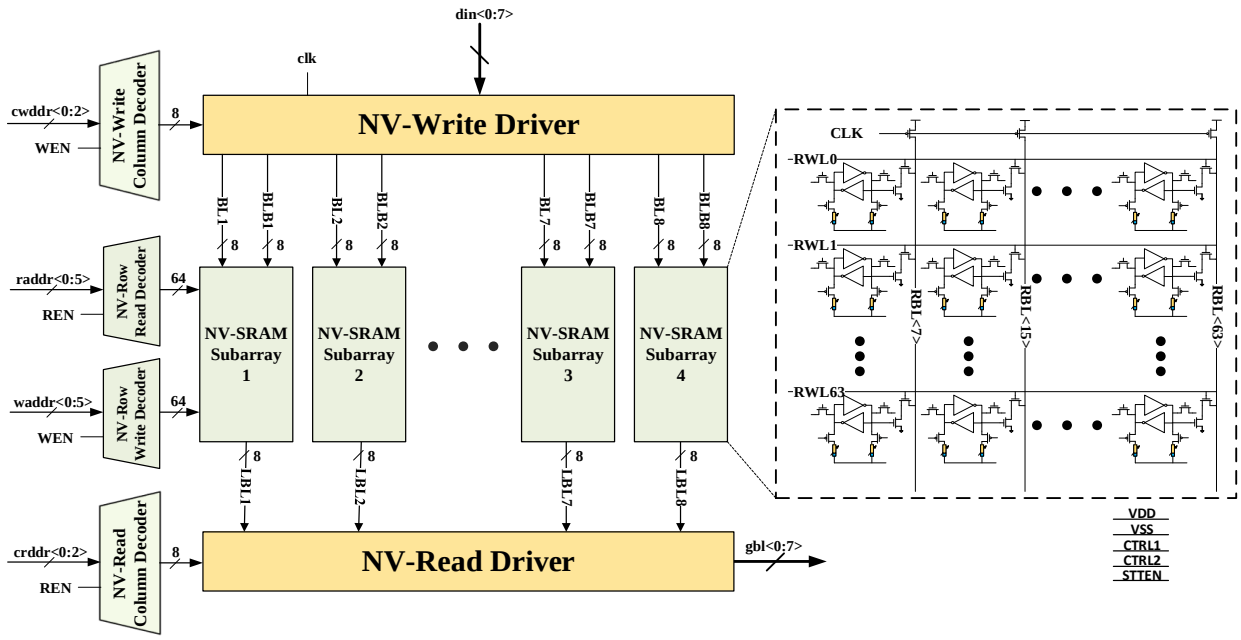


Figure 48. STT-RAM structure with the peripheral circuit.

8.4. Result

In this section, the simulation results are shown including circuit functions, performance, and power consumption. Figure 49 shows the simulation result for a cycle of single STT-RAM cell operation. The values of *q* and *qb* are recovered to their previous values after the restore operation.

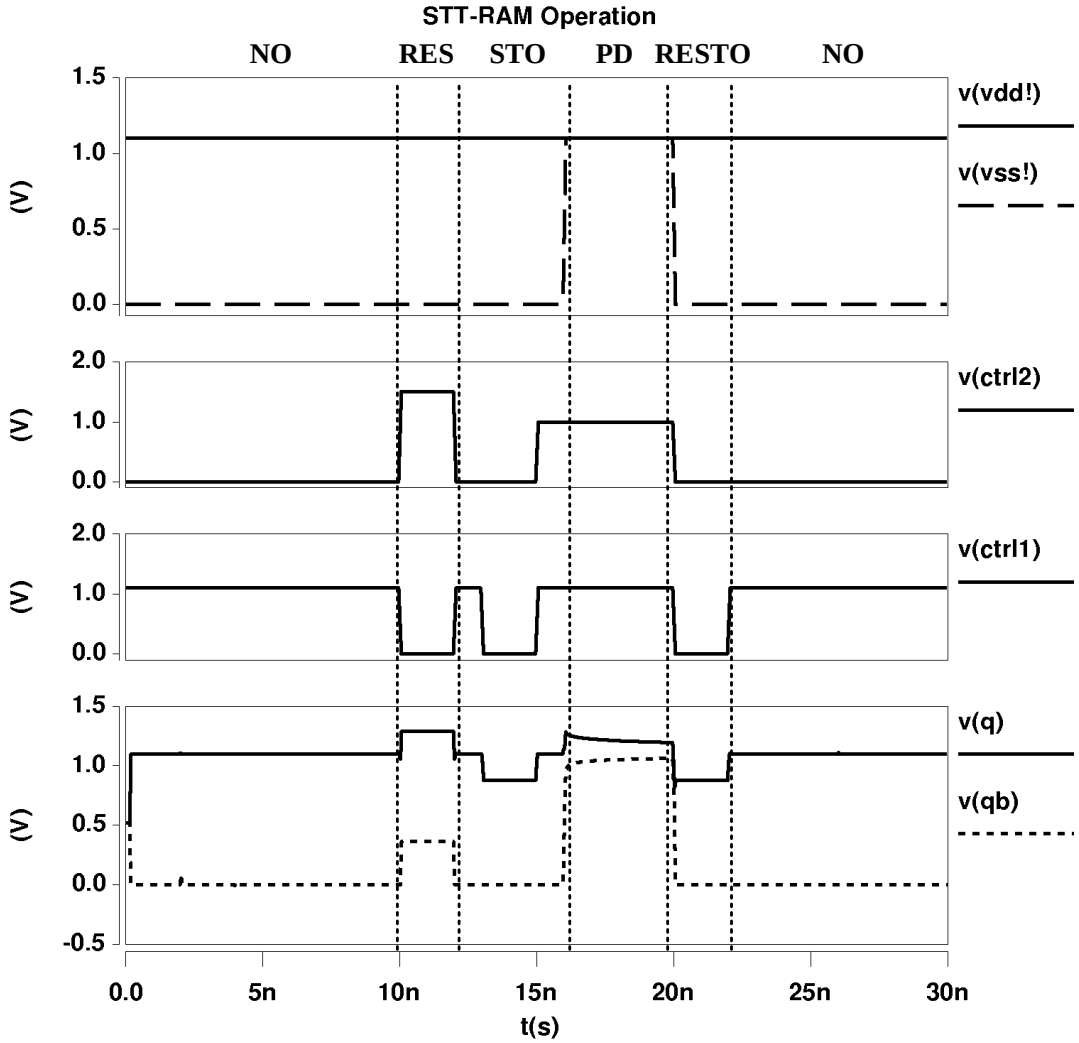


Figure 49. Simulation of a store/restore cycle.

Since performing store/restore operations dissipates a certain amount of energy, we compared it with the SRAM holding (no operation executed, just hold the stored value) energy to obtain the minimum time required for power-down operation to enable energy saving. Energy

dissipations of STT-RAM and CMOS based SRAM during holding operation can be seen in Figure 50. The result shows that after 108 seconds of power down, the STT-RAM achieves energy saving.

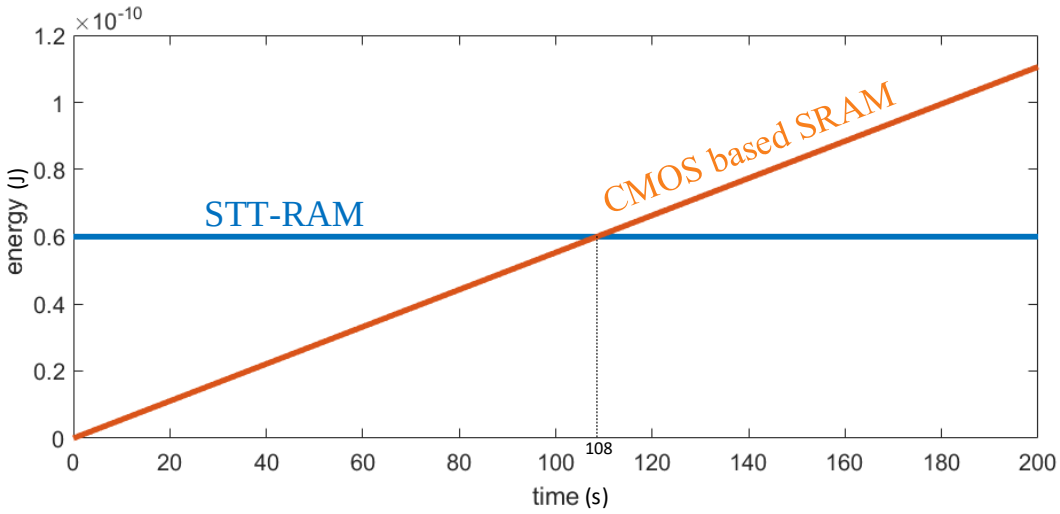


Figure 50. Energy dissipations of STT-RAM during store/restore process and CMOS based SRAM during normal holding operation

9. MACHINE INTELLIGENCE EMBEDDED DEVICE FOR WELDING QUALITY CONTROL [51][5]

9.1. Introduction

Welding is commonly used for connecting metal components in these critical metallic infrastructures, such as agricultural facilities, wind turbines, railways, bridges, and pipelines. However, welding processes vulnerably lead to forming cracks, pores, and other defects on the surface. These defects not only could result in severer cracks and corrosion, but also may ultimately lead to malfunction and failure of metal components. Inspection of welds is thus critical to ensure the welding quality during fabrication, construction process, and later in-service stage. The visual inspection is the crucial and most cost-effective step to determine if the welding quality is passed or rejected. However, fast and accurately determining welding quality is a challenging task in the conventional visual inspection process, which is highly dependent on the experience and expertise of inspectors, and it is fairly subjective and sometimes even misleading. To meet the gap, we bring machine intelligence to welding visual inspection. Specifically, we developed a low-cost portable embedded device to support advanced machine learning algorithms for real-time welding image processing.

Figure 51 shows the typical welding quality control, includes two steps: i) *visual inspection* and ii) *detail inspection if required*. Visual inspection is the first step and it is also the critical and most cost-effective method for welding quality control. It is usually performed by certified welding inspectors who have certified training in welding quality control and defect assessment. During the visual inspection process, over twenty different categories of welding

[5] This work was supported by North Dakota Department of Commerce.

imperfections on the suffice, such as cracks, porosity, inclusions, lack of penetration, lack of fusion, undercut, insufficient weld throat, and misalignment, will be considered to determine if a weldment is passed or rejected. If rejected, the defect information is provided for a redo. If passed by visual inspection, a detail inspection is needed for hidden defect study. In the detail inspection process, nondestructive examination/testing (NDE/NDT) methods such as ultrasonic testing (UT), radiographic testing (RT) and magnetic particle inspection (MT), and phased array ultrasonic technology (PA-UT), are applied to extract hidden defect information. However, this process requires extensive skills in operating expensive equipment and complicated data interpretation, thereby resulting in much higher cost and longer inspection time. Accordingly, an effective first-step visual inspection process will significantly enhance the cost-effectiveness and enable rapid decision making for welding quality control.

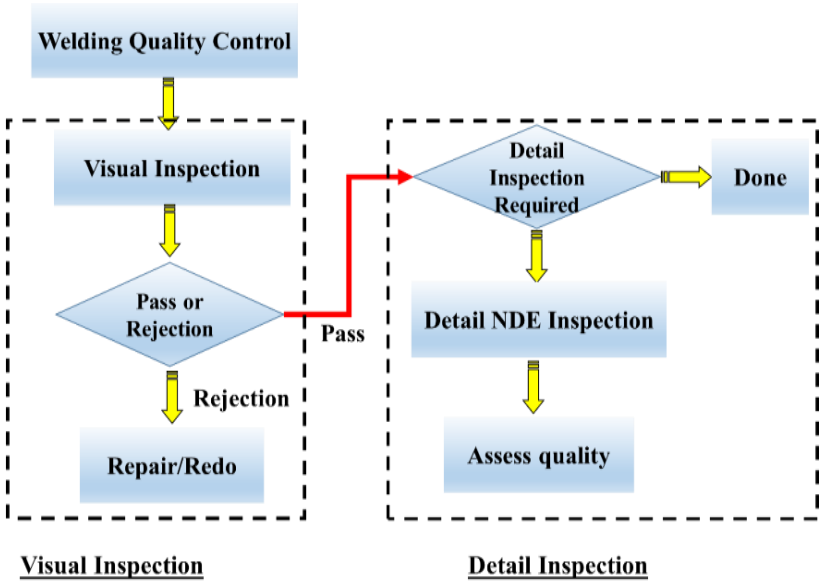


Figure 51. Flowchart of welding quality control (visual inspection and detail inspection) and application of the proposed device to visual inspection.

However, fast and accurately determining welding quality is a very challenging task in today's visual inspection process. First, the threshold acceptance criteria are complex, which is

dependent on precise defect information (e.g. shape, size, and location) collection. For over twenty different weld defects, they have their own characterization from appearance to texture and therefore each category of defects has specific threshold acceptance criteria. Second, the current relied hand tools for visual inspectors are unable to provide an accurate and rapid measurement. Different from detail inspection with advanced equipment (e.g. UT, RT, PUT) available, visual inspectors still rely on hand tools for defect measurement, which is difficult to collect accurate information fast, sometimes even fairly subjective and misleading. Third, with the shortage of welding visual inspectors, today's workforce is aging. Last, but not least, for many welding locations such as tall buildings or bridges, it is not easily accessible for visual inspectors.

In this section, we bring machine intelligence to enhance visual inspection in welding quality control by developing a low-cost and reliable portable embedded device with advanced machine learning techniques. Our developed device significantly enhances the effectiveness of the visual inspection, which will further enable rapid and cost-effective decision making for welding quality control.

9.2. Proposed Technique

We developed a low-cost portable embedded device for fast image processing with advanced machine learning techniques, thereby providing real-time defect information aiding visual welding inspection.

Figure 52 shows an overview of our technology. It mainly consists of three parts: (i) designing machine learning algorithms for accurate defect information extraction and decision making, (ii) implementing the developed machine learning algorithms on an embedded device,

and (iii) developing a portable device with custom system design and optimization. The implementation details for each part will be provided in the next section.

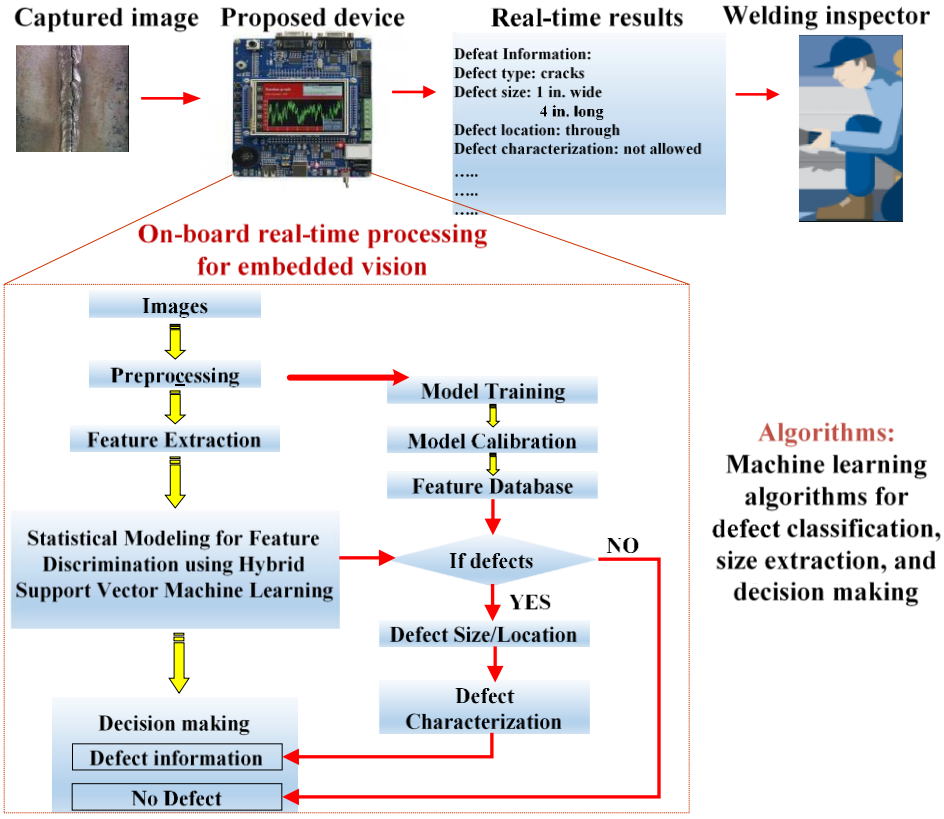


Figure 52. Proposed technique.

9.3. Image Processing and Decision Making

We have developed new image processing and machine learning algorithms for defect information extraction and classification with high reliability. Specifically, instead of using one picture as in our preliminary study, two pictures are captured as input images in our new algorithms. These two pictures have the same camera position but the different angles of light. First, we convert two pictures into grayscale. As shown in Figure 53, the light comes from one side of the weld bead, then the other side.

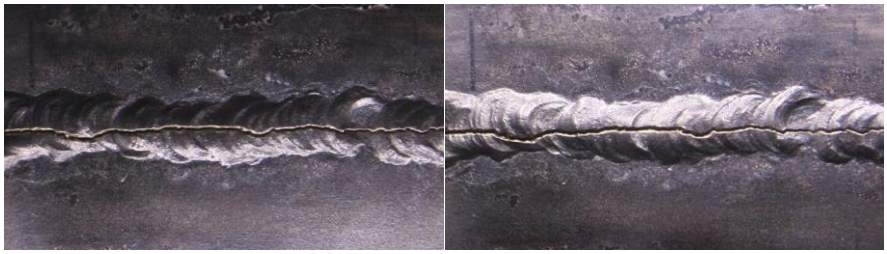


Figure 53. Two input pictures

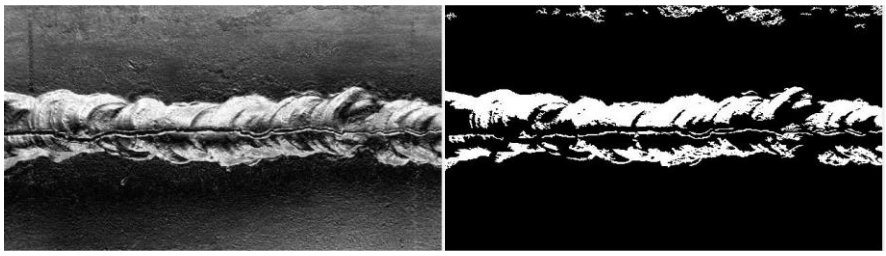


Figure 54. First composited picture in grayscale and binary.

Figure 54 shows the first composited picture in grayscale and binary. The first picture is composited using the absolute values of the subtraction of pixels on the same position from two input pictures, which means the closer the brightness of input pixels is, the darker the composite pixel is. By converting the obtained composite picture to binary (black and white), we can extract a clear boundary for the weld bead. The second picture is composited simply by taking the brighter pixel on the same position from two input pictures. The defects maintain darker color for different angles of lights. Hence, the second composite picture has a clear vision on defects. By using the boundary information extracted from the first binary image, noises located outside of the boundary in the second binary image are eliminated, which significantly enhance the reliability of extracted defect data. Obtained defect information can be seen in Figure 55.

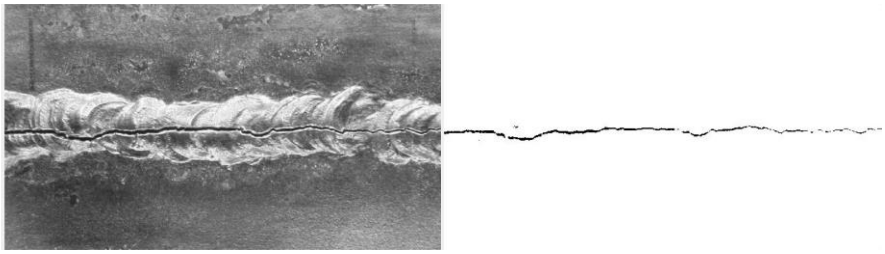


Figure 55. Second composited picture in grayscale and binary.

Based on the extracted defect and welding size information, we have developed a new machine-learning algorithm based on Support Vector Machine (SVM) to classify the types of defects. First, we train an SVM model based on various classified training welding images. During this process, the pictures need to be divided into many cells, and the SVM algorithm is applied to obtain and record features for each cell. Then, we label the features based on the defect type of training image. All the features with labels are stored as a lookup table. The features for input images to be classified need to be compared with the training feature lookup table. The defect type label corresponding to the closest training features in the lookup table decides the defect type in the input image.

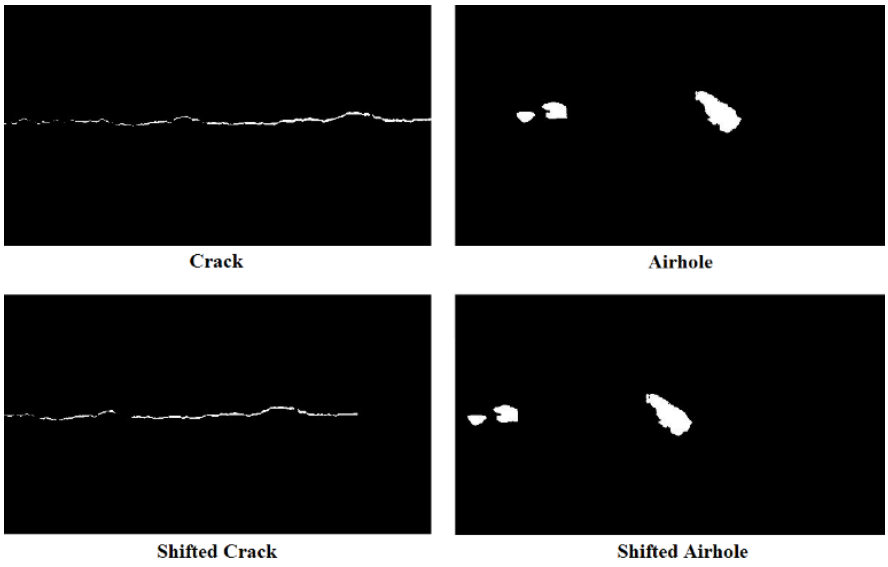


Figure 56. Shifted defect binary images

During this process, a major challenge is that if the defect is not at the center of the image. To solve this, in this project, we develop a displacement algorithm to shift the defect to the center of the image. A Binary Large OBject (BLOB) is a single entity binary data, a chunk of white data in this case. Use the central point of the largest BLOB as the central point of the shifted image (see Figure 56). After shifting, the SVM classifier can classify the defect type correctly. At this point, the algorithm development is finished. Figure 57 shows the complete training and test process.

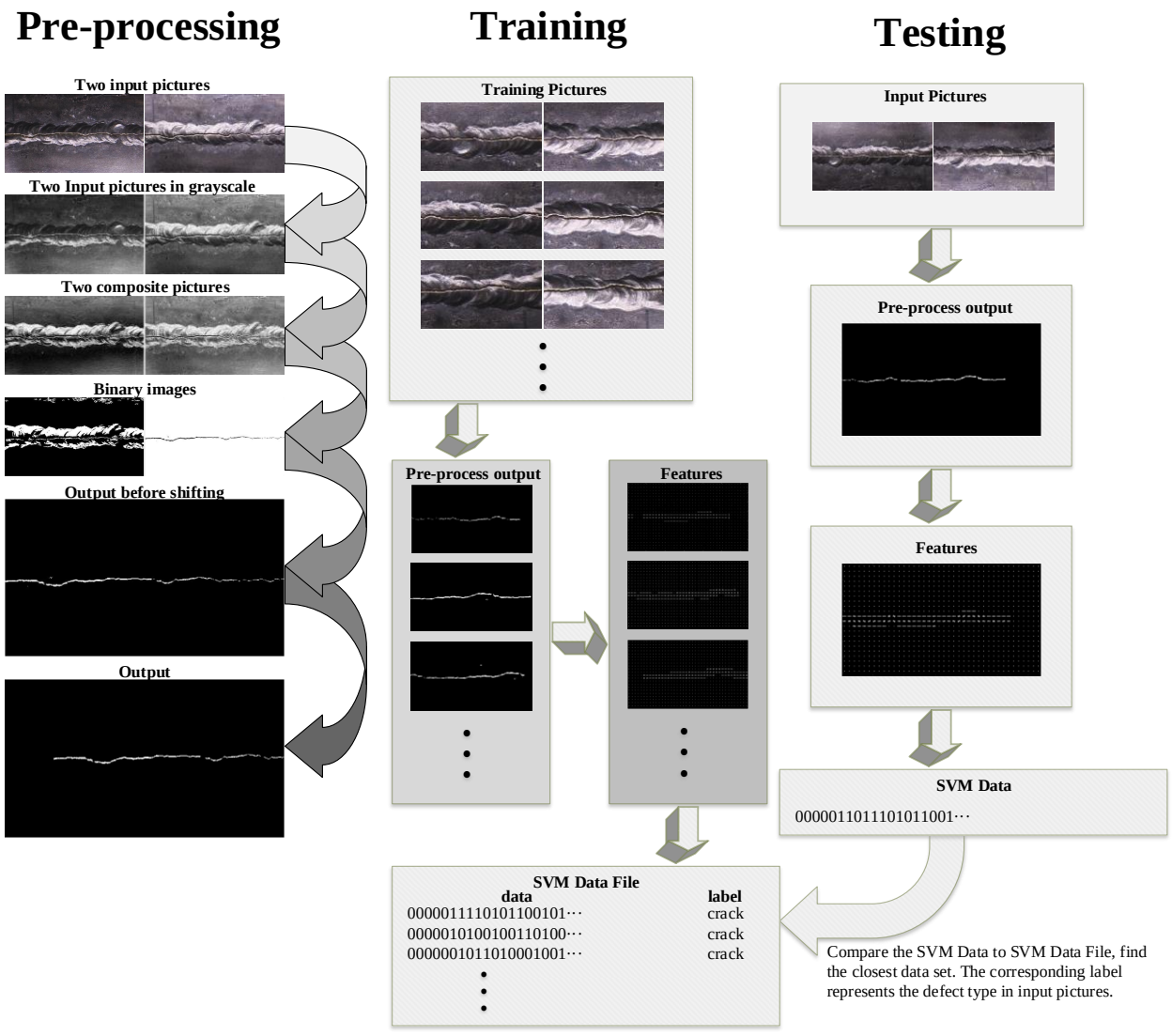


Figure 57. Developed image processing and machine learning algorithms.

9.4. Device Prototype

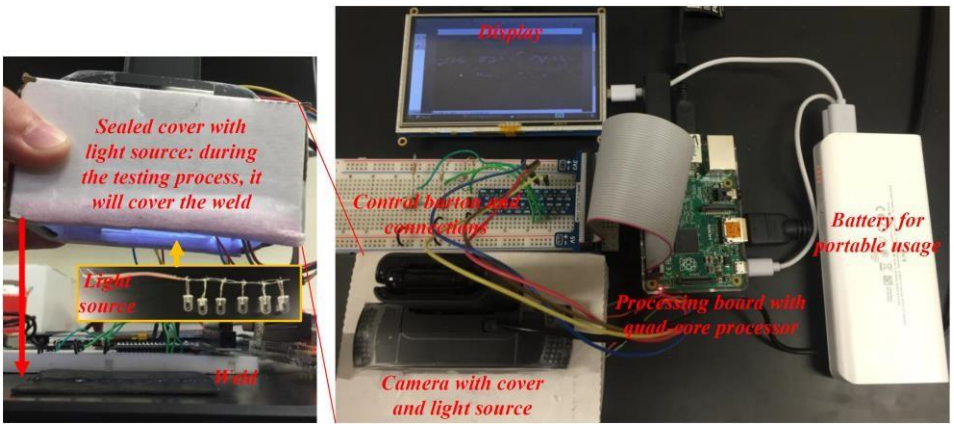
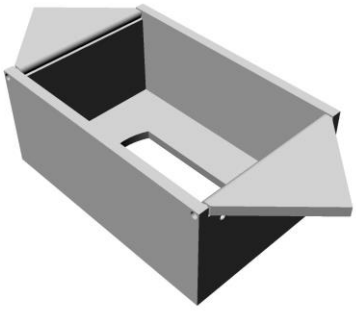


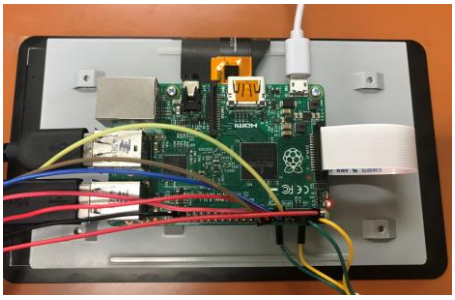
Figure 58. Our developed device prototype.



(a) enclosure 3d model

(b) enclosure

(c) screen and battery



(d) core device and pin connection

Figure 59. Optimized device with an enclosure.

Figure 58 shows the developed device prototype, based on Raspberry PI with a quad-core Cortex A53 processor, which consists of a Raspberry Pi processing board with a quad-core processor, camera, display, and battery. The developed SVM based image processing and

machine learning algorithms can achieve high reliability. The algorithm is implemented in C++ and can be completed within 1 second. In order to enhance the reliability of the developed device, we have optimized the device and developed an enclosure using 3D printing technology to fit different welding shapes with light interference. As shown in Figure 59, the wings on the sides of the box are used for weldment with different shapes.

Different welding samples are tested for device reliability. As shown in Figure 60, the device works well for different defects with high reliability and generates the results within 1 second.

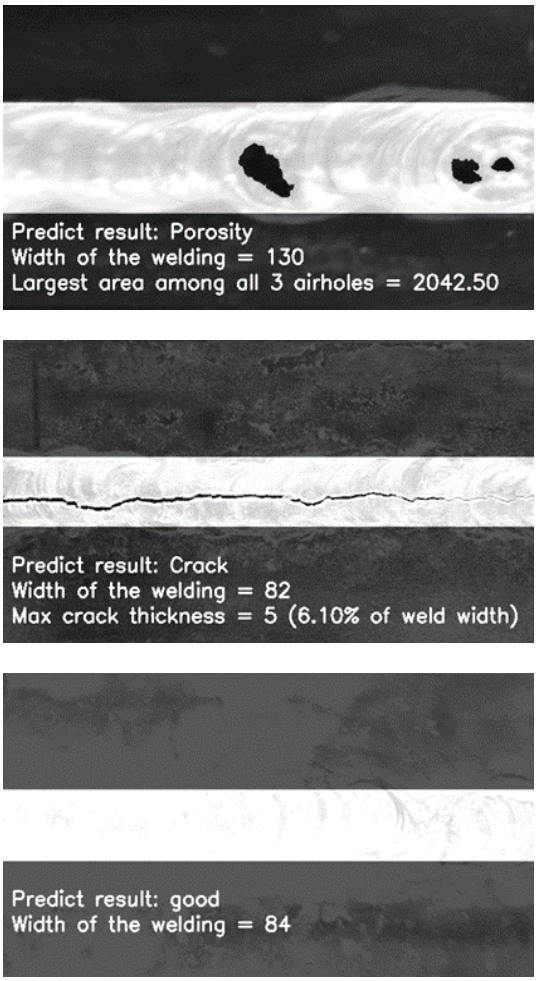


Figure 60. Testing results using different welding samples.

9.5. Conclusion

In this paper, we have developed a low-cost and reliable portable embedded device with advanced machine learning techniques, thereby bringing machine intelligence to enhance visual inspection in welding quality control. This technology has great potential to benefit welding quality control due to the concept of low cost and speed, which will ultimately improve the quality and thus structural safety of civil metallic infrastructure.

10. CONCLUSION AND FUTURE WORK

This chapter summarizes the contributions presented within this dissertation and shows the improvement of the state-of-the-art technologies. A direction for future work will also be introduced.

10.1. Conclusion

In Chapter 3, a method to automatically analyze the SRAM failure rate was described. This study on SRAM stability laid a solid foundation for memory design. In particular, the failure rate associated with SRAM size and type can be plugged into the simulation to determine the parameters of the SRAM cell in different applications.

In Chapter 4, a data-pattern enabled self-recovery SRAM for big video data was presented. An efficient SRAM circuit was designed to enable bit-cell self-recovery at near-threshold voltage by applying the data patterns discovered using the association rule data-mining technique. The proposed design, with a low area overhead of 7.94%, reduces 81.52% of the dynamic power consumption and 82.45% of the leakage power consumption compared with nominal voltage operations. Compared to recent research such as bit cell sizing [15], data-shifting [41], and data-squeezing techniques [39], the designed SRAM provides the best video quality with the least area overhead.

In Chapter 5, a neural network synaptic storage was presented using a Data-driven self-correction technique. Based on the data characteristics obtained using the data-mining technique from Chapter 4, the proposed memory achieves 45.6% and 83.2% in active and leakage power savings, respectively, as compared with the conventional memory design. With the low area cost of 3.17% and less than 1% degradation, the presented memory provides better classification and

less area overhead at similar power efficiency, as compared with recent research on low-power synaptic memory [22],

In Chapters 6 and 7, novel bit-truncation techniques were discussed. These bit-truncation techniques were combined with viewer awareness and a *PSNR* improvement mathematic model. Two low-power video memories were detailed: viewer-aware intelligent video memory and content-adaptive video memory. Viewer-aware bit-truncation techniques, which minimize the negative impact on viewer experience, were also implemented. As compared to our previous efficient video memory designs [28, 29], the new design achieves better video quality with similar power savings.

10.2. Future Work

In the current bit-line conditioning circuits (Figure 39), even though the bit-lines are not pre-charged before reading and writing operations, there is still leakage current while the SRAM cells are accessed. How to improve the truncation bit-line circuit to minimize the leakage power should be included in the future work.

The current study of video memory bit-truncation techniques can only truncate fixed numbers of bits for a video. But the reality is that while playing video, the surrounding environment might not stay the same. Therefore, a feedback mechanism can be very useful. One important observation made during the process of video testing was that if we can detect where the viewer's focus lies in different videos, then we can further improve the quality in these sensitive areas of videos in the future.

As mentioned in [35], macroblocks in the region of interest (ROI), i.e., human faces, can be detected and extracted during the video compression process. If we were able to truncate fewer MSBs (keep more detail) in ROIs, then we can achieve better quality with similar power

savings. In order to do that, the current circuit design needs to be changed, and video tests will be carried out with new processing techniques applied to the videos. An expected output result can be seen in Figure 61. Both images have 4 LSBs truncated, but the second image applies 2 bits truncation for the detected face area. As we can see in the first image, the forehead and cheek area have some obvious quality degradation, which significantly reduces the accept rate during the test. By detecting the human face, we can truncate fewer LSBs (keep more detail) for that area to improve the viewer's experience.

Figure 61. Top: 4 bits truncated; bottom: 4 bits truncated with 2 bits truncated in the detected face area.

REFERENCES

[1] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2017–2022 White Paper. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-738429.html

[2] F. Sampaio, M. Shafique, B. Zatt, S. Bampi, and J. Henkel, "Energy-Efficient Architecture for Advanced Video Memory," in *Proc. 2014 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, Nov. 2014, pp. 132-139.

[3] T. Liu, T. Lin, S. Wang, W. Lee, J. Yang, K. Hou, and C. Lee, "A 125 uW, fully scalable MPEG-2 and H.264/AVC video decoder for mobile applications," *IEEE J. Solid-State Circuits*, vol. 42, no. 1, pp. 161–169, Jan. 2007.

[4] D. Zhou, S. Wang, H. Sun, J. Zhou, J. Zhu, Y. Zhao, J. Zhou, S. Zhang, S. Kimura, T. Yoshimura, S. Goto, "A 4Gpixel/s 8/10b H.265/HEVC Video Decoder Chip for 8K Ultra HD Applications," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC)*, Feb. 2016, pp. 266-267.

[5] O. Hirabayashi *et al*., "A process-variation-tolerant dual-power-supply SRAM with 0.179μm^2 Cell in 40nm CMOS using level-programmable wordline driver," *2009 IEEE International Solid-State Circuits Conference - Digest of Technical Papers*, San Francisco, CA, 2009, pp. 458-459,459a.

[6] K. Nii *et al*., "A 45-nm Bulk CMOS Embedded SRAM With Improved Immunity Against Process and Temperature Variations," in *IEEE Journal of Solid-State Circuits*, vol. 43, no. 1, pp. 180-191, Jan. 2008.

[7] K. Kushida *et al*., "A 0.7 V Single-Supply SRAM With 0.495μm^2 Cell in 65 nm Technology Utilizing Self-Write-Back Sense Amplifier and Cascaded Bit Line Scheme," in *IEEE Journal of Solid-State Circuits*, vol. 44, no. 4, pp. 1192-1198, April 2009.

[8] K. Kushida *et al*., "A 27% active and 85% standby power reduction in dual-power-supply SRAM using BL power calculator and digitally controllable retention circuit," *2013 IEEE Asian Solid-State Circuits Conference (A-SSCC)*, Singapore, 2013, pp. 25-28.

[9] R. V. Joshi, R. Kanj and V. Ramadurai, "A Novel Column-Decoupled 8T Cell for Low-Power Differential and Domino-Based SRAM Design," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 19, no. 5, pp. 869-882, May 2011.

[10] K. Takeda *et al*., "A read-static-noise-margin-free SRAM cell for low-VDD and high-speed applications," in *IEEE Journal of Solid-State Circuits*, vol. 41, no. 1, pp. 113-121, Jan. 2006.

[11] Y. Chiu *et al*., "40 nm Bit-Interleaving 12T Subthreshold SRAM With Data-Aware Write-Assist," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 61, no. 9, pp. 2578-2585, Sept. 2014.

[12] S. A. Verkila, S. K. Bondada and B. S. Amrutur, "A 100MHz to 1GHz, 0.35V to 1.5V Supply 256 x 64 SRAM Block Using Symmetrized 9T SRAM Cell with Controlled Read," *21st International Conference on VLSI Design (VLSID 2008)*, Hyderabad, 2008, pp. 560-565.

[13] M. E. Sinangil and A. P. Chandrakasan, "Application-Specific SRAM Design Using Output Prediction to Reduce Bit-Line Switching Activity and Statistically Gated Sense Amplifiers for Up to 1.9×Lower Energy/Access," in *IEEE Journal of Solid-State Circuits*, vol. 49, no. 1, pp. 107-117, Jan. 2014.

[14] H. Fujiwara *et al*., "A Two-Port SRAM for Real-Time Video Processor Saving 53% of Bitline Power with Majority Logic and Data-Bit Reordering," *ISLPED'06 Proceedings of the 2006 International Symposium on Low Power Electronics and Design*, Tegernsee, 2006, pp. 61-66.

[15] J. Kwon, I. J. Chang, I. Lee, H. Park and J. Park, "Heterogeneous SRAM Cell Sizing for Low-Power H.264 Applications," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 59, no. 10, pp. 2275-2284, Oct. 2012.

[16] Y. Benmoussa, J. Boukhobza, E. Senn and D. Benazzouz, "Energy Consumption Modeling of H.264/AVC Video Decoding for GPP and DSP," *2013 Euromicro Conference on Digital System Design*, Los Alamitos, CA, 2013, pp. 890-897.

[17] J. P. Kulkarni and K. Roy, "Ultralow-Voltage Process-Variation-Tolerant Schmitt-Trigger-Based SRAM Design," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 20, no. 2, pp. 319-332, Feb. 2012.

[18] S. Mukhopadhyay, H. Mahmoodi and K. Roy, "Modeling of failure probability and statistical design of SRAM array for yield enhancement in nanoscaled CMOS," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 12, pp. 1859-1880, Dec. 2005.

[19] J. A. Croon, S. Decoutere, W. Sansen and H. E. Maes, “Physical modeling and prediction of the matching properties of MOSFETs,” in *Proc. 30th ESSCC*, 2004, pp. 193-196.

[20] Y. LeCun, C. Cortes and C. J. Burges, "THE MNIST DATABASE of handwritten digits," 1998. [Online]. Available: http://yann.lecun.com/exdb/mnist/.

[21] N. Gong, J. Edstrom, D. Chen, and J. Wang, “Data-Pattern Enabled Self-Recovery Multimedia Storage System for Near-Threshold Computing,” in *Proc. 34th ICCD*, pp. 492-498, Arizona, Oct. 2016.

[22] G. Srinivasan, P. Wijesinghe, S. S. Sarwar, A. Jaiswal, and K. Roy, “Significance driven hybrid 8T-6T SRAM for energy-efficient synaptic storage in artificial neural networks,” in *Proc. 2016 DATE,* Mar. 2016.

[23] S. Y. Choi,M. R. Luo, andM. R. Pointer, “The influence of the relative luminance of the surround on the perceived quality of an image on a large display,” in *Proc. 15th Color Imaging Conf.,* 2007, pp. 157–162.

[24] Z. Wei and K. N. Ngan, “A temporal just-noticeable *distortion profile for video in DCT domain,”* in *Proc. 15th IEEE International Conference on Image Processing*, pp.1336-1339, 12-15, Oct. 2008

[25] J. Xue and C. W. Chen, “Towards viewing quality optimized video adaptation,” in *2011 IEEE In-ternational Conference on Multimedia and Expo (ICME)*, pp.1-6, July 2011

[26] J. Xue and C. W. Chen, “Mobile JND: Environment adapted perceptual model and mobile video quality enhancement,” in *Proc. 3rd Multimedia Syst. Conf., 2012, ser. MMSys ’12*, pp. 173–183, New York, NY, USA: ACM.

[27] N. Gong, S. Jiang, A. Challapalli, S. Fernandes, and R. Sridhar, “Ultra-Low Voltage Split-Data-Aware Embedded SRAM for Mobile Video Applications,” *IEEE Transactions on Circuits and Systems II*, vol. 59, no. 12, pp. 883-887, Dec. 2012.

[28] J. S. Wang, P. Y. Chang, T. S. Tang, J. W. Chen, and J. I. Guo, “Design of subthreshold SRAMs for energy-efficient quality-scalable video applications,” *IEEE Trans. Emerging Sel. Topics Circuits Syst.*, vol. 1, no. 2, pp. 183-192, Jun. 2011.

[29] H. Noguchi et al., “A 10T non-precharge two-port SRAM for 74% power reduction in video processing,” in *Proc. IEEE Computer Society Annual Symp. VLSI Circuits*, March 2007, pp. 107-112.

[30] I. Chang, D. Mohapatra, and K. Roy, “A priority-based 6T/8T hybrid SRAM architecture for aggressive voltage scaling in video applications,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 2, pp. 101-112, Feb. 2011.

[31] M. Shafique *et al.*, "Application-Guided Power-Efficient Fault Tolerance for H.264 Context Adaptive Variable Length Coding," in *IEEE Transactions on Computers*, vol. 66, no. 4, pp. 560-574, 1 April 2017.

[32] M. Shafique *et al.*, "Application-Guided Power-Efficient Fault Tolerance for H.264 Context Adaptive Variable Length Coding," in *IEEE Transactions on Computers*, vol. 66, no. 4, pp. 560-574, 1 April 2017.

[33] M. Shafique, B. Molkenthin and J. Henkel, "An HVS-based Adaptive Computational Complexity Reduction Scheme for H.264/AVC video encoder using Prognostic Early

Mode Exclusion," *2010 Design, Automation & Test in Europe Conference & Exhibition (DATE 2010)*, Dresden, 2010, pp. 1713-1718.

[34] FreePDK45. [Online]. Available: http://www.eda.ncsu.edu/wiki/FreePDK45:Contents.

[35] F. Peng, X. Zhu and M. Long, "An ROI Privacy Protection Scheme for H.264 Video Based on FMO and Chaos," in *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 10, pp. 1688-1699, Oct. 2013.

[36] E. Seevinck, F. J. List and J. Lohstroh, "Static-noise margin analysis of MOS SRAM cells," in *IEEE Journal of Solid-State Circuits*, vol. 22, no. 5, pp. 748-754, Oct. 1987.

[37] J. Xue and C.-W. Chen, "A Study on Perception of Mobile Video with Surrounding Contextual Influences," in Proceedings of the 4th International Workshop on Quality of Multimedia Experience (QoMEX), Melbourne, Austrailia, Jul. 2012.

[38] F. Frustaci, M. Khayatzadeh, D. Blaauw, D. Sylvester and M. Alioto, "SRAM for ErrorTolerant Applications with Dynamic Energy-Quality Management in 28 nm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 50, no. 5, pp. 1310-1323, May 2015.

[39] A. Ferrerón, D. Suárez-Gracia, J. Alastruey-Benedé, T. Monreal-Arnal and P. Ibáñez, "Concertina: Squeezing in Cache Content to Operate at Near-Threshold Voltage," in *IEEE Transactions on Computers*, vol. 65, no. 3, pp. 755-769, 1 March 2016.

[40] S. Zhou, S. Katariya, H. Ghasemi, S. Draper and N. S. Kim, "Minimizing total area of low-voltage SRAM arrays through joint optimization of cell size, redundancy, and ECC," *2010 IEEE International Conference on Computer Design*, Amsterdam, 2010, pp. 112-117.

[41] S. Ganapathy, G. Karakonstantis, A. Teman, and A. Burg, “Mitigating the Impact of Faults in Unreliable Memories for Error-Resilient Applications,” in *Proc. Design Automation Conf. (DAC)*, 2015, pp. 1-6.

[42] K. Huang, Y. Ha, R. Zhao, A. Kumar and Y. Lian, "A Low Active Leakage and High Reliability Phase Change Memory (PCM) Based Non-Volatile FPGA Storage Element," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 61, no. 9, pp. 2605-2613, Sept. 2014.

[43] R. Venkatesan et al., "Cache Design with Domain Wall Memory," in IEEE Transactions on Computers, vol. 65, no. 4, pp. 1010-1024, 1 April 2016.

[44] D. Chen, J. Edstrom, **Y. Gong** et al., "Viewer-Aware Intelligent Efficient Mobile Video Embedded Memory," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 4, pp. 684-696, April 2018.

[45] J. Edstrom, **Y. Gong**, D. Chen, J. Wang and N. Gong, "Data-driven Intelligent Efficient Synaptic Storage for Deep Learning," in *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 64, no. 12, pp. 1412-1416, Dec. 2017.

[46] J. Edstrom, D. Chen, **Y. Gong**, J. Wang and N. Gong, "Data-Pattern Enabled Self-Recovery Low-Power Storage System for Big Video Data," in *IEEE Transactions on Big Data*, vol. 5, no. 1, pp. 95-105, 1 March 2019.

[47] Y. Xu, H. Das, Y. Gong and N. Gong, "On Mathematical Models of Optimal Video Memory Design," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 1, pp. 256-266, Jan. 2020.

[48] J. Edstrom, **Y. Gong** et al., "Content-Adaptive Memory for Viewer-Aware Energy-Quality Scalable Mobile Video Systems," in *IEEE Access*, vol. 7, pp. 47479-47493, 2019. **(Co-first author)**

[49] **Yifu Gong**, Na Gong, Ligang Hou and Jinhui Wang, "Platform design for compatible semi-custom design flow," *2016 13th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT)*, Hangzhou, 2016, pp. 1624-1626.

[50] **Yifu Gong**, Na Gong, Ligang Hou and Jinhui Wang, "MTJ based data restoration in non-volatile SRAM," *2016 13th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT)*, Hangzhou, 2016, pp. 1011-1013.

[51] **Yifu Gong**, Zhibin Lin, Jinhui Wang, Na Gong, "Bringing Machine Intelligence to Welding Visual Inspection: Development of Low-Cost Portable Embedded Device for Welding Quality Control," *Electronic Imaging, Intelligent Robotics and Industrial Applications using Computer Vision 2018*, California, 2018, pp. 279-1-279-4(4).

APPENDIX A. SRAM SNM SIMULATION TOP BATCH

```
::    SRAM SNM SIMULATION
::    YIFU GONG - 2017
::    Top batch to simulate SRAM failure rate
::    Department: NDSU ECE Graduate Research
::    Parameters: Voltage(mV), Iteration, SRAM(6T or 8T)
::                      Process corners(ss or ff)×2, lambda, technology node

@echo off

start /wait readwrite.bat 500 1000000 6T ss ff 4 45
start /wait readwrite.bat 500 1000000 6T ss ss 4 45
start /wait readwrite.bat 500 1000000 6T ff ff 4 45
start /wait readwrite.bat 500 1000000 6T ff ss 4 45

start /wait readwrite.bat 500 1000000 8T ss ff 4 45
start /wait readwrite.bat 500 1000000 8T ss ss 4 45
start /wait readwrite.bat 500 1000000 8T ff ff 4 45
start /wait readwrite.bat 500 1000000 8T ff ss 4 45
```

APPENDIX B. SRAM SNM SIMULATION SECOND BATCH

```
::    SRAM SNM SIMULATION
::    YIFU GONG - 2017
::    Second batch to simulate SRAM failure rate
::    Department: NDSU ECE Graduate Research
::    Perform simulations with given parameters
::    Self-update

@echo off
setlocal enableextensions disabledelayedexpansion

set a=%1
set mont=%2
set t=%3
set Nsf=%4
set Psf=%5
set lambda=%6
set nm=%7

shift
shift
shift
shift
shift
shift
shift

set textFile1=snm_read_%t%.sp
set textFile2=snm_write_%t%.sp
set outpot1=snm_read_%t%.lis
set outpot2=snm_write_%t%.lis
set MatlabFile1=read_data
set MatlabFile2=write_data
set Matlabplot=failureplot

copy NUL %nm%_failure_%t%_%Nsf%_%Psf%_%lambda%lambda.csv
set loopcount=10

:loop
START /b /WAIT python replace.py %1 %textFile1% %a% %mont% %Nsf% %Psf%
%lambda% %nm%
C:\synopsys\Hspice_L-2016.03-1\WIN64\hspice %textFile1% > %outpot1% -mp
START /b /WAIT python convert_file_read.py %2 %outpot1%
START /b /WAIT matlab -nodisplay -nosplash -nodesktop -nojvm -r
v=%a%;mont=%mont%;file='%nm%_failure_%t%_%Nsf%_%Psf%_%lambda%lambda.csv';%Mat
labFile1%

START /b /WAIT python replace.py  %3 %textFile2% %a% %mont% %Nsf% %Psf%
%lambda% %nm%
C:\synopsys\Hspice_L-2016.03-1\WIN64\hspice %textFile2% > %outpot2% -mp
START /b /WAIT python convert_file_write.py %4 %outpot2%
START /b /WAIT matlab -nodisplay -nosplash -nodesktop -nojvm -r
v=%a%;mont=%mont%;file='%nm%_failure_%t%_%Nsf%_%Psf%_%lambda%lambda.csv';%Mat
labFile2%
```

```
set /a a=a+50
set /a loopcount=loopcount-1
if %loopcount%==0 goto exitloop
goto loop
:exitloop

matlab -nodisplay -nosplash -nodesktop -r %Matlabplot%

exit
```

APPENDIX C. SNM_WRITE_6T NETLIST

```
.TEMP 80.0000
.GLOBAL VDD
.GlOBAL GND

.include 45nm_NMOS_tt.pm
.include 45nm_PMOS_tt.pm
.param mont = 50000
+ Vdd = 1.0v
+ GND = 0
.param m = 4
+ L = 45n
+ LP = 'L'
+ LN = 'L'
+ LA = 'L'
+ WP = 90n
+ WN = 90n
+ WA = 90n
+ BITCAP = 1E-12
*.PARAM  dxvth=agauss(0,0.1,1)
.PARAM dxvthn1 = agauss(0,'0.024*sqrt(45*67.5e-18/(WN*LN))',1)
+        dxvthn2 = agauss(0,'0.024*sqrt(45*67.5e-18/(WN*LN))',1)
+        dxvthp1=agauss(0,'-0.0292*sqrt(45*67.5e-18/(WP*LP))',1)
+        dxvthp2=agauss(0,'-0.0292*sqrt(45*67.5e-18/(WP*LP))',1)
+        dxvtha1 = agauss(0,'0.024*sqrt(45*67.5e-18/(WA*LA))',1)
+        dxvtha2 = agauss(0,'0.024*sqrt(45*67.5e-18/(WA*LA))',1)

*sources
**supply
VVDD VDD 0 dc = VDD
Vvss GND 0 dc = 0

**access control
Vwwl WWL 0 dc = VDD
* one inverter
MPL QBD P VDD VDD PMOS W='WP' L='LP' delvto='dxvthp1'
MNL QBD P GND GND NMOS W='WN' L='LN' delvto='dxvthn1'

* one inverter
MPR QD P VDD VDD PMOS W='WP' L='LP' delvto='dxvthp2'
MNR QD P GND GND NMOS W='WN' L='LN' delvto='dxvthn2'

* access transistors
Mr WBLB WWL QBD  VDD NMOS W = 'WA' L='LA' delvto='dxvtha1'
Ml WBL  WWL  QD  GND NMOS W = 'WA' L='LA' delvto='dxvtha2'

VBLB WBLB 0 dc = VDD
VBL  WBL 0 dc = 0

Vin P 0
.DC Vin LIN 30 0v VDD SWEEP MONTE=mont

.PRINT DC V(QD) V(QBD)
.OPTIONS brief=1 NOMOD NOWARN INGOLD=2
.end
```

APPENDIX D. SNM_READ_6T NETLIST

```
.TEMP 80.0000
.GLOBAL VDD
.GlOBAL GND

.include 45nm_NMOS_tt.pm
.include 45nm_PMOS_tt.pm
.param mont = 50000
+ Vdd = 1.0v
+ GND = 0
.param m = 4
+ L = 45n
+ LP = 'L'
+ LN = 'L'
+ LA = 'L'
+ WP = 50n
+ WN = 100n
+ WA = 75n
+ BITCAP = 1E-12
*.PARAM  dxvth=0.1
.PARAM dxvthn1 = agauss(0,'0.024*sqrt(100*67.5e-18/(WN*LN))',1)
+        dxvthn2 = agauss(0,'0.024*sqrt(100*67.5e-18/(WN*LN))',1)
+        dxvthp1=agauss(0,'-0.0292*sqrt(50*67.5e-18/(WP*LP))',1)
+        dxvthp2=agauss(0,'-0.0292*sqrt(50*67.5e-18/(WP*LP))',1)
+        dxvtha1 = agauss(0,'0.024*sqrt(75*67.5e-18/(WA*LA))',1)
+        dxvtha2 = agauss(0,'0.024*sqrt(75*67.5e-18/(WA*LA))',1)

*sources
**supply
VVDD VDD 0 dc = VDD
Vvss GND 0 dc = 0

**access control
Vwwl WWL 0 dc = VDD
* one inverter
MPL QBD P VDD VDD PMOS W='WP' L='LP' delvto='dxvthp1'
MNL QBD P GND GND NMOS W='WN' L='LN' delvto='dxvthn1'

* one inverter
MPR QD P VDD VDD PMOS W='WP' L='LP' delvto='dxvthp2'
MNR QD P GND GND NMOS W='WN' L='LN' delvto='dxvthn2'

* access transistors
Mr WBLB WWL QBD  GND NMOS W = 'WA' L = 'LA' delvto='dxvtha1'
Ml WBL  WWL  QD  GND NMOS W = 'WA' L = 'LA' delvto='dxvtha2'

.IC V(WBLB) = VDD
.IC V(WBL) = VDD
Vin P 0
.DC Vin LIN 30 0v VDD SWEEP MONTE=mont
.PRINT DC V(QD) V(QBD)
.OPTIONS NOMOD NOWARN POST INGOLD=2
.end
```

APPENDIX E. SNM_WRITE_8T NETLIST

```
.TEMP 80.0000
.GLOBAL VDD
.GlOBAL GND

.include 45nm_NMOS_tt.pm
.include 45nm_PMOS_tt.pm
.param mont = 50000
+ Vdd = 1.0v
+ GND = 0

.param m = 4
+ L = 45n
+ LP = 'L'
+ LN = 'L'
+ LA = 'L'
+ L7 = 'L'
+ L8 = 'L'
+ WP = 90n
+ WN = 90n
+ WA = 90n
+ W7 = 180n
+ W8 = 180n
+ UL = '-VDD/sqrt(2)'
+ UH = 'VDD/sqrt(2)'
+ BITCAP = 1E-12

*.PARAM  dxvth=agauss(0,0.1,1)
.PARAM dxvthn1 = agauss(0,'0.024*sqrt(45*67.5e-18/(WN*LN))',1)
+      dxvthn2 = agauss(0,'0.024*sqrt(45*67.5e-18/(WN*LN))',1)
+      dxvthp1=agauss(0,'-0.0292*sqrt(45*67.5e-18/(WP*LP))',1)
+      dxvthp2=agauss(0,'-0.0292*sqrt(45*67.5e-18/(WP*LP))',1)
+      dxvtha1 = agauss(0,'0.024*sqrt(45*67.5e-18/(WA*LA))',1)
+      dxvtha2 = agauss(0,'0.024*sqrt(45*67.5e-18/(WA*LA))',1)
+      dxvthn7 = agauss(0,'0.024*sqrt(45*67.5e-18/(W7*L7))',1)
+      dxvthn8 = agauss(0,'0.024*sqrt(45*67.5e-18/(W8*L8))',1)

*sources
**supply
VVDD VDD 0 dc = VDD
Vvss GND 0 dc = 0

**access control
Vwwl WWL 0 dc = VDD
Vrwl RWL 0 dc = 0

* one inverter
MPL QBD P VDD VDD PMOS W='WP' L='LP' delvto='dxvthp1'
MNL QBD P GND GND NMOS W='WN' L='LN' delvto='dxvthn1'

* one inverter
MPR QD P VDD VDD PMOS W='WP' L='LP'  delvto='dxvthp2'
MNR QD P GND GND NMOS W='WN' L='LN'  delvto='dxvthn2'

* access transistors
```

```
Mr WBLB WWL QBD  GND NMOS W = 'WA' L='LA' delvto='dxvtha1'
Ml WBL  WWL  QD  GND NMOS W = 'WA' L='LA' delvto='dxvtha2'

* read transistors
M7 QD net GND  GND NMOS W = 'W7' L = 'L7' delvto='dxvthn7'
M8 RWL  net  RBL  GND NMOS W = 'W8' L = 'L8' delvto='dxvthn8'

.IC V(WBLB) = VDD
.IC V(WBL) = 0
.IC V(RBL) = 0

Vin P 0
.dc Vin LIN 30 0v VDD SWEEP MONTE=mont

.PRINT DC V(QD) V(QBD)

.OPTIONS brief=1 NOMOD NOWARN POST INGOLD=2

.end
```

APPENDIX F. SNM_READ_8T NETLIST

```
.TEMP 80.0000
.GLOBAL VDD
.GlOBAL GND

.include 45nm_NMOS_tt.pm
.include 45nm_PMOS_tt.pm
.param mont = 50000
+ Vdd = 1.0v
+ GND = 0

.param m = 4
+ L = 45n
+ LP = 'L'
+ LN = 'L'
+ LA = 'L'
+ L7 = 'L'
+ L8 = 'L'
+ WP = 90n
+ WN = 90n
+ WA = 90n
+ W7 = 180n
+ W8 = 180n
+ UL = '-VDD/sqrt(2)'
+ UH = 'VDD/sqrt(2)'
+ BITCAP = 1E-12

*.PARAM  dxvth=agauss(0,0.1,1)
.PARAM dxvthn1 = agauss(0,'0.024*sqrt(45*67.5e-18/(WN*LN))',1)
+      dxvthn2 = agauss(0,'0.024*sqrt(45*67.5e-18/(WN*LN))',1)
+      dxvthp1=agauss(0,'-0.0292*sqrt(45*67.5e-18/(WP*LP))',1)
+      dxvthp2=agauss(0,'-0.0292*sqrt(45*67.5e-18/(WP*LP))',1)
+      dxvtha1 = agauss(0,'0.024*sqrt(45*67.5e-18/(WA*LA))',1)
+      dxvtha2 = agauss(0,'0.024*sqrt(45*67.5e-18/(WA*LA))',1)
+      dxvthn7 = agauss(0,'0.024*sqrt(45*67.5e-18/(W7*L7))',1)
+      dxvthn8 = agauss(0,'0.024*sqrt(45*67.5e-18/(W8*L8))',1)

*sources
**supply
VVDD VDD 0 dc = VDD
Vvss GND 0 dc = 0

**access control
Vwwl WWL 0 dc = 0
Vrwl RWL 0 dc = VDD
* one inverter
MPL QBD P VDD VDD PMOS W='WP' L='LP' delvto='dxvthp1'
MNL QBD P GND GND NMOS W='WN' L='LN' delvto='dxvthn1'

* one inverter
MPR QD P VDD VDD PMOS W='WP' L='LP' delvto='dxvthp2'
MNR QD P GND GND NMOS W='WN' L='LN' delvto='dxvthn2'

* access transistors
Mr WBLB WWL QBD  GND NMOS W = 'WA' L = 'LA' delvto='dxvtha1'
```

```
Ml WBL  WWL  QD  GND NMOS W = 'WA' L = 'LA' delvto='dxvtha2'

* read transistors
M7 Q net GND  GND NMOS W = 'WA' L = 'L7' delvto='dxvthn7'
M8 RWL  net  RBL  GND NMOS W = 'WA' L = 'L8' delvto='dxvthn8'


.IC V(RBL) = VDD
*.IC V(WBLB) = 0
*.IC V(WBL) = 0

*.IC V(Q)  = 0
*.IC V(QB) = VDD

Vin P 0
.DC Vin LIN 30 0v VDD SWEEP MONTE=mont

.PRINT DC V(QD) V(QBD)

.OPTIONS NOMOD NOWARN POST INGOLD=2

.end
```

APPENDIX G. PYTHON UPDATE NETLIST

```
#     SRAM SNM SIMULATION
#     YIFU GONG - 2017
#     Python script to update netlist after each simulation
#     Department: NDSU ECE Graduate Research

import sys
# main
file = str(sys.argv[1])
param_1 = int(sys.argv[2])/1000.00
param_2 = int(sys.argv[3])
param_3 = str(sys.argv[4])
param_4 = str(sys.argv[5])
param_5 = str(sys.argv[6])
param_6 = str(sys.argv[7])

with open(file, "r") as f:
    lines = f.readlines()
lines[4] = ".include "+param_6+"nm_NMOS_" + param_3 + ".pm\n"
lines[5] = ".include "+param_6+"nm_PMOS_" + param_4 + ".pm\n"
lines[6] = ".param mont = " + str(param_2) + "\n"
lines[7] = "+ Vdd = " + str(round(param_1, 2)) + "v" + "\n"
lines[10] = ".param m = " + param_5 + "\n"
lines[11] = "+ L = " + param_6 + "n\n"
with open(file, "w") as f:
    for line in lines:
        f.write(line)
```

APPENDIX H. PYTHON EXTRACT DATA

```
#      SRAM SNM SIMULATION
#      YIFU GONG - 2017
#      Two separate python script to extract meaningful data
#      Department: NDSU ECE Graduate Research

import sys
# main
file = str(sys.argv[1])
i=0
output = open("formatted_data_write.csv",'w')
with open(file) as f:
    data = f.readlines()
    for line in data:
        values = line.split()
        if "   *** monte carlo  index = " in line:
            i+=1
        if "a" not in line and "b" not in line and "c" not in line and "d"
not in line and "f" not in line and "g" not in line and "h" not in line and
"i" not in line and "j" not in line and "k" not in line and "l" not in line
and "m" not in line and "n" not in line and "o" not in line and "p" not in
line and "q" not in line and "r" not in line and "s" not in line and "t" not
in line and "u" not in line and "v" not in line and "w" not in line and "x"
not in line and "y" not in line and "z" not in line:
            if len(values) == 3:
                u = values[0]
                qd = values[1]
                qbd = values[2]
                output.write('{},{},{},{}\n'.format(i,u[:],qd[:],qbd[:]))

import sys
# main
file = str(sys.argv[1])
i=0
output = open("formatted_data_read.csv",'w')
with open(file) as f:
    data = f.readlines()
    for line in data:
        values = line.split()
        if "   *** monte carlo  index = " in line:
            i+=1
        if "a" not in line and "b" not in line and "c" not in line and "d"
not in line and "f" not in line and "g" not in line and "h" not in line and
"i" not in line and "j" not in line and "k" not in line and "l" not in line
and "m" not in line and "n" not in line and "o" not in line and "p" not in
line and "q" not in line and "r" not in line and "s" not in line and "t" not
in line and "u" not in line and "v" not in line and "w" not in line and "x"
not in line and "y" not in line and "z" not in line:
            if len(values) == 3:
                u = values[0]
                qd = values[1]
                qbd = values[2]
                output.write('{},{},{},{}\n'.format(i,u[:],qd[:],qbd[:]))
```

APPENDIX I. BUTTERFLY ROTATE COORDINATES

```
%     SRAM SNM SIMULATION
%     YIFU GONG - 2017
%     Matlab script to rotate butterfly curves
%     Department: NDSU ECE Graduate Research

clear all
warning('off','all')
data = csvread('formatted_data_read.csv');
dpt=30;
x = NaN * ones(dpt,1);
y1 = NaN * ones(dpt,1);
y2 = NaN * ones(dpt,1);
pass = 0;
fail = 0;
total = 0;
plot(0,0);hold on
axis equal

for i=1:50000
    j = (i-1)*dpt+1;
    k = 1;
    while k<=dpt && data(j,1)==i
        x(k) = data(j,2);
        y1(k) = data(j,3);
        y2(k) = data(j,4);
        j = j+1;
        k = k+1;
    end
    [xi,yi] = curveintersect(y1,x,x,y2);
    [m,n] = size(xi);
    if m==3
        if  abs(xi(2)/yi(2)-1)<0.5 && abs(yi(2)/xi(2)-1)<0.5 &&
abs((yi(1)+yi(3))/yi(2)-2)<1
            pass = pass + 1;
            total = total + 1;
            figure(1)
            plot(y1,x,'LineWidth',1);hold on
            plot(x,y1,'LineWidth',1);hold off
        end
        u1 = round(x/sqrt(2) - y1/sqrt(2), 4);
        v1 = x/sqrt(2) + y1/sqrt(2);

        uv1(1,:) = min(u1):1/10000:max(u1);
        uv1(2,:) = interp1(u1,v1,uv1(1,:));

        u2 = round(-x/sqrt(2) + y1/sqrt(2), 4);
        v2 = x/sqrt(2) + y1/sqrt(2);

        uv2(1,:) = min(u2):1/10000:max(u2);
        uv2(2,:) = interp1(u2,v2,uv2(1,:));

        figure(2)
        plot(uv1(1,:),uv1(2,:),'color',[0,0,0]+0.3,'LineWidth',4);hold on
        plot(uv2(1,:),uv2(2,:),'color',[0,0,0]+0.7,'LineWidth',4);
```

```
        u = max(min(uv1(1,:)),min(uv2(1,:))):1/10000:
min(max(uv1(1,:)),max(uv2(1,:)));
        a = (uv1(1,1) - uv2(1,1)) * 10000;
        [b,c] = size(uv1);
        if a > 0
            uv1 = uv1(:,a+1:c);
            uv2 = uv2(:,1:c-a);
        else
            uv2 = uv2(:,1:c+a);
            uv1 = uv1(:,-a+1:c);
        end

        plot(u,uv1(2,:)-uv2(2,:),'color','red','LineWidth',4);
        break
    end
end

print(gcf,'SNM.png','-dpng','-r300');
```

APPENDIX J. MATLAB SCRIPT CALCULATE WRITE FAILURE RATE

```
%     SRAM SNM SIMULATION
%     YIFU GONG - 2017
%     Matlab script to calculate write failure rate
%     Department: NDSU ECE Graduate Research

warning('off','all')
data = csvread('formatted_data_write.csv');
dpt=30;
x = NaN * ones(dpt,1);
y1 = NaN * ones(dpt,1);
y2 = NaN * ones(dpt,1);
pass = 0;
fail = 0;
total = 0;
plot(0,0);hold on
axis equal
xlim([0 1])
ylim([0 1])

for i=1:mont
    j = (i-1)*dpt+1;
    k = 1;
    while k<=dpt && data(j,1)==i
        x(k) = data(j,2);
        y1(k) = data(j,3);
        y2(k) = data(j,4);
        j = j+1;
        k = k+1;
    end
    [xi,yi] = curveintersect(y1,x,x,y2);
    [m,n] = size(xi);
    if m==1
        pass = pass + 1;
        total = total + 1;
    else
        fail = fail + 1;
        total = total + 1;
    end
end
passPercentage = pass/total;
failPercentage = fail/total;
fileID = fopen(file,'a');
fprintf(fileID,',%i,%d\n',v/1000,failPercentage);
fclose(fileID);
clc
clear all
exit
```

APPENDIX K. MATLAB SCRIPT CALCULATE READ FAILURE RATE

```
%     SRAM SNM SIMULATION
%     YIFU GONG - 2017
%     Matlab script to calculate read failure rate
%     Department: NDSU ECE Graduate Research

warning('off','all')
data = csvread('formatted_data_read.csv');
dpt=30;
x = NaN * ones(dpt,1);
y1 = NaN * ones(dpt,1);
y2 = NaN * ones(dpt,1);
pass = 0;
fail = 0;
total = 0;
plot(0,0);hold on
axis equal
xlim([0 1])
ylim([0 1])

for i=1:mont
    j = (i-1)*dpt+1;
    k = 1;
    while k<=dpt && data(j,1)==i
        x(k) = data(j,2);
        y1(k) = data(j,3);
        y2(k) = data(j,4);
        j = j+1;
        k = k+1;
    end
    [xi,yi] = curveintersect(y1,x,x,y2);
    [m,n] = size(xi);
    if m==3
        if  abs(xi(2)/yi(2)-1)<0.5 && abs(yi(2)/xi(2)-1)<0.5 &&
abs((yi(1)+yi(3))/yi(2)-2)<1
            pass = pass + 1;
            total = total + 1;
        else
            fail = fail + 1;
            total = total + 1;
        end
    else
        fail = fail + 1;
        total = total + 1;
    end
end
passPercentage = pass/total;
failPercentage = fail/total;
fileID = fopen(file,'a');
fprintf(fileID,'%i,%d',v/1000,failPercentage);
fclose(fileID);
clc
clear all
exit
```

APPENDIX L. MATLAB SCRIPT COMPARE PSNR

```
%     COMPARE PSNR
%     YIFU GONG - 2018
%     Matlab script to compare PSNR at 0s and 10 situation
%     Department: NDSU ECE Graduate Research

function [p_sure,p_random] = check_psnr
files = dir(fullfile('tmp\video*.yuv'));
for j = 1:5
    for i = 1:5
        temp = Fixed_trunc_sure(strcat(files(j).folder, '\',files(j).name),
strcat(files(j).folder, '\','trunc_',files(j).name), 320, 240,
(1:50),'3',i,100/100);
        p_sure(i,j) = temp;
    end
end

for j = 1:5
    for i = 1:5
        temp = Fixed_trunc_random(strcat(files(j).folder, '\',files(j).name),
strcat(files(j).folder, '\','trunc_',files(j).name), 320, 240,
(1:50),'3',i,100/100);
        p_random(i,j) = temp;
    end
end

figure(1)
plot(p_sure)
hold on
t = 0:4;!
Le = 20*log10(255)-10*log10((4.^(t+1 )-1)/3);
plot(t+1,Le,'b','LineWidth',5)

figure(2)
plot(p_random)
hold on
t = 0:4;
Le = 20*log10(255)-10*log10((4.^(t+1 )-1)/6);
plot(t+1,Le,'b','LineWidth',5)
```

Note

Some of the data in this dissertation was obtained by Yifu Gong prior to his inclusion on the approved NDSU IRB protocol for this research project. Mr. Gong had no knowledge of this error and bears no responsibility for it. The NDSU Graduate School has therefore elected to publish Mr. Gong's dissertation in fulfillment of his requirements for graduation.