

POWER-EFFICIENT ADAPTIVE MEMORY DESIGN AND OPTIMIZATION FOR VIDEO  
AND DEEP LEARNING

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

By  
Hritom Das

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

Major Department:  
Electrical and Computer Engineering

November 2020

Fargo, North Dakota

North Dakota State University  
Graduate School

---

**Title**

POWER-EFFICIENT ADAPTIVE MEMORY DESIGN AND  
OPTIMIZATION FOR VIDEO AND DEEP LEARNING

---

**By**

Hritom Das

---

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. Scott C. Smith (Co-Advisor)

---

Chair

Dr. Danling Wang (Co-Advisor)

---

Dr. Dharmakeerthi Nawarathna

---

Dr. Jacob Glower

---

Dr. Mingao Yuan

---

Approved:

November 25, 2020

---

Date

Dr. Benjamin Braaten

---

Department Chair

## ABSTRACT

Memory devices such as Static Random-Access Memory (SRAM) and Dynamic Random-Access Memory (DRAM) are dominating members of today's semiconductor industry. Most of the silicon area in a digital system is occupied by memory devices. The video decoder and deep learning are especially constrained by memory devices to process a large amount of data. For example, memory devices are consuming lots of power for video processing. Nowadays, all mobile electronics, such as mobile phones and laptops, are using video data a lot. Due to that, the battery life of mobile devices is highly dependent on power consumption of memory devices. To enhance the battery life of mobile devices, supply voltage can be scaled down. However, memory devices are error prone at low supply voltages. To obtain high quality video, a functionally stable memory design is needed, which means we must provide a higher  $V_{DD}$  or use a larger memory cell. As a result, there will be a tradeoff between quality, and silicon area or power consumption. For mobile devices, memory needs to be designed to operate in the sub-threshold region to maximize battery life; however, reducing the supply voltage slows down memory devices, resulting in poor video quality. Hence, memory design is very complicated and time consuming. So, a smart way to design memory devices for a specific application is needed. Mathematical models can be developed to design memory devices based on specific requirements such as silicon area, while optimizing video quality for a target supply voltage. Similarly, optimized memory is needed to better support differentially private deep learning algorithms in local devices. This dissertation first develops a mathematical model for designing optimal memory devices for videos, then develops an optimized memory for differentially private deep learning systems in edge computing devices, and finally develops a run-time

adaptable Error Correction Code (ECC) video storage scheme, with minimal area overhead and negligible video quality degradation, in order to significantly reduce power.

## ACKNOWLEDGEMENTS

The accomplishment of this dissertation would not have been possible at all without the help and support of many people. First, I would like to express my deepest gratitude to my supervisors Dr. Na Gong, Dr. Scott Smith, and Dr. Danling Wang for the chance to work on an exciting research topic and for their support and guidance with patience. I would like to thank the ECE department and research park-2 of North Dakota State University for providing an ideal environment for my graduate studies here.

I would also like to thank my other committee members, Dr. Dharmakeerthi Nawarathna, Dr. Jacob Glower, and Dr. Mingao Yuan for their feedback and assistance in presentations, papers and other advice given during my doctoral study. I am thankful for all the help provided by the other members of my lab, especially Dr. Yifu Gong, Dr. Jonathon David Edstrom, and Ali Haidous. Their help with preparing experiments, calculating simulation results, and verifying designs was paramount in completing the necessary work for conferences, journals, and the completion of my doctoral requirements.

In reference to IEEE copyrighted material, which is used with permission in this dissertation, the IEEE does not endorse any of North Dakota State University's products or services. Internal or personal use of this material is permitted. I am also thankful for the National Science Foundation (NSF) grant, CCF-1855706, which supported my research work.

## **DEDICATION**

I dedicate this dissertation to my parents, Birendra Nath Das and Bela Das.

## TABLE OF CONTENTS

ABSTRACT.....	iii
ACKNOWLEDGEMENTS.....	v
DEDICATION.....	vi
LIST OF TABLES.....	x
LIST OF FIGURES.....	xi
LIST OF ABBREVIATIONS.....	xiii
1. INTRODUCTION.....	1
1.1. Motivation.....	1
1.2. Memory Design for Various Purposes.....	1
2. ON MATHEMATICAL MODELS OF OPTIMAL VIDEO MEMORY DESIGN <sup>1</sup> .....	5
2.1. Introduction.....	5
2.2. The Expected Mean-Square Error.....	6
2.3. Model 1: Optimal Design for Single SRAM.....	8
2.3.1. The Mathematical Model.....	8
2.3.2. Numerical Study of Model 1.....	10
2.4. Model 2: Optimal Design for Hybrid SRAM without Overhead.....	14
2.4.1. The Mathematical Model.....	15
2.4.2. Numerical Study of Model 2.....	16
2.5. Model 3: Optimal Design for Hybrid Memory with Various Technologies.....	19
2.5.1. Integration Cost of SRAM and DRAM.....	20
2.5.2. The Mathematical Model.....	22
2.5.3. Numerical Study of Model 3.....	25
2.6. Discussion.....	28
2.6.1. Relationship of the Three Developed Models.....	28

2.6.2. Comparison with Prior Work .....	29
<b>3. MEMORY OPTIMIZATION FOR ENERGY-EFFICIENT DIFFERENTIALLY PRIVATE DEEP LEARNING<sup>2</sup> .....</b>	<b>31</b>
3.1. Introduction .....	31
3.2. Learning with Differential Privacy .....	32
3.2.1. Privacy Preservation in Deep Learning .....	32
3.2.2. Differentially Private Deep Learning and State of the Art.....	33
3.3. Impact of Memory Failures in Differentially Private Deep Learning Systems .....	36
3.3.1. Impact of Image Quality on Classification Accuracy .....	37
3.3.2. Protecting Most Significant Bits (MSBs) of Data.....	38
3.3.3. Impact of Memory Failure on Privacy/Accuracy Trade-off.....	40
3.3.4. Integer Linear Programs (ILP) Model based Memory Design.....	41
3.4. Embedded Memory Design for Deep Learning .....	42
3.4.1. Optimized Memory Design .....	43
3.4.2. Power Efficiency .....	46
3.4.3. Input Data Quality and Accuracy .....	47
3.4.4. Accuracy at Different Privacy Levels .....	50
<b>4. FLEXIBLE LOW-COST POWER-EFFICIENT VIDEO MEMORY WITH ECC-ADAPTATION.....</b>	<b>51</b>
4.1. Introduction .....	51
4.2. State-of-the-Art .....	51
4.2.1. Video-Specific Memory with Design-Time Fixed Quality.....	52
4.2.2. Adaptive Memory with Dynamic Power-Quality Management .....	52
4.3. Proposed Low-Cost ECC Storage Scheme .....	53
4.3.1. Traditional ECC.....	53



4.3.2. Bit Significance Characteristics of Video Data and Proposed Storage Scheme for Parity Bits .....	57
4.4. ECC Adaptation Based on Requirements and Failure Rate Based on Voltage Scaling.....	60
4.4.1. Failure Characteristics of 6T SRAM.....	60
4.4.2. Errors Injected, Including in Parity Bits.....	61
4.4.3. ECC Under Various Failure Rates .....	63
4.4.4. Proposed Runtime ECC Adaptation Scheme .....	64
4.5. Proposed Memory .....	65
4.5.1. Reusable ECC Encoder for ECC1511 and ECC74 .....	67
4.5.2. Reusable ECC Decoder for ECC1511 and ECC74 .....	68
4.5.3. Correction Unit.....	69
4.5.4. Output MUX.....	70
4.6. Results .....	71
4.6.1. Timing Diagram .....	71
4.6.2. Power Efficiency .....	72
4.6.3. Video Quality .....	74
5. CONCLUSIONS AND FUTURE WORK.....	75
REFERENCES .....	77

## LIST OF TABLES

<u>Table</u>		<u>Page</u>
1.	Data of Numerical Example 1 ( $VDD = 0.75V$ ) .....	10
2.	Statistics of the $q_k(s_k)$ Fitting in Numerical Example 1 .....	12
3.	Results and Comparisons of Numerical Example 1 ( $VDD = 0.75V$ ) .....	12
4.	Data of The Numerical Example 2 ( $VDD = 0.5V$ ) .....	17
5.	Results and Comparisons of the Numerical Example 2 ( $VDD = 0.5V$ ) .....	18
6.	6T and 8T SRAM Data of Numerical Example 3 ( $VDD = 0.5v$ ), with Area-Overhead Caused by 3T DRAM .....	25
7.	3T DRAM Data for Numerical Example 3 ( $VDD = 0.5v$ ) .....	26
8.	Results and Comparisons of Numerical Example 3 ( $VDD = 0.5V$ ) .....	26
9.	Memory Failure Rate .....	44
10.	Results and Comparisons of Proposed Memory for Deep Learning .....	46
11.	Power Consumption of Optimized Memory at 45nm CMOS Technology @ 0.5V .....	47
12.	Input Data Quality and Accuracy .....	49
13.	At Particular Privacy Parameters, the Impact of Privacy Level on Test Accuracy .....	50
14.	ECC Sequence and Message Bit Placement for Traditional ECC74 and ECC1511 .....	55
15.	Impact of Traditional ECC (Parity Bit Position) on Video Storage .....	56
16.	Proposed ECC (Message Bit and Parity Bit Placement) .....	58

## LIST OF FIGURES

<u>Figure</u>	<u>Page</u>
1. Monte Carlo-Based Failure Rate vs. Predicted Failure Rate, Numerical Example 1.....	11
2. Comparison of the Optimal Design and the Traditional Design, Numerical Example 1 .....	13
3. Layout of the Optimal and Traditional Design, Numerical Example 1 .....	13
4. Video Output Example for $stotal = 9.6$ from Table 3., Numerical Example 1( $VDD = 0.75V$ ): (A) Original Video; (B) Video Stored by the Optimal Memory Design; (C) Video Stored by the Traditional Memory Design.....	14
5. Comparison of the Optimal Design and the Traditional Design, Numerical Example 2 .....	18
6. Layout of the Optimal and Traditional Design, Numerical Example 2 .....	19
7. Video Output Quality Example for $stotal = 8.7$ from Table 5, Numerical Example 2 ( $VDD = 0.5V$ ): (A) Original Video; (B) Video Stored by the Optimal Design; (C) Video Stored by the Traditional Design.....	19
8. Refresh Period Enabled Energy-Quality Adaptation for DRAM.....	21
9. Schematic and Layout of 3T DRAM, 6T and 8T SRAM Bitcells Compatible with DRAM.....	22
10. Comparison of the Optimal and Traditional Design of Numerical Example 3, and Part of the Optimal Design of Numerical Example 2 (For Both $VDD = 0.5V$ ) .....	26
11. Layout of the Optimal and Traditional Design, $stotal = 8.0$ , Numerical Example 3.....	27
12. Video Output Quality Example For $stotal = 8.0$ and $8.4$ From Table 8, Numerical Example 3 ( $VDD = 0.5V$ ): (A) Original Video; (B) Video Stored by the Optimal Design, $stotal = 8.0$ ; (C) Video Stored by the Traditional Design, $stotal = 8.0$ ; (D) Video Stored By the Optimal Design, $stotal = 8.4$ ; (E) Video Stored by the Traditional Design, $stotal = 8.4$ .....	28
13. Relationship of the Three Models, Which Covers Different Mathematical Approach for Different Memory Technology.....	29
14. Input Data Memory Optimization for Deep Learning System with Energy-Efficiency, Privacy, and Inference Accuracy .....	35

15.	Differentially Private Convolutional Neural Network Used in Our Analysis.....	36
16.	Influence of Dataset Quality on Test Accuracy (Using MNIST Dataset).....	38
17.	Impact of Memory Failure Rate on the Accuracy of the Learning System. ....	39
18.	Impact of Memory Failure Rate on Privacy/Accuracy Tradeoff: (A) Without MSB Protected And (B) With 2 MSBs Protected. ....	39
19.	Different Memory Designs: (A) 6T SRAM Schematic and Minimum-Sized Layout Design in 45 nm Technology (C61) and (B) 8T SRAM Schematic and Minimum-Sized Layout Design (C81) in the Same 45 nm Technology. ....	44
20.	Video Output Quality with Traditional ECC. (A) Original Frame, (B) ECC74 Parity Bits Stored with PSNR = 27.75 dB, and (C) ECC1511 Parity Bits Stored with PSNR = 8.27 dB. ....	56
21.	Encoded Video Frame (Akiyo) with (a) ECC74 Where Parity was Stored in the LSBs with PSNR = 41.2582 and (b) ECC1511 Where Parity Bits were Stored in LSBs with PSNR = 39.8426 dB.....	59
22.	Relation Between Supply Voltage (VDD) and SRAM bitcell Failure Rate in a 45nm CMOS Technology. ....	61
23.	Error Map and Stored Video Frame with Proposed ECC74 Under 0.1% Faulty Memory Bitcells.....	63
24.	Error Map and Stored Video Frame with Proposed ECC1511 Under 0.1% Faulty Memory Bitcells.....	63
25.	ECC Adaptation Based on Failure Rate and Corresponding PSNR. ....	65
26.	Proposed Adaptive ECC Memory. ....	67
27.	ECC Encoder.....	68
28.	ECC Decoder.....	69
29.	Correction Unit.....	70
30.	Output MUX.....	70
31.	Timing Diagram. ....	71
32.	Power Comparison of Proposed ECC Memory with Traditional Memory.....	73
33.	PSNR Values of 100 Videos at 0.1% and 0.9% Failure Rates.....	74

## LIST OF ABBREVIATIONS

CMOS .....	Complementary Metal-Oxide-Semiconductor
SRAM .....	Static Random-Access Memory
DRAM .....	Dynamic Random-Access Memory
NVM .....	Non-Volatile Memory
ECC .....	Error Correcting Code
LSB .....	Least Significant Bit
MSB .....	Most Significant Bit
IoT .....	Internet of Things
PSNR.....	Peak Signal-To-Noise Ratio
NMOS .....	N-Type Metal-Oxide-Semiconductor
PMOS .....	P-Type Metal-Oxide-Semiconductor
VDD .....	Supply Voltage
MUX .....	Multiplexer
XOR.....	Exclusive OR
MSE .....	Mean Squared Error
CNN .....	Convolutional Neural Network
NLP .....	Non-linear Programs
ILP .....	Integer Linear Programs

# 1. INTRODUCTION

## 1.1. Motivation

Memory devices are very instrumental components of modern electronics: such as, mobile phones, laptops, cameras, servers, smart watches, and so on. There are several video streaming sites such as YouTube, Netflix, and Hulu. These video streaming sites generate a huge amount of video data every single day. These immense amounts of data create pressure on storage memory systems. To store and process video data, a memory system is required that is functionally stable, energy efficient, and economical in fabrication. There are lots of issues in memory devices such as aging and process variation. Due to these two reasons, memory shows functional instability, which is known as failure rate in a memory system. To minimize memory failure rates, researchers utilize different techniques, which add cost to these systems. Researchers strive to optimize power consumption of mobile multimedia applications, because the embedded memory is frequently accessed for motion estimation and buffering for video processing. These two issues are the main reasons for high power consumption in mobile devices [1]. Scaling supply voltage is one of the main attempts in VLSI design to reduce power consumption [2], since power is proportional to the square of the supply voltage; however, this also has some drawbacks. If the supply voltage is over-scaled, the memory system will be functionally weak. Nowadays, video is one of the biggest of Big Data [3-4]. According to Cisco, video data is predicted to comprise around 78% of all data by 2021 [3].

## 1.2. Memory Design for Various Purposes

Embedded memory plays a vital role in processing video data. About 65% of the silicon area of a video decoder is occupied by embedded memory [5], and memory accounts for approximately 50% of total power consumption [6]. Traditional or homogeneous memory design

shows higher failure rates compared to heterogeneous sizing of memory [7]. SRAM is one of the most mature and highly preferable memories in the industry. Because of its high speed and reliability, SRAM's demand is always high. Hybrid memory design such as 6T + 8T SRAM or 8T + 10T SRAM show better performance as a trade-off to quality and power [8-9]. Recently, DRAM has become more popular than SRAM in some specific fields, due to its smaller area and lower power consumption for reading and writing. DRAM is also being integrated with SRAM to achieve better performance in hybrid memory schemes. Higher order bits are stored in SRAM and lower order bits are stored in error prone DRAM [10].

Another hot topic is privacy of deep learning in Internet of Things (IoT) devices. By the blessing of high-speed internet, IoT devices, such as smart watches, are gaining popularity day by day. Smart watches can collect cardiac activities, blood pressure, and sugar level by sensors [25-26]. This data can be used for deep learning to monitor health conditions of a user. However, it is not guaranteed that the data would be fully private during analysis and sharing with healthcare industries. Usually, users need to upload their private health data to the providers, and thus have no control over its storage or usage [27-29]. There are several ways to provide protection to private data for edge computing, including differential privacy. Differential privacy is achieved by adding noise during computations in deep learning algorithms [31]. Due to the added noise, the output cannot be correlated with any particular training item. During the introduction of noise to the computation, the privacy budget ( $\epsilon$ ) is cost. Here,  $\epsilon$  stands for the privacy loss in a system. If  $\epsilon$  is smaller, then privacy is higher, but accuracy of the model is lower. Moreover, memory consumes a lot of power in edge computing devices. In AlexNet, approximately 3000M memory accesses are required for the whole learning system [32]. Moreover, in DianNao, 56% of the chip area is occupied by memory, which consumes 60% of

the total power [33]. Due to the huge demand of memory devices in IoT systems, developing power efficient memory to support differentially private deep learning systems is much needed.

Memory devices are error prone. Due to aging and process variation, faulty bits are introduced during reading and writing operations. To solve this error in memory devices, Error Correction Code (ECC) is very popular [34]. From the software side, ECC is a mature and old technique, thus there is no real novelty potential. However, hardware implementation of ECC is both economical and has great potential for novelty. ECC can be used for runtime adaptation for different scenarios. There are different types of ECC, such as ECC74, ECC1511, and so on. According to the requirement of a system, if the memory can adapt the ECC, then the system can save power and improve the quality of processed data due to optimistic voltage scaling that introduces some errors, but ones that can be easily corrected by ECC.

Bit truncation is one of the most popular methods to achieve energy efficient memory design, especially for video memory where the least significant bits (LSBs) have much less effect on video quality compared to the most significant bits (MSBs). Depending on the specific video data and the viewing surroundings (e.g., sunny/dark/overcast), one or more bits can be truncated without degrading video quality as perceived by the viewer. A luminosity sensor can provide information about the surroundings, such that the memory could adapt the number of truncated bits accordingly. Bit truncation can be applied frame by frame, such that more bits can be truncated due to the variation of frame data and still achieve acceptable video quality [35]. The truncation information could be stored on the server, such that the server could perform bit truncation in the transmitted data. Bit truncation has an enormous potential to provide better a power quality tradeoff for mobile video applications.



The rest of this dissertation is organized as follows. Chapter 2 presents a mathematical model for memory design that optimizes video quality given a specific are requirement. Chapter 3 develops a method to design hybrid memory for differentially private deep learning in IoT devices in order to guarantee user data privacy while maintaining classification accuracy and reducing power via supply voltage scaling. Chapter 4 develops a memory that automatically adapts its ECC scheme based on supply voltage in order to significantly reduce power while maintaining good video quality. And Chapter 5 provides conclusions and directions for future work.

## 2. ON MATHEMATICAL MODELS OF OPTIMAL VIDEO MEMORY DESIGN<sup>1</sup>

### 2.1. Introduction

Big video data today imposes huge pressures on storage. The variation and aging induced memory failures significantly influence the video output quality. Recently, researchers have developed different memory designs for videos, deep learning, and other data-intensive applications, which enable better energy-quality tradeoffs with design constraints (e.g. silicon area of die). Unfortunately, designing memory has been proven to be a very challenging problem due to (i) various design constraints; (ii) multiple memory bit-cell design options; and (iii) challenging layout integration and cost analysis using different memory technologies. This chapter develops novel mathematical models for optimizing embedded video memory design without utilizing a time-consuming and laborious ASIC design process. The problems are formulated as nonlinear programs and integer linear programs. Different SRAM designs and hybrid SRAM and DRAM designs are considered in the models. The results of the numerical studies show that by applying the proposed methods, the average mean-square-error (MSE) of the video storage can be greatly reduced, by more than 90% in many cases.

The main contributions of this work are: three mathematical models and memory system development for 6T SRAM, 8T SRAM, and 3T DRAM, to optimize design cost, such as silicon area. In addition, video quality is improved significantly with lower MSE.

---

<sup>1</sup>**Hritom Das** was in charge of all memory (SRAM, DRAM) system design and simulation in Cadence, failure rate calculation, MSE calculation for different videos, layout design, area overhead calculation, and video quality verification. Drs. Na Gong, Yiwen Xu, and Yifu Gong provided the modeling and simulation support.

## 2.2. The Expected Mean-Square Error

For video hardware designers, the mean square error (MSE) is a widely-adopted quantity to measure the quality of a video [7-10]. Consider a video including  $m$ -by- $n$  pixels where each pixel is composed of 8 cells. Let  $y_{ijk}^{(O)}$  and  $y_{ijk}^{(D)}$  denote the binary data of the  $k^{\text{th}}$  cell in the  $i^{\text{th}}$  row  $j^{\text{th}}$  column ( $k = 0, \dots, 7; i = 1, \dots, m; j = 1, \dots, n$ ) of the pixel of the original and degraded video, respectively. The MSE is defined by

$$MSE = \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n \left( y_{ij}^{(D)} - y_{ij}^{(O)} \right)^2$$

where the degradations are caused by hardware memory failures, and

$$y_{ij}^{(O)} = \sum_{k=0}^7 y_{ijk}^{(O)} \text{ and } y_{ij}^{(D)} = \sum_{k=0}^7 y_{ijk}^{(D)}.$$

However, if we consider videos in general, rather than a specific video, the MSE should be considered as a random variable. This is because, first, memory failures caused by process variations are random in nature; and second, the MSE depends on the original video signal which is also random (or, video-wised). We replace  $y_{ijk}^{(O)}$  and  $y_{ijk}^{(D)}$  by two random variables:  $Y_{ijk}^{(O)}$  and  $Y_{ijk}^{(D)}$ . Denote the probability that the  $ijk^{\text{th}}$  bitcell is failed as  $q_{ijk}$ , and we assume that the status of cells are mutually independent. Let

$$X_{ijk} := Y_{ijk}^{(D)} - Y_{ijk}^{(O)}.$$

Clearly, the distribution of  $X_{ijk}$  is

$$P\{X_{ijk} = 0\} = 1 - q_{ijk} := p_{ijk},$$

$$P\{|X_{ijk}| = 1\} = q_{ijk}.$$

**Property 1.** The expectation of the mean square error (MSE) of the video is

$$E(MSE) = \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n \sum_{k=0}^7 4^k q_{ijk}$$

*Proof.*

$$\begin{aligned} E(MSE) &= E \left[ \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n \left( Y_{ij}^{(D)} - Y_{ij}^{(O)} \right)^2 \right] \\ &= E \left\{ \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n \left[ \sum_{k=0}^7 2^k \left( Y_{ijk}^{(D)} - Y_{ijk}^{(O)} \right) \right]^2 \right\} \\ &= \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n E \left[ \sum_{k=0}^7 2^k X_{ijk} \right]^2 \end{aligned}$$

Note that

$$\begin{aligned} &E \left[ \sum_{k=0}^7 2^k X_{ijk} \right]^2 \\ &= E \left[ \sum_{k=0}^7 4^k X_{ijk}^2 \right] + E \left( \sum_{k_1=0}^7 \sum_{k_2=0, k_2 \neq k_1}^7 2^{k_1+k_2} X_{ijk_1} X_{ijk_2} \right) \\ &= \sum_{k=0}^7 4^k E(X_{ijk}^2) + \sum_{k_1=0}^7 \sum_{k_2=0, k_2 \neq k_1}^7 2^{k_1+k_2} E(X_{ijk_1} X_{ijk_2}) \\ &= \sum_{k=0}^7 4^k q_{ijk} + \sum_{k_1=0}^7 \sum_{k_2=0, k_2 \neq k_1}^7 2^{k_1+k_2} E(X_{ijk_1}) E(X_{ijk_2}) \quad (1) \\ &= \sum_{k=0}^7 4^k q_{ijk} \quad (2) \end{aligned}$$

(1) holds since,  $X_{ijk}$ 's are mutually independent, and (2) holds because, without losing generality, we assume that

$$P\{X_{ijk} = 1\} = P\{X_{ijk} = -1\} = \frac{q_{ijk}}{2}, \quad \forall k = 0, \dots, 7 \quad \blacksquare$$

The proposed expected MSE will be used as the objective function in this paper to optimize hardware-design, as discussed in the following sections. It is worth mentioning that in the

above discussion we assume that the number of bitcells in a pixel is 8. In real applications, it can also be considered as a parameter  $l$  by simply modifying  $k = 0, \dots, 7$  to  $k = 0, \dots, l - 1$ , and the above conclusion still holds.

### 2.3. Model 1: Optimal Design for Single SRAM

SRAM has been the workhorse for embedded memory design, including video applications, for several decades. In many SRAM design problems, the target supply voltage ( $V_{DD}$ ) is an engineering specification and hence can be considered as a known parameter. Reducing  $V_{DD}$  enables an enhanced power efficiency, but meanwhile SRAM failure rate gets significantly increased. Specifically, memory failures are very sensitive to process variations at a low  $V_{DD}$ . In our analysis, to obtain the failure rates of different memory bitcells, 100,000 HSPICE Monte-Carlo simulations are performed in the worst process corners: read failures of 6T bitcells in “fast NMOS and slow PMOS” (FS) corner and write failures of 8T bitcells in “slow NMOS and fast PMOS” (FS) corner” [8], [11].

Under the target  $V_{DD}$ , in some cases one can fit the function of the failure rate of the  $ijk^{th}$  bitcell,  $q_{ijk}$  and its silicon area  $s_{ijk}$ . In many real applications, the function  $q_{ij}(s_{ij})$  can be fitted by

$$q_{ijk} = \exp(-\alpha s_{ijk} + \beta) \quad (3)$$

#### 2.3.1. The Mathematical Model

Suppose the function  $q_{ij}(s_{ij})$  is known. We formulate the optimal design problem for single SRAM as the follows.

$$[M1] \quad \min_s \sum_{i=1}^m \sum_{j=1}^n \sum_{k=0}^7 4^k q_{ijk}(s_{ijk}) \quad (4)$$

$$\text{s.t.} \quad \sum_{i=1}^m \sum_{j=1}^n \sum_{k=0}^7 s_{ijk} \leq s_{total} \quad (5)$$

$$s_{ijk} \geq s_{min}, \quad \forall i, j, k \quad (6)$$

The objective function (4) is to minimize the expected MSE of the whole SRAM, without considering the constant coefficient  $\frac{1}{mn}$ . Constraint (5) assures that the total silicon area should be no more than a given constant,  $s_{total}$ . In real applications this constraint can represent the resource limit in terms of silicon area, budget, or performance, etc., although in [M1] only the silicon area is included. Constraint (6) gives the minimum area of making a bitcell,  $s_{ijk}$ , which in this chapter is derived from a 45-nm CMOS technology [12].

It is worth mentioning that in typical hardware designs, to avoid significant implementation cost, all pixels are identical; that is, the memory design for storing one pixel is the same as that storing another pixel. Thus, we can cancel the  $ij$  indices in [M1] and simplify the model.

Table 1. Data of Numerical Example 1 ( $V_{DD} = 0.75V$ )

6T only	Height ( $\mu m$ )	Width ( $\mu m$ )	Area ( $\mu m^2$ )	Area Ratio $s_k$	Failure rate $q_k$
C61	0.45	1.52250	0.68513	1.00000	0.172400
C62	0.45	1.60250	0.72113	1.05255	0.110000
C63	0.45	1.69750	0.76388	1.11494	0.066500
C64	0.45	1.75757	0.79091	1.15440	0.052200
C65	0.45	1.84750	0.83138	1.21346	0.034230
C66	0.45	1.93753	0.87189	1.27260	0.022225
C67	0.45	2.00750	0.90338	1.31856	0.015450
C68	0.45	2.08750	0.93937	1.37110	0.009545
C69	0.45	2.14750	0.96638	1.41051	0.006740
C610	0.45	2.21750	0.99788	1.45649	0.004415
C611	0.45	2.28750	1.02938	1.50246	0.002640
C612	0.45	2.35750	1.06088	1.54844	0.001470
C613	0.45	2.43750	1.09688	1.60099	0.000790
C614	0.45	2.51750	1.13288	1.65353	0.000466
C615	0.45	2.58750	1.16438	1.69951	0.000300
C616	0.45	2.66750	1.20038	1.75205	0.000146
C617	0.45	2.73750	1.23188	1.79803	0.000060
C618	0.45	2.82750	1.27238	1.85714	0.000020
C619	0.45	2.89750	1.30388	1.90312	0.000010
C620	0.45	2.97750	1.33988	1.95567	0.000003
C621	0.45	3.05750	1.37588	2.00821	0.000002

### 2.3.2. Numerical Study of Model 1

Consider Table 1, where the data of a 6T SRAM design under  $V_{DD}=0.75V$ . Assume that all pixels are identical. For convenience, we use the area ratio shown in the 5th column (based on the smallest design, C61), instead of the original area, to define  $s_k$ . For example, the area ratio of C62= $1.05255 = 0.72113/0.68513$ . Thus, we can set  $s_{\min}=1$  in (6).

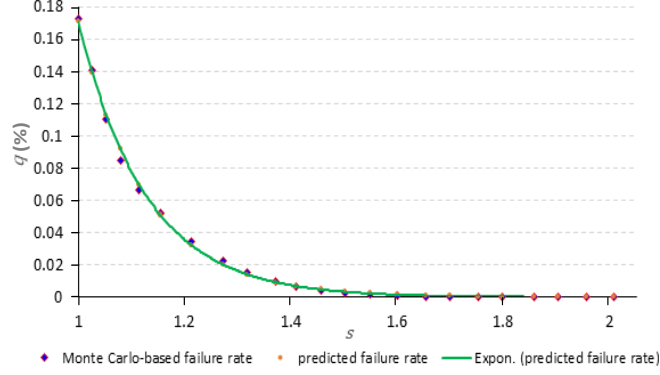


Figure 1. Monte Carlo-Based Failure Rate vs. Predicted Failure Rate, Numerical Example 1.

It should be noted that the slowest memory bitcell (with the smallest silicon area) in our design still meets the multi-megahertz performance requirement of video applications [5].

We fit  $q_k(s_k)$  according to (3) by MATLAB curve fitting toolbox (based on nonlinear least square method), and get

$$q_k = \exp(-7.834s_k + 6.065).$$

Table 2 shows the statistics of the fitting, and Figure 1 shows the graphical comparison between the failure rate (using HSPICE Monte Carlo simulations) and the predicted data.

The [M1] of this numerical study is formulated as

$$\begin{aligned} \min_s \quad & \sum_{k=0}^7 4^k \exp(-7.834s_k + 6.065) \\ \text{s.t.} \quad & \sum_{k=0}^7 s_k \leq s_{total} \\ & s_k \geq 1, \quad \forall k \end{aligned}$$

We solve the problem for  $s_{total} = 8.0, 8.2, \dots, 9.4$  using MOSEK solver. The optimal objective values and solutions under these  $s_{total}$ 's are shown in Table 3, from column 2 to column 10. The computation time of the optimal solution for each  $s_{total}$  case is less than 0.1



second. First, one can see that with the increase of  $s_{total}$  the optimal values (i.e., the minimum expected MSE) of the problem decrease exponentially, as shown in Figure 2. Second, we always have  $s_7 \geq s_6 \geq \dots \geq s_0$ , no matter what the  $s_{total}$  is. This makes sense, since the highest-order bit ( $s_7$ ) is the most significant in a pixel while the lowest-order is the least significant. We also compare our optimal design with the traditional design (where  $s_k = s_{total}/8, \forall k$ , i.e., all bitcells have the same area, according to [13]) at each  $s_{total}$  value. One can see that for  $s_{total} \geq 8.8$  the proposed optimal design can reduce the expected MSE by more than 80% compared with the traditional design.

Table 2. Statistics of the  $q_k(s_k)$  Fitting in Numerical Example 1

Parameter	95% confidence interval	Mean	R-square	Adjusted R-square	SSE	RMSE
$\alpha$	(7.632, 8.036)	7.834	99.91%	99.90%	$3.571 \times 10^{-5}$	0.0014
$\beta$	(5.854, 6.275)	6.065				

Table 3. Results and Comparisons of Numerical Example 1 ( $V_{DD} = 0.75V$ )

$s_{total}$	Optimal design									Traditional design		Improvement
	Obj. value	$s_7$	$s_6$	$s_5$	$s_4$	$s_3$	$s_2$	$s_1$	$s_0$	Obj. value	$s_k$	
8.0	3724.7	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	3724.65	1.0	0.00%
8.2	1509.0	1.2	1.0	1.0	1.0	1.0	1.0	1.0	1.0	3062.17	1.0	50.72%
8.4	815.8	1.3	1.1	1.0	1.0	1.0	1.0	1.0	1.0	2517.51	1.1	67.60%
8.6	495.4	1.4	1.2	1.0	1.0	1.0	1.0	1.0	1.0	2069.74	1.1	76.06%
8.8	317.5	1.4	1.3	1.1	1.0	1.0	1.0	1.0	1.0	1701.61	1.1	81.34%
9.0	212.0	1.5	1.3	1.2	1.0	1.0	1.0	1.0	1.0	1398.95	1.1	84.85%
9.2	147.7	1.6	1.4	1.2	1.0	1.0	1.0	1.0	1.0	1150.13	1.2	87.16%
9.4	104.5	1.6	1.4	1.3	1.1	1.0	1.0	1.0	1.0	945.56	1.2	88.95%
<b>9.6</b>	<b>75.34</b>	<b>1.7</b>	<b>1.5</b>	<b>1.3</b>	<b>1.1</b>	<b>1.0</b>	<b>1.0</b>	<b>1.0</b>	<b>1.0</b>	<b>777.38</b>	<b>1.2</b>	<b>90.31%</b>
9.8	55.60	1.7	1.5	1.4	1.2	1.0	1.0	1.0	1.0	639.11	1.2	91.30%

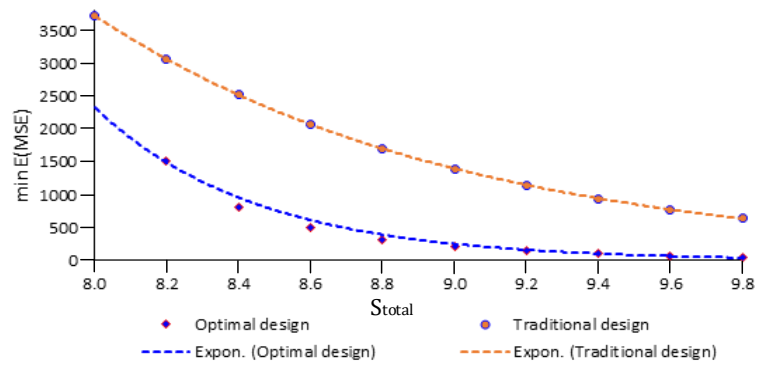
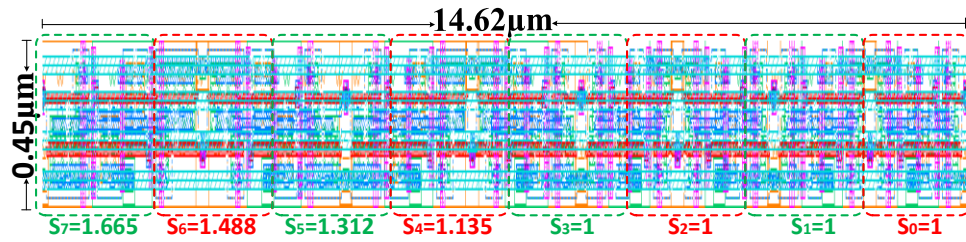
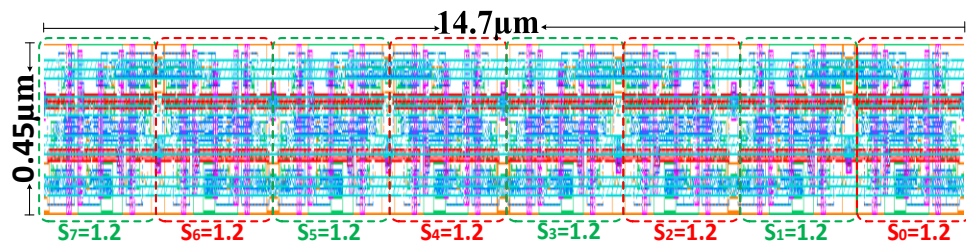


Figure 2. Comparison of the Optimal Design and the Traditional Design, Numerical Example 1



$S_{total} = 9.6$ , optimal design of numerical example 1



$S_{total} = 9.6$ , traditional design of numerical example 1

Figure 3. Layout of the Optimal and Traditional Design, Numerical Example 1



Figure 4. Video Output Example for  $s_{total} = 9.6$  from Table 3., Numerical Example 1 ( $V_{DD} = 0.75V$ ): (A) Original Video; (B) Video Stored by the Optimal Memory Design; (C) Video Stored by the Traditional Memory Design.

Figure. 3 shows the layout of the optimal design and the traditional design of this example, where both  $S_{total}$ s are 9.6 and all bitcells are 6T SRAM. In the traditional design, all the memory bitcell areas are equally sized as 1.2. By comparison, in the optimal design larger memory bitcells are selected to store MSBs ( $S_7, S_6 \dots S_0$ ) to improve the video output quality and the smallest bitcells (C61) are adopted for the LSBs to meet the silicon area constraint. The comparisons of the original video, video stored by the optimal memory design, and by the traditional design are shown in Figure. 4. It can be seen that the optimal memory design delivers much higher video quality as compared to the traditional design. Specifically, one can see from Table 3 that as  $S_{total}$  is 9.6, the MSE under the optimal and traditional designs are 75.34 and 777.38, thereby enabling a 90.31% improvement using the optimal design.

#### 2.4. Model 2: Optimal Design for Hybrid SRAM without Overhead

Recently, many alternative more-than-6T SRAM bitcells (e.g., 8T, 10T) have been developed to enhance the reliability as compared to traditional 6T, thereby enabling low-voltage operation. Sizing up transistors in 6T bitcells can also effectively reduce the failure rate [7]. However, both the more-than-6T bitcells and the sizing technique induce large silicon area overhead. To achieve a tradeoff between the cost (e.g., area, weight, or money cost) and the

video quality (e.g., the MSE), hybrid SRAM memory designs, such as 6T/8T [8], 8T/10T [9], and bitcells with different sizing techniques [7], have been developed by researchers. In those designs, integration bitcells with different design options typically does not bring additional silicon area cost.

### 2.4.1. The Mathematical Model

Suppose we have  $r_w$  options of SRAM bitcell  $w, w = 1, \dots, t$  (for example, type 1: 6T with  $r_1 = 4$  options, type 2: 8T with  $r_2 = 3$  options). Let  $r := \sum_{w=1}^t r_w$  be the total number of design options. Our goal is to make a decision on selecting one option for each bitcell to minimize the expected MSE of the whole memory chip. To this end, we define decision variable

$$x_{ijkl} = \begin{cases} 1, & \text{if option } l \text{ is chosen for the } ijk^{th} \text{ cell} \\ 0, & \text{otherwise} \end{cases}, \quad (7)$$

$$(i = 1, \dots, m; j = 1, \dots, n; k = 0, \dots, 7; l = 1, \dots, r)$$

where the first  $r_1$  options (i.e.,  $l = 1, \dots, r_1$ ) represent the options of the first type of SRAM, the next  $r_2$  options (i.e.,  $l = r_1 + 1, \dots, r_1 + r_2$ ) represents the second type,  $\dots$ , and the last  $r_t$  options (i.e.,  $l = r_1 + \dots + r_{(t-1)} + 1, \dots, r$ ) represent the options of the  $t^{th}$  type. Assuming that the failure rate of the  $l^{th}$  option of cell  $ijk$  is a known constant  $q_{ijkl}$ , the problem can be formulated as the following problem.

$$[M2] \quad \min_x \sum_{i=1}^m \sum_{j=1}^n \sum_{k=0}^7 \sum_{l=1}^r 4^k q_{ijkl} x_{ijkl} \quad (8)$$

$$\text{s.t. } \sum_{l=1}^r x_{ijkl} \geq 1, \quad i = 1, \dots, m; j = 1, \dots, n; k = 0, \dots, 7 \quad (9)$$

$$\sum_{i=1}^m \sum_{j=1}^n \sum_{k=0}^7 \sum_{l=1}^r s_{ijkl} x_{ijkl} \leq s_{total} \quad (10)$$

$$x_{ijkl} \in \{0,1\}, \quad i = 1, \dots, m; j = 1, \dots, n; k = 0, \dots, 7 \quad (11)$$

The objective function (8) is to minimize the expected MSE of the whole video.

Constraint (9) guarantees that for each cell one can choose exactly one design option (among the

total  $r$  options). It is worth mentioning that (9) is equivalent to  $\sum_{l=1}^r x_{ijkl} = 1$ , since this is a minimization problem. The total-area constraint, (10), assures that the total area of the design cannot exceed the limit  $s_{total}$ , where  $s_{ijkl}$  is a known parameter indicating the area cost of the  $ijk^{th}$  bitcell if it is selected to apply the  $l^{th}$  design option. We calculate the total area cost by directly summing up the area cost of each cell, since different SRAM bitcells typically can be laid out in a mirrored fashion and usually there is no area overhead for bitcell integration in hybrid SRAM design [8-9]. Finally, constraint (11) states that  $x_{ijkl}$ 's are binary variables. Different from [M1] which is an Non-Linear Programming (NLP) problem, [M2] is an Integer Linear Programming (ILP) problem.

#### 2.4.2. Numerical Study of Model 2

Using Model 2, we study the optimal hybrid SRAM design for 6T (type 1) and 8T (type 2) structures. The data used in this numerical study is shown in Table 4, where we have totally  $r=r_1+r_2=4+3=7$  options. We assume that all pixels are using the same memory design. Compared with the 6T options, one can find that the 8T SRAM requires a larger area but has a much lower failure rate.

It should be noted that the layout of the smallest 8T bitcell design (C81) has been optimized to minimize the failure rate, which enables approximately 10-time failure rate reduction in the 45-nm technology as compared to the conventional transistor sizing approach [14].

We solve [M2] of this problem for  $s_{total}=8.0, 8.2, \dots, 9.4$  using Gurobi solver (version 7.0.2), and the computation time of the optimal design for each  $s_{total}$  case is less than 0.2 second. The optimal values and solutions are shown in Table 5. The result of the proposed optimal design is also compared with the traditional design. In the traditional design, all bitcells

select the same option as reported in [13], i.e., the option with the largest area such that the total area does not exceed the given  $s_{total}$ . For example, if  $s_{total}=8.4$ , then all  $s_k$ 's will be C62, since  $1.02627 \cdot 8 = 8.210 < 8.4 < 1.05255 \cdot 8 = 8.418$  (C63). One can find that the optimal design reduces more than 90% of the expected MSE compared with the traditional design at  $s_{total}=8.2, \dots, 8.9$ . In addition, with a uniform (0.1 unit) increase to the  $s_{total}$ 's, the variance of improvement of the expected MSE in the traditional design is much larger than that in the optimal design. Specifically, there exists a sharp change in the expected MSE between  $s_{total}=8.7$  and 8.8 in the traditional design (see Figure. 5), which is due to the great difference between C81 and C64 in terms of their failure rates. When  $s_{total}=8.8$ , we are able to select C81 for all cells in the traditional design, whereas at  $s_{total}=8.7$  only C64. By contrast, due to the flexible selective options, the improvements on the expected MSE (under the same  $s_{total}$ 's) from the proposed optimal design is much more stable and hence easier for quality control.

Figure 6 shows the layout of the optimal and traditional design reported in Table 5 at  $s_{total}=8.7$ , where 6T SRAM and 8T SRAM bitcells are used to store the pixel data. Traditional

Table 4. Data of The Numerical Example 2 ( $V_{DD} = 0.5V$ )

Memory Type	Height ( $\mu m$ )	Width ( $\mu m$ )	Area ( $\mu m^2$ )	Area Ratio $s_k$	Failure rate $q_{kl}$
6T: C61	0.45	1.523	0.685	1	0.3436
6T: C62	0.45	1.563	0.703	1.026	0.3074
6T: C63	0.45	1.603	0.721	1.053	0.2771
6T: C64	0.45	1.643	0.739	1.079	0.2521
8T: C81	0.45	1.663	0.751	1.096	0.00082
8T: C82	0.45	1.700	0.765	1.117	0.00009
8T: C83	0.45	1.740	0.783	1.143	0.00002

design adopts the same bitcell (C64) to meet the silicon area constraint. In the optimal design, larger 8T SRAMs (C82 and C83) are utilized for MSBs; smaller 8T (C81)) and smallest 6T (C61) bitcells are used to store LSBs. The comparison of the original video, video stored by the optimal memory design and by the traditional design is shown in Figure. 7. One can clearly see the improvement of the optimal design (in which the MSE is 2.5) compared with the traditional design (in which the MSE is 5507.13). It is worth mentioning that in this numerical study the  $V_{DD}$  is set to be only 0.5V for near-threshold operation, but the video stored by the proposed optimal design still has a very high quality.

Table 5. Results and Comparisons of the Numerical Example 2 ( $V_{DD} = 0.5V$ )

$S_{total}$	Optimal design									Traditional design		Improve ment
	Obj. value	$s_7$	$s_6$	$s_5$	$s_4$	$s_3$	$s_2$	$s_1$	$s_0$	Obj. value	$s_k$	
8.0	7505.94	C61	C61	C61	C61	C61	C61	C61	C61	7505.94	C61	0.00%
8.1	1889.83	C81	C61	C61	C61	C61	C61	C61	C61	7505.94	C61	74.82%
8.2	485.81	C81	C81	C61	C61	C61	C61	C61	C61	7506.94	C61	93.53%
8.3	134.80	C81	C81	C81	C61	C61	C61	C61	C61	6715.15	C62	97.99%
8.4	47.05	C81	C81	C81	C81	C61	C61	C61	C61	6715.15	C62	99.30%
8.5	25.11	C81	C81	C81	C81	C81	C61	C61	C61	6053.25	C63	99.59%
8.6	7.67	C82	C81	C81	C81	C81	C81	C61	C61	6053.25	C63	99.87%
<b>8.7</b>	<b>2.50</b>	<b>C83</b>	<b>C83</b>	<b>C82</b>	<b>C81</b>	<b>C81</b>	<b>C81</b>	<b>C61</b>	<b>C61</b>	<b>5507.13</b>	<b>C64</b>	<b>99.95%</b>
8.8	1.12	C83	C83	C82	C81	C81	C81	C81	C61	17.91	C81	93.73%

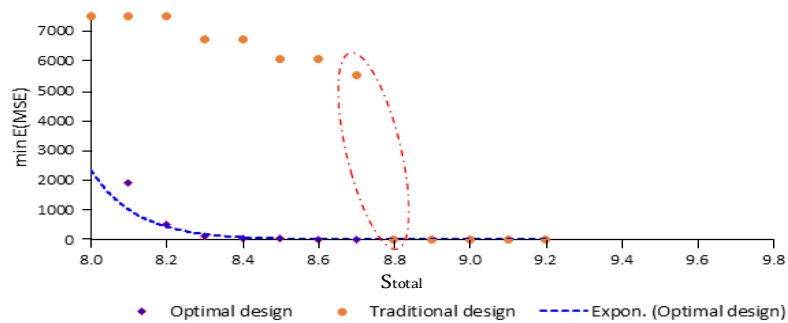


Figure 5. Comparison of the Optimal Design and the Traditional Design, Numerical Example 2



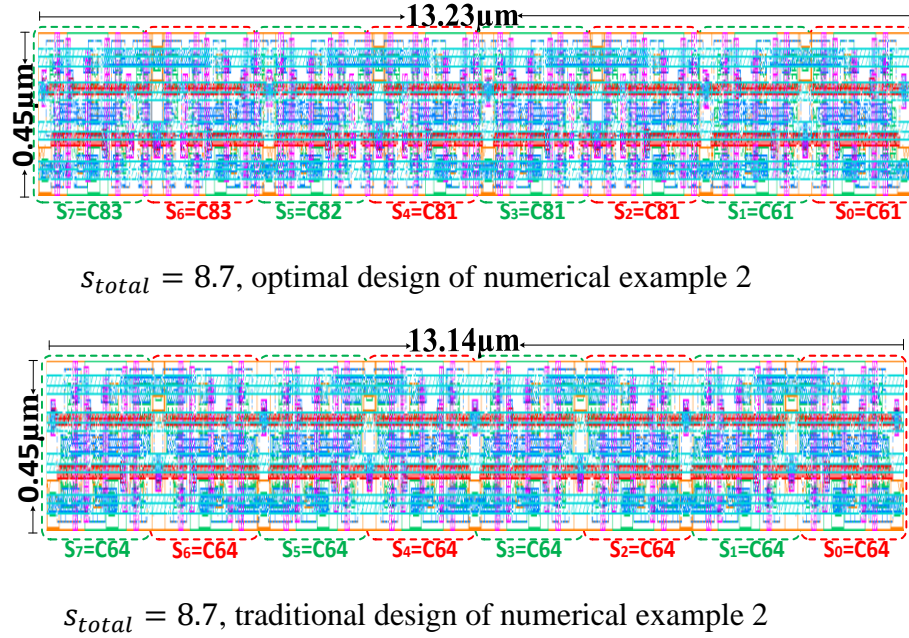


Figure 6. Layout of the Optimal and Traditional Design, Numerical Example 2



Figure 7. Video Output Quality Example for  $s_{total} = 8.7$  from Table 5, Numerical Example 2 ( $V_{DD} = 0.5V$ ): (A) Original Video; (B) Video Stored by the Optimal Design; (C) Video Stored by the Traditional Design.

### 2.5. Model 3: Optimal Design for Hybrid Memory with Various Technologies

Very recently, industry and researchers have made lots of efforts to find feasible low-cost and energy-efficient alternatives beyond SRAM to meet the huge storage requirements, for example, embedded DRAM and emerging non-volatile memory (NVM) technologies. Integrating different memory technologies for hybrid memory design, such as hybrid SRAM+DRAM memory, enables emerging opportunities in developing more advanced



memories. However, it also becomes more challenging and complex to find an optimal design. This is because, first, the bitcell structures using different memory and different technologies vary significantly, thereby usually causing integration silicon area overhead in such hybrid memory designs. By contrast, the hybrid memory design investigated in Section 2.5 is assumed to have the same memory technology and has no area overhead. Second, different technologies (e.g., SRAM, DRAM, NVM) have their own specific energy-failure-cost characteristics, and each technology has its design options (e.g., bitcells and sizing for SRAM).

Without the loss of generality, we consider three types of memory structures in this section: 3T DRAM, 6T SRAM and 8T SRAM. The area integration between the DRAM and the SRAMs will be discussed.

### **2.5.1. Integration Cost of SRAM and DRAM**

Compared to SRAM, DRAM allows more compact storage but requires frequent refresh operations to avoid memory failures, which takes more energy consumption. Traditional DRAM design schemes, including commercial memories, are implemented based on the worst-case refresh-cycle, which is determined by the leakiest cell in the DRAM array. However, when the voltage supply is fixed, the additional needs of the refresh operations will lead to a significant energy consumption due to the expensive and periodic activation of individual rows during the refresh process [15]. Researchers have developed different DRAM bitcells (e.g., 1T1C, 2T). Among them, 3T demonstrates enhanced efficiency, since it does not need boosted power supply to write the data into the store node (SN), significantly reducing the power consumption during writing operations [10]. The refresh period (i.e., data retention time (DRT)) enabled energy-quality tradeoff for the 3T DRAM bitcells is shown in Figure 8. As the refresh period of the DRAM increases, the refresh power consumption is reduced, but the memory failure rate grows

due to the failed data retention process. Such energy-quality tradeoff property of the DRAM provides design opportunities for using mathematical models to explore the “balance” (i.e., the optimal design). For hybrid memory using different technologies such as SRAM and DRAM, the hybrid bitcell integration usually causes additional silicon area overhead due to the significant difference among the bitcell structures. Figure 9 shows an example of the hybrid 3T DRAM, 6T (C61) SRAM and 8T (C81) SRAM, where the 6T (C61) and 8T (C81) SRAMs are from Table 4. Since the transmission gate consists of both NMOS and PMOS transistors, the 3T DRAM requires an additional wordline signal (WWLp) compared to the SRAM bitcells. To transmit this additional wordline through the hybrid memory bitcells, the layout of the 6T and 8T SRAM bitcells needs to be increased to meet the design rules, thereby causing the silicon area overhead.

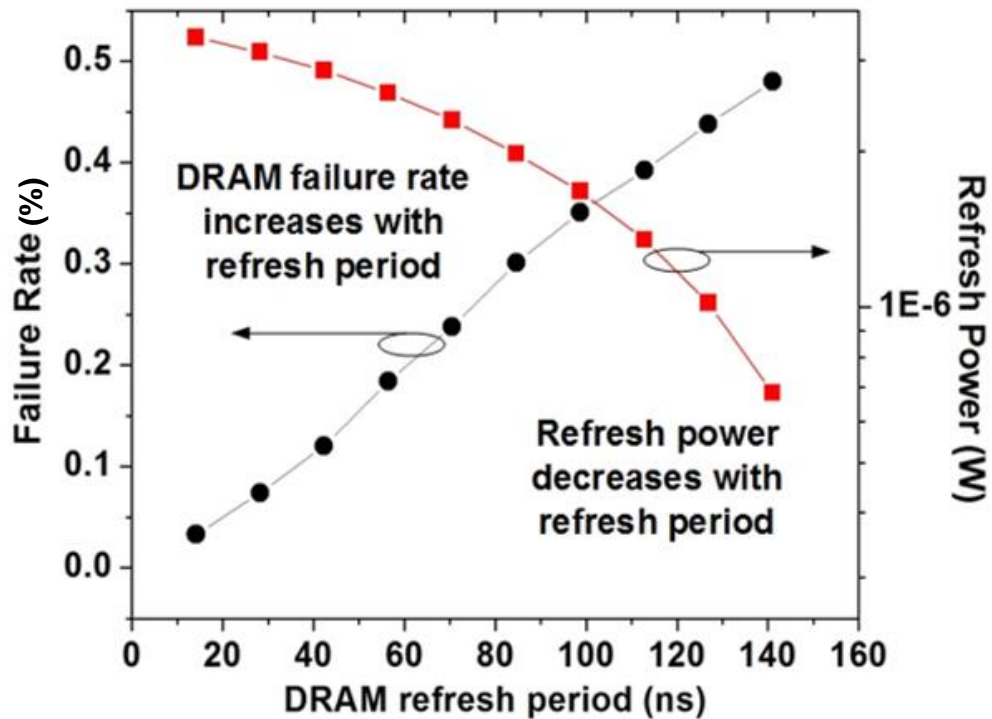


Figure 8. Refresh Period Enabled Energy-Quality Adaptation for DRAM

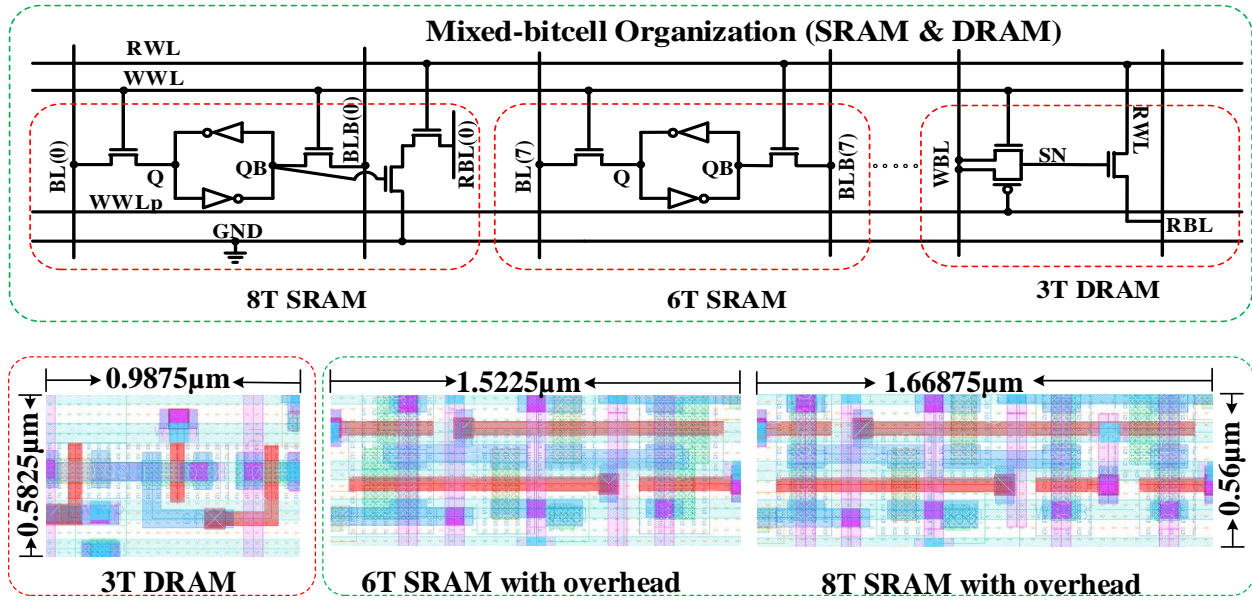


Figure 9. Schematic and Layout of 3T DRAM, 6T and 8T SRAM Bitcells Compatible with DRAM.

In addition, the height of 3T DRAM bitcell layout is usually greater than that of SRAM bitcells, which also leads to additional area costs. In Figure 9, due to the hybrid structure with the 3T DRAM, one can see that the height of the new C61 and the new C81 SRAMs are both increased from the original  $0.45\mu\text{m}$  (based on the 45-nm predictive technology, as shown in Table 4) to  $0.56\mu\text{m}$ .

### 2.5.2. The Mathematical Model

To tackle the issues of integration silicon area overhead and deal with various design options using different memory technologies, a new analytical model is needed. Suppose we have  $r_6$ ,  $r_8$  and  $r_3$  options for selection of 6T SRAM, 8T SRAM and 3T DRAM, respectively. Let  $r := r_6 + r_8 + r_3$  indicate the total number of design options for each bitcell. We assume that all pixels have the same design for notation convenience.

Our goal is again to make a decision for each bitcell to minimize the expected MSE of the whole memory chip. Similar to Model 2, the main decision variable,  $x_{kl}$ , is defined by (7),

where options 1 to  $r_6$ ,  $r_6+1$  to  $r_6+r_8$ , and  $r_6+r_8+1$  to  $r$  represent the specific options of 6T SRAM, 8T SRAM and 3T DRAM, respectively. However, the total-area constraint (corresponding to (10) in [M2]) needs to be modified, since the dimensions of 6T and 8T SRAMs will be changed if a 3T DRAM is included in a pixel. To this end, we define a new binary decision variable  $\delta$  to indicate whether a pixel includes any 3T DRAM structure (where  $\delta=1$  means “yes”) by the following constraints:

$$\begin{cases} \frac{1}{8r_3} \sum_{k=0}^7 \sum_{l=r_6+r_8+1}^r x_{kl} \leq \delta \leq \sum_{k=0}^7 \sum_{l=r_6+r_8+1}^r x_{kl}, \\ \delta \in \{0,1\} \end{cases} \quad (12)$$

Clearly, if any 3T DRAMs exist in a memory array, the left part of the first constraint in (12) assures  $\delta = 1$ ; otherwise, the right part of the first constraint in (12) guarantees that  $\delta = 0$ . Following (12), the new total-area constraint can be formulated as follows:

$$\sum_{k=0}^7 \left[ \sum_{l=1}^{r_6+r_8} (s_{kl}(1-\delta) + s_{kl}^{(3T)}\delta) x_{kl} + \sum_{l=r_6+r_8+1}^r s_{kl} x_{kl} \right] \leq s_{total} \quad (13)$$

where  $s_{kl}$  (for  $l = r_6 + r_8 + 1, \dots, r$ ) represents the area cost of the  $k^{th}$  bitcell applying option  $l$  as a 3T DRAM;  $s_{kl}$  (for  $l = 1, \dots, r_6 + r_8$ ) and  $s_{ijkl}^{(3T)}$  ( $l = 1, \dots, r_6 + r_8$ ) represent the area cost of the  $k^{th}$  bitcell applying option  $l$  as a 6T or 8T SRAM, with and without a 3T DRAM included in the pixel. To linearize the item  $\delta x_{kl}$  in (13), we create the following constraints:

$$\begin{cases} y_{kl} \leq \delta \\ y_{kl} \leq x_{kl} \\ y_{kl} \geq \delta + x_{kl} - 1, \forall k, \forall l = r_6 + r_8 + 1, \dots, r, \\ y_{kl} \in \{0,1\} \end{cases}$$

(so that we have  $y_{kl} = \delta x_{kl}$ ). After simplification, the whole problem can be formulated as the following ILP.

$$\begin{aligned}
\text{[M3]} \quad & \min_{x, \delta, y} \sum_{k=0}^7 \sum_{l=1}^r 4^k q_{kl} x_{kl} \\
\text{s.t.} \quad & \sum_{l=1}^r x_{kl} \geq 1, \quad k = 0, \dots, 7 \\
& \sum_{k=0}^7 \left[ \sum_{l=1}^{r_6+r_8} (s_{kl}(1-\delta) + s_{kl}^{(3T)} \delta) x_{kl} \right. \\
& \quad \left. + \sum_{l=r_6+r_8+1}^r s_{kl} x_{kl} \right] \leq S_{total} \\
& \frac{1}{8r_3} \sum_{k=0}^7 \sum_{l=r_6+r_8+1}^r x_{kl} \leq \delta \leq \\
& \quad \sum_{k=0}^7 \sum_{l=r_6+r_8+1}^r x_{kl} \\
& y_{kl} \leq \delta, \quad \forall k, \forall l = 1, \dots, r \\
& \delta + x_{kl} - 1 \leq y_{kl} \leq x_{kl}, \quad \forall k, l \\
& x_{kl}, \delta, y_{kl} \in \{0, 1\}, \quad \forall k, l
\end{aligned}$$

Table 6. 6T and 8T SRAM Data of Numerical Example 3 ( $V_{DD} = 0.5\text{v}$ ), with Area-Overhead Caused by 3T DRAM

Memory Type (with area-overhead)	Height ( $\mu\text{m}$ )	Width ( $\mu\text{m}$ )	Area ( $\mu\text{m}^2$ )	Area Ratio $s_k$	Failure rate $q_{kl}$
6T: C61	0.56	1.5225	0.8526	$s_{k1}^{(3T)} = 1.24$	0.3436
6T: C62	0.56	1.5625	0.875	$s_{k2}^{(3T)} = 1.28$	0.3074
6T: C63	0.56	1.6025	0.8974	$s_{k3}^{(3T)} = 1.31$	0.2771
6T: C64	0.56	1.6425	0.9198	$s_{k4}^{(3T)} = 1.34$	0.2521
8T: C81	0.56	1.6688	0.9345	$s_{k5}^{(3T)} = 1.36$	0.00082
8T: C82	0.56	1.7000	0.9520	$s_{k6}^{(3T)} = 1.39$	0.00009
8T: C83	0.56	1.7400	0.9744	$s_{k7}^{(3T)} = 1.42$	0.00002

### 2.5.3. Numerical Study of Model 3

We have  $r = r_6 + r_8 + r_3 = 4 + 3 + 2 = 9$  design options in the data used for this numerical study. If no 3T DRAM is selected in a pixel, we continue using the data in Table 4 for 6T and 8T SRAMs. Otherwise, the data of these SRAMs is shown in Table 6. The data of 3T DRAM is shown in Table 7, where two 3T options with the same area cost but different DRTs are included. We solve the problem for  $s_{total} = 7.0, 7.2, \dots, 8.4$  using Gurobi solver (version 7.0.2), and the computation time of the optimal design for each  $s_{total}$  case is less than 0.2 second. The optimal values and solutions are shown in Table 8. When the allowed  $s_{total}$  is very small, most bitcells apply C31 as it has the smallest area cost among all nine options. One can see that with the increase of  $s_{total}$ , the trend of update in the optimal design is from 3T to 6T, and then to 8T. As  $s_{total} \geq 8.2$ , the results of the optimal design of this example are the same as those of numerical example 2 (see Table 4). This is because no options select 3T (which has the

Table 7. 3T DRAM Data for Numerical Example 3 ( $V_{DD} = 0.5V$ )

Memory Type	Height ( $\mu m$ )	Width ( $\mu m$ )	Area ( $\mu m^2$ )	Area Ratio $s_k$	DRTs	Failure rate $q_{kl}$
3T: C31	0.583	0.987	0.5752	$s_{k8} = 0.84$	$1.13 \times 10^{-6}$	0.392
3T: C32	0.583	0.987	0.5752	$s_{k9} = 0.84$	$1.41 \times 10^{-6}$	0.480

Table 8. Results and Comparisons of Numerical Example 3 ( $V_{DD} = 0.5V$ )

$S_{total}$	Optimal design									Traditional design		Improvement
	Obj. value	$s_7$	$s_6$	$s_5$	$s_4$	$s_3$	$s_2$	$s_1$	$s_0$	Obj. value	$s_k$	
7.0	8563.24	C31	C31	C31	C31	C31	C31	C31	C31	8563.24	C31	0.00%
7.2	6680.72	C63	C31	C31	C31	C31	C31	C31	C31	8563.24	C31	21.98%
7.4	2141.04	C83	C31	C31	C31	C31	C31	C31	C31	8563.24	C31	75.00%
7.6	2141.04	C83	C31	C31	C31	C31	C31	C31	C31	8563.24	C31	75.00%
7.8	551.87	C81	C81	C31	C31	C31	C31	C31	C31	8563.24	C31	93.56%
<b>8.0</b>	<b>535.49</b>	<b>C83</b>	<b>C83</b>	<b>C31</b>	<b>C31</b>	<b>C31</b>	<b>C31</b>	<b>C31</b>	<b>C31</b>	<b>7505.94</b>	<b>C61</b>	<b>92.87%</b>
8.2	485.81	C81	C81	C61	C61	C61	C61	C61	C61	7505.94	C61	93.53%
8.4	47.05	C81	C81	C81	C81	C61	C61	C61	C61	6715.15	C62	99.30%

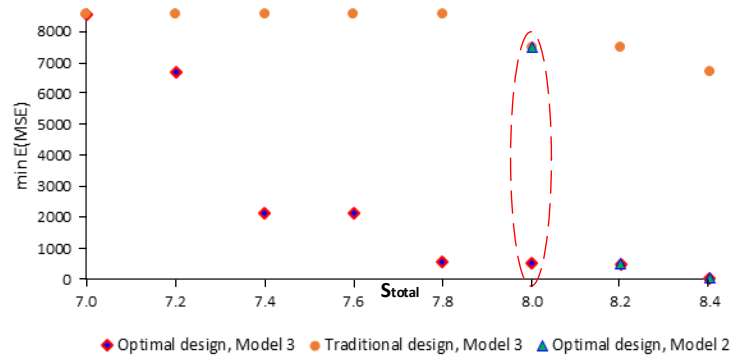


Figure 10. Comparison of the Optimal and Traditional Design of Numerical Example 3, and Part of the Optimal Design of Numerical Example 2 (For Both  $V_{DD} = 0.5V$ )

highest failure rate), as the allowed total area is big enough to include cells composed of only 6T and 8T SRAM.

In addition to the comparison between the optimal and traditional design of this numerical example, we also compares the results with part of those from the optimal solution of numerical example 2 in Figure 10.

A notable point is at  $s_{total} = 8.0$  where the optimal solution of this numerical example provides a solution with much higher video quality than that of numerical example 2 at the same  $s_{total}$ . The main reason of such a great improvement is because of the options of 3T DRAM which can be added to low-level bitcells to save space for adding reliable 8T SRAMs to high-level bitcells of the pixel.

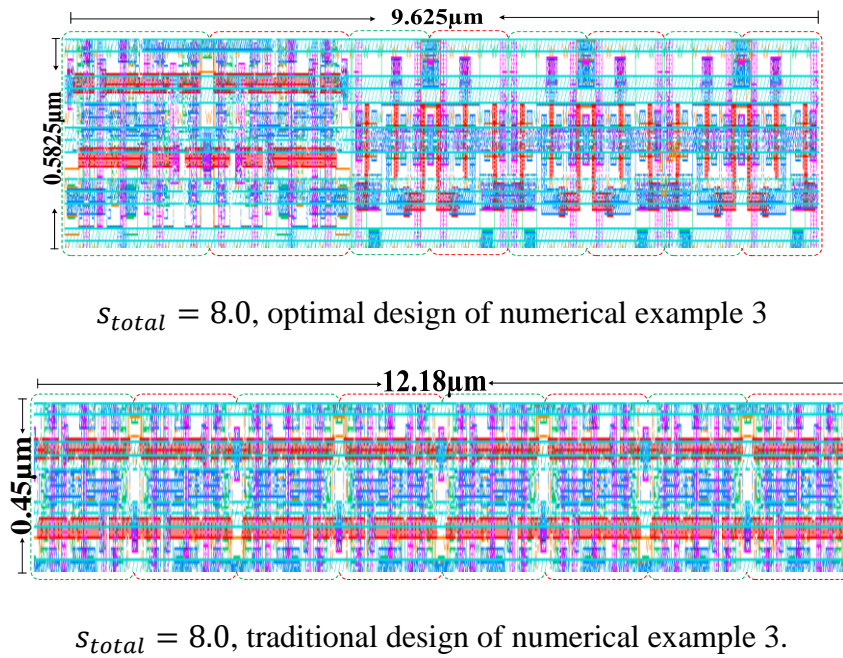


Figure 11. Layout of the Optimal and Traditional Design,  $s_{total} = 8.0$ , Numerical Example 3



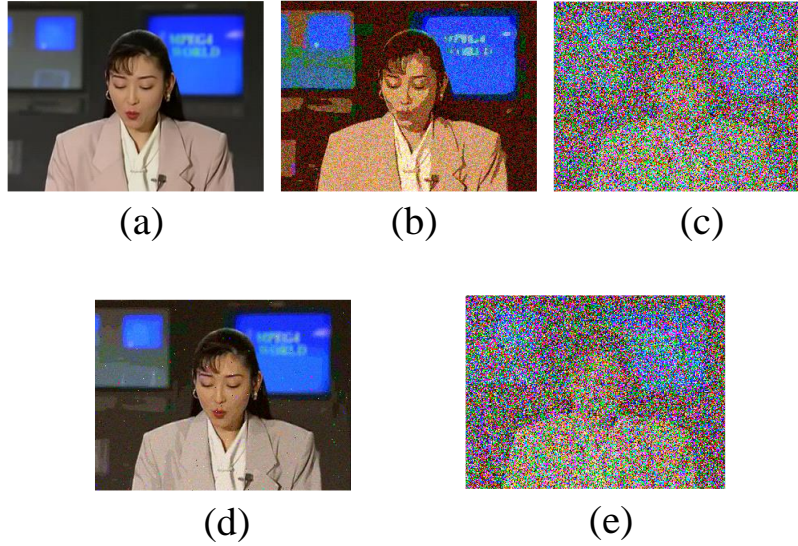


Figure 12. Video Output Quality Example For  $s_{total} = 8.0$  and  $8.4$  From Table 8, Numerical Example 3 ( $V_{DD} = 0.5V$ ): (A) Original Video; (B) Video Stored by the Optimal Design,  $s_{total} = 8.0$ ; (C) Video Stored by the Traditional Design,  $s_{total} = 8.0$ ; (D) Video Stored By the Optimal Design,  $s_{total} = 8.4$ ; (E) Video Stored by the Traditional Design,  $s_{total} = 8.4$

The layout and video output quality comparison at  $s_{total} = 8.0$  are shown in Figure 11. and Figure 12 ((a)—(c)). Although the video output quality of the optimal design is not good enough Figure 12 (b), it is much better than that of the traditional design under the same  $s_{total}$  (Figure 12 (c)). In addition, note that 8.0 is an extremely small area cost. If we increase it a little to 8.4, then the quality of the proposed optimal design will be significantly improved while that of the traditional design does not change too much (see Figure 12 (d) and (e)).

## 2.6. Discussion

### 2.6.1. Relationship of the Three Developed Models

In this chapter, three mathematical models have been developed for optimizing video memory design using nonlinear programs and integer linear programs.

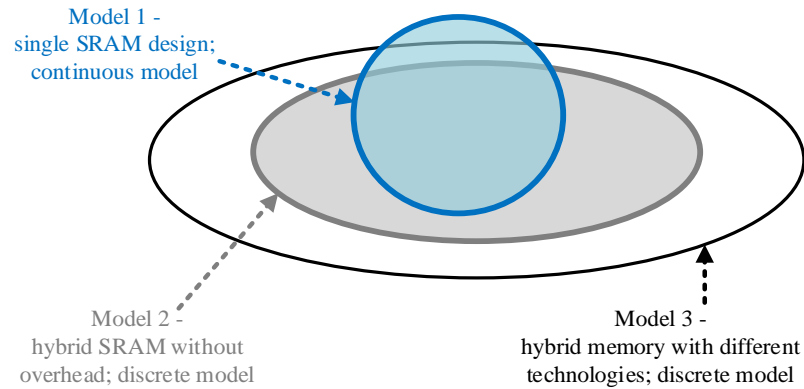


Figure 13. Relationship of the Three Models, Which Covers Different Mathematical Approach for Different Memory Technology.

Specifically, the developed three models include (i) Model 1: single SRAM design; (ii) Model 2: hybrid SRAM without bitcell integration cost; and (iii) Model 3: hybrid memory using different technologies such as SRAM and DRAM. The relationship of the three models are illustrated in Figure 13. Clearly, Model 3 is a generalization of Model 2, since various memory technologies and area overhead are considered. Although Model 1 is only for single SRAM design and Model 2 is for hybrid SRAM design, one cannot simply think that Model 2 is a generalization of Model 1. This is because Model 2 is a discrete model while Model 1 is continuous. In Figure 13 we use the part that belongs to Model 1 but not Model 2 or Model 3 to indicate this point.

### 2.6.2. Comparison with Prior Work

To the best of our knowledge, none of the literature comprehensively considers the general optimal memory design under a cost/area constraint for hybrid SRAMs and for designs with various memory technologies. For example, [7] presents a heterogeneous sizing scheme, but it is only for 6T SRAM under several (discrete) options, which is a special case of our Model 2. We further provide a continuous model for 6T SRAM optimization (i.e., Model 1). [8] develops

a hybrid 6T/8T SRAM, but it simply suggests storing high-order bits in 8T bitcells and low-order bits in 6T bitcells, without giving any detailed mathematical models (except for the formulation of MSE in the Appendix) or algorithms to obtain an optimal design when there are various choices of 6T/8T. Other existing studies on memory design, such as [9] and [16] are only for SRAMs and do not include any optimization models or algorithms.

### 3. MEMORY OPTIMIZATION FOR ENERGY-EFFICIENT DIFFERENTIALLY PRIVATE DEEP LEARNING<sup>2</sup>

#### 3.1. Introduction

With the advent of Internet of Things (IoT) technologies and availability of a large amount of data, deep learning has been applied in a variety of artificial intelligence (AI) applications. However, sharing personal data using IoT edge devices carries inherent risks to individual privacy. Meanwhile, the energy and memory resources needed during the inference process become a constraint to the resource-limited IoT edge devices. This paper brings memory hardware optimization to meet the tight power budget in IoT edge devices by considering the privacy, accuracy, and power efficiency tradeoff in differentially private deep learning systems. Based on a detailed analysis of these characteristics, an Integer Linear Programming (ILP) model is developed to minimize mean square error (MSE), thereby enabling optimal dataset memory design. The simulation results in 45-nm CMOS technology show that the proposed technique can enable near-threshold energy-efficient memory operation for different privacy requirements, with less than 1% degradation in classification accuracy.

The main contributions of this work are as follows: the power efficiency, accuracy, and privacy characteristics of differentially private deep learning systems were analyzed. An input data memory design with upsized and 8T+6T hybrid bit-cells for optimization was presented. The design shows less than 1% degradation in classification accuracy for different privacy levels, with reduced power consumption.

---

<sup>2</sup> **Hritom Das** was in charge of all memory (SRAM) system design and simulation in Cadence, failure rate calculation, layout design, and calculation of power consumption for different supply voltages. Drs. Na Gong, Yiwen Xu, and Jonathon Edstrom provided the simulation and modeling for deep learning.

## **3.2. Learning with Differential Privacy**

### **3.2.1. Privacy Preservation in Deep Learning**

Privacy research has drawn attention in both industry and research communities. Large industry leaders, including: Apple, Facebook, and Google, have concluded that these types of threats can be accomplished by invasive analysts even when data has been anonymized [17-19]. For example, in 2006 AOL released a list of 20 million web search queries which was found to have leaked the identity of a woman [20]. Similarly, Netflix introduced an open competition in 2006 that released a dataset that also leaked private data [21-22]. One other area with potential privacy issues is biomedical research. For example, in genome wide association studies, the identity and any diseases a person has could be revealed based on results included in research papers [23]. Due to privacy risks such as these, a conscious effort to reduce data leaks has become of great interest, especially for companies using machine learning algorithms on collected big data.

The privacy of deep learning models, such as neural networks, have recently come into question due to weaknesses and attack models that have been previously exploited [24]. Due to high requirements of computation and storage resources, today's deep learning systems are typically built upon large, centralized data repositories. Many cloud providers also give access to computing platforms and learning frameworks for model training, such as Amazon Sagemaker and Google Cloud ML Engine. Based on this centralized-training paradigm, data owners need to upload their private data to the cloud provider and they do not have control over how their private data is being used. For instance, if a deep learning model was trained on the records of patients with a certain disease, learning that an individual's record was part of the training data directly affects their privacy, and it opens a door to potential misuse (e.g., exploitation for the

purpose of recruitment, insurance pricing, or granting loans) due to the following three potential privacy threats: (i) it is very easy for a malicious provider to steal the data if the provider has full access to the data [28]; (ii) even without full access to the data, the malicious provider can extract sensitive data from the trained models [29]; and (iii) A malicious remote user can also retrieve information of the training data by carefully querying the trained models [30].

### 3.2.2. Differentially Private Deep Learning and State of the Art

To preserve data privacy, differential privacy [31] is becoming the gold standard to offer both utility to the applications and rigorous privacy guarantees. The formal definition is as follows: a randomized mechanism  $M$  is considered to be  $(\epsilon, \delta)$ -differentially private if, for two adjacent inputs  $d$  and  $d'$ , it holds that  $\Pr[M(d) \in S] \leq e^{\epsilon} \cdot \Pr[M(d') \in S] + \delta$ , where  $S$  is any subset of the outputs. The privacy cost parameter  $\epsilon$  is used to control the tradeoff between the privacy and the accuracy where smaller values of  $\epsilon$  provide more privacy. The guarantee of differential privacy is: if an individual's data is used in a differentially private calculation, the probability of any result of the calculation changes by at most a factor of  $e^{\epsilon}$  in comparison to if that individual's data is not used in the calculation [36]. The parameter  $\delta$  is the probability of failure where the given differentially private mechanism may violate an individual's privacy. This  $\delta$  value explains the possibility of "bad events" that may result in a large loss in privacy. Specifically, when training an  $(\epsilon, \delta)$ - differentially private neural network, the probability of violating the privacy,  $\delta$ , is calculated after each step for a given privacy cost,  $\epsilon$ .

Recent works have adopted the use of  $(\epsilon, \delta)$ -differential privacy in order to protect the data of individuals. In [37], the authors presented a technique involving an ensemble of teachers that could train on subsets of a sensitive data. After training, the teachers would further train a student model based on public data that was labeled using the ensemble. The student model is

trained based on the noisy voting of the various teachers that were trained using the model so that a stronger privacy guarantee can be enabled by the system. In [38], a method creating generative adversarial networks (GANs) that include differentially private mechanisms to provide privacy guarantees was presented. This technique for training a differentially private GAN only allows the analyst to inspect a model that already guarantees some level of differential privacy. Both the teacher ensemble and differentially private GAN training techniques employ the use of a privacy accountant (i.e. the moments accountant), described in [39], in order to compute a tighter bound on the differential privacy.

In order to ensure differential privacy, perturbation can be introduced at various parts of the workflow, including: input, output, and objective perturbation [40]. Also, different types of noise can be added to the training and test datasets. The moments accountant shows that for the Gaussian (i.e.  $\sim N(0, \sigma^2)$ ) noise mechanism, if the value of standard deviation for this noise mechanism is chosen to be:

$$\sigma = \frac{1}{\epsilon} (2 \log \frac{1.25}{\delta})^{1/2} \quad (14)$$

then the noise mechanism will satisfy  $(\epsilon, \delta)$ -differentially privacy for a given sensitivity,  $S_f$ . Using this moments accountant technique to compute a tight bound on the privacy allows for each step in the training algorithm to result in  $(\epsilon, \delta)$ -differential privacy with respect to the lot.

The system we propose uses the moments accountant to train a differentially private ConvNet model on the server (cloud) where sensitive data is used for training. By enabling the moments accountant for training we can guarantee privacy, but at the cost of some accuracy loss.

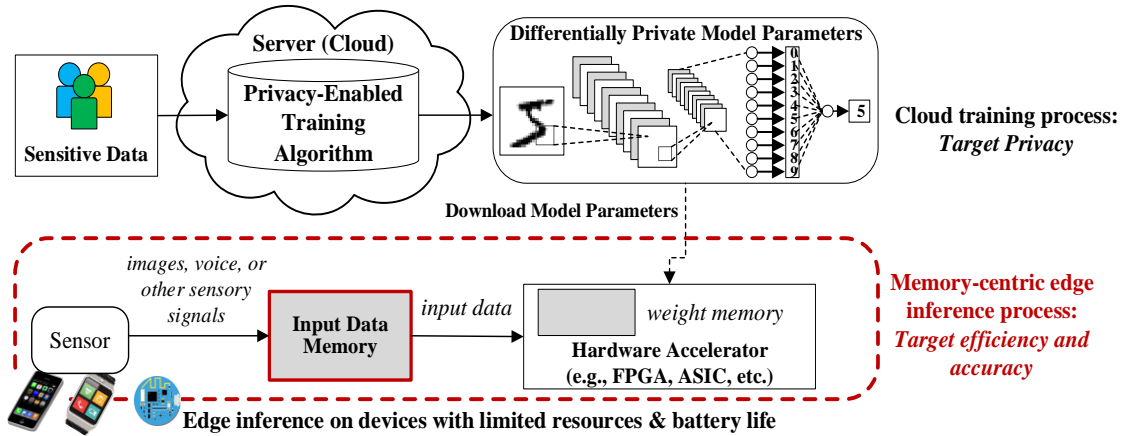


Figure 14. Input Data Memory Optimization for Deep Learning System with Energy-Efficiency, Privacy, and Inference Accuracy

This trained, differentially private model will then be downloaded to the edge computing devices for inference tasks. A diagram of the proposed system design can be seen in Figure 14. Since inference is taken care of on the local devices, the privacy of the testing data being presented to the devices is not a big concern.

The energy and resources needed during the inference process has become another constraint to the resource-limited IoT devices. Deep learning models can take up a large portion of an embedded device’s memory space and inference tasks. In particular, data movement on these devices can consume the majority of the total power [41]. Software compression techniques for reducing the size of each weight in deep learning models have been introduced, such as the Tensor Flow Lite API [42], which allows for 4× reduction in total model size. For hardware improvements, one of the most important issues that has been focused on is the intensive memory access of the embedded IoT devices. Very recently, [43] presented a memory-based noise addition technique for differentially private deep learning systems, illustrating the significance of the embedded memory to edge inference tasks. However, this technique adopted



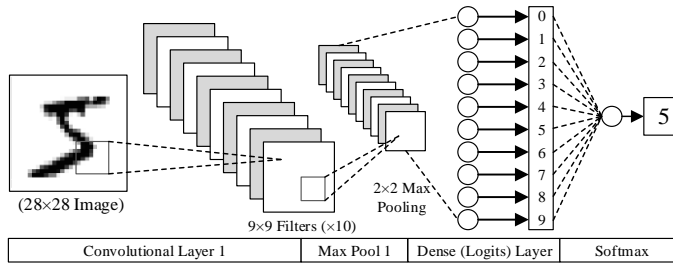


Figure 15. Differentially Private Convolutional Neural Network Used in Our Analysis.

the traditional memory design, which misses out on many optimization opportunities to trade off among privacy, accuracy, and efficiency.

This chapter aims to optimize memory design to better support differentially private deep learning algorithms in local devices. To enhance the power efficiency of memories, memory failures are usually introduced due to process variations during the device manufacturing process. We first analyze the impact of memory failures on accuracy and privacy and then conclude the guidelines to optimize the memory for privacy, efficiency, and accuracy in AI applications with different requirements.

### 3.3. Impact of Memory Failures in Differentially Private Deep Learning Systems

In our analysis, we define a convolutional neural network model using the TensorFlow framework [44] in order to gain insight on how different types and levels of noise may influence the privacy-accuracy tradeoff. The model involves using an objective perturbation through additive Gaussian noise, and uses the moment’s accountant [39] to compute the privacy cost after each step in the training process. The ConvNet model we tested was based on the architecture described in [43] with a single convolutional layer, as can be seen in Figure 15. The widely used MNIST dataset [45] was used for our initial simulations. MNIST consists of 60,000 training samples and 10,000 test samples, where each sample is a handwritten digit ranging from “0” to “9”; each sample is an image that contains 784 features representing 28×28 pixels.

### 3.3.1. Impact of Image Quality on Classification Accuracy

In order to investigate the relationship between the quality of the test dataset and its impact on the test classification accuracy, we inject bit level errors at varying memory failure rates (probabilities) to each image in the test dataset. Since the MNIST dataset consists of images, the well-known peak signal-to-noise ratio (PSNR) metric is used to evaluate quality, which is defined in [46] as

$$\text{PSNR} = 10 \log_{10} \left( \frac{255^2}{\text{MSE}} \right) \quad (15)$$

where MSE is the mean square error between the original images (Org) and the degraded images (Deg).

Accordingly, by evaluating the PSNR values for a wide range of error injected test datasets using MNIST and comparing the test classification accuracy, we identify that the higher the image quality in the test dataset, the higher the output accuracy of the system will be overall. This relationship between PSNR and test classification accuracy is illustrated in Figure 16. Based on this monotonically increasing behavior, if the PSNR value of the dataset is improved, the accuracy will be enhanced. Accordingly, during the memory design process, if the memory hardware can enable the optimal quality of the dataset, the accuracy will be improved accordingly. As shown in Figure 16, as the PSNR values of the MNIST dataset are increased from 5dB to 15dB, the accuracy is increased from 10% to 90% while meeting the privacy guarantee for the differentially private deep learning systems. It should be noted that PSNR is used in our analysis to evaluate the image quality of MNIST dataset, but considering different types of IoT data, MSE will be a general quality evaluation metric, which will be discussed in Section 3.3.4.

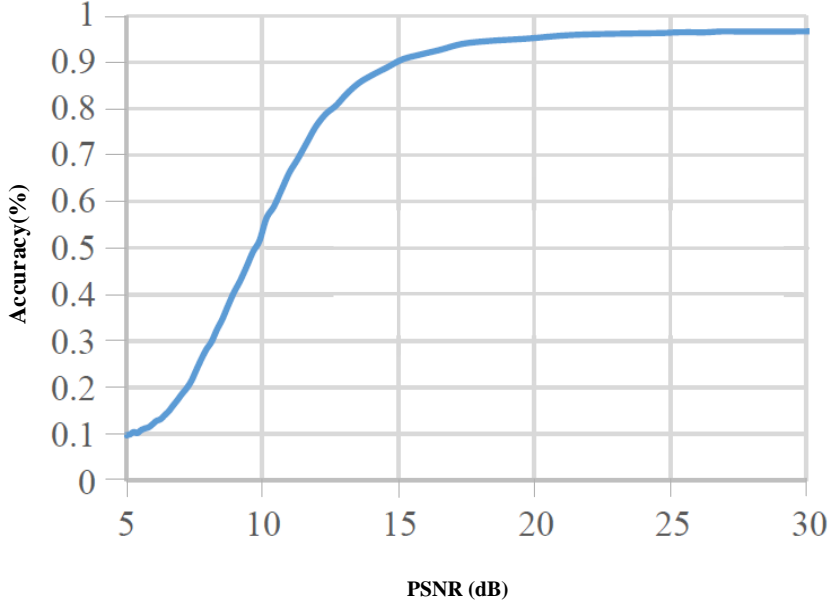


Figure 16. Influence of Dataset Quality on Test Accuracy (Using MNIST Dataset).

### 3.3.2. Protecting Most Significant Bits (MSBs) of Data

The amount of Gaussian noise that is used during training influences how accurate the inference of the finalized model performs. Therefore, different models with varying amounts of noise (i.e. sigma values) and epsilon values with a set delta value of  $10^{-5}$  have been studied. For sigma, we tested 4 different noise levels,  $\sigma \in \mathbf{Z} : 1 \leq \sigma \leq 4$ , and for each sigma value we tested 6 separate epsilon values,  $\epsilon \in \mathbf{Z} : 5 \leq \epsilon \leq 10$ .

The relevant results for the  $(\sigma, \epsilon)$  pairs we tested are shown in Figure 18. Our study shows that the best  $(\sigma, \epsilon)$  pairs (i.e. the values of sigma and epsilon that provide the best test classification accuracy) for the MNIST dataset are:  $\sigma = 1, \epsilon = 9$  and  $\sigma = 2, \epsilon = 8$  as memory failure rates of the dataset are increased. When training using these values for the parameters, the probability of violating the privacy is recalculated after each step in the training process until the end delta value  $\delta = 10^{-5}$  to stay within a modest privacy budget [39].

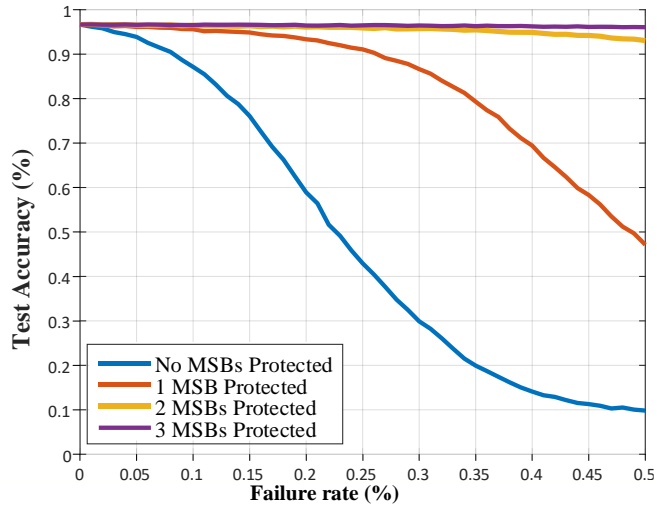


Figure 17. Impact of Memory Failure Rate on the Accuracy of the Learning System.

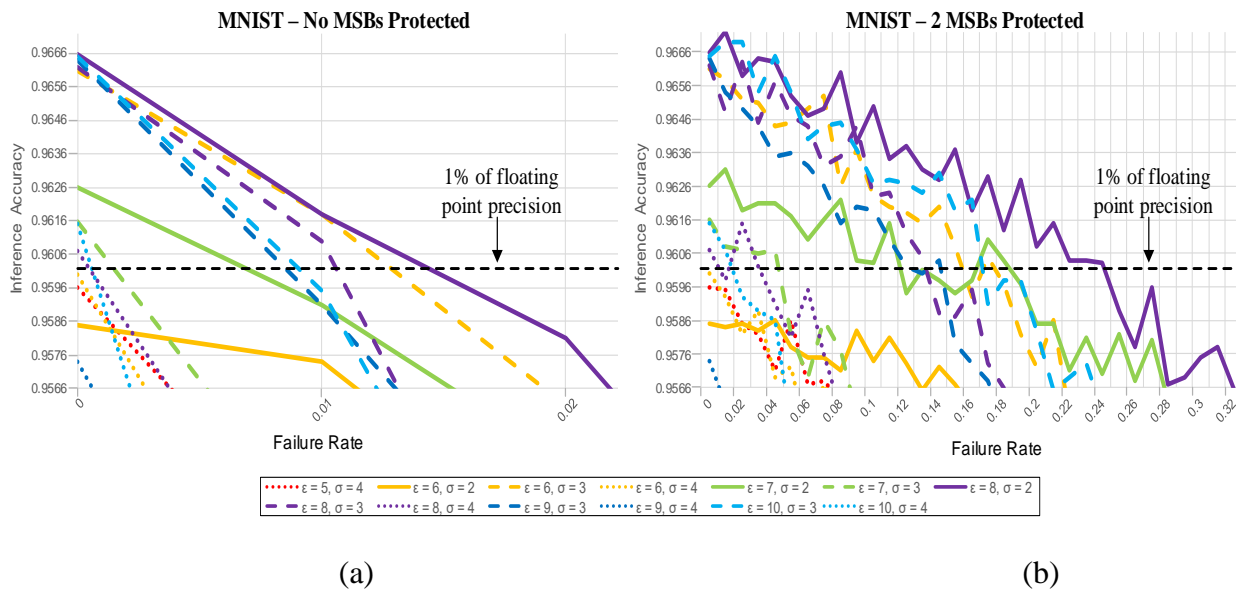


Figure 18. Impact of Memory Failure Rate on Privacy/Accuracy Tradeoff: (A) Without MSB Protected And (B) With 2 MSBs Protected.

One effective technique for increasing the PSNR of the test dataset when errors are present is to protect the most significant bits (MSBs) of the data from memory failures [51], [11]. To study the impact of the memory failures, we further investigate the individual cases of protecting 1, 2, or 3 MSBs and compare against the case without protecting any bits to see the

influence of the MSBs on the test classification accuracy. Figure 17 displays the test classification accuracy of  $\sigma=2$ ,  $\epsilon=8$  differentially private ConvNet with the varying amount of MSBs protected. The protection of 2 or 3 bits has a significant influence on boosting the accuracy of the system to acceptable levels.

### 3.3.3. Impact of Memory Failure on Privacy/Accuracy Trade-off

Additionally, the impact of the memory failure on the privacy/accuracy trade-off is studied in differentially private deep learning systems. It can be seen from Figure 18 (a) that, the parameter  $\epsilon$  represents the general trade-off between privacy level and accuracy of the differentially private deep learning system. A larger value can potentially enable higher accuracy. Additionally, for this specific Gaussian (i.e.  $\sim N(0, \sigma^2)$ ) noise addition mechanism, the value of  $\sigma$  also directly indicates the trade-off between privacy and accuracy. As shown in Figure 18 (a), in general, as  $\sigma$  (i.e. the amount of noise) increases, the accuracy decreases.

When comparing Figure 18 (a) and (b), it can be observed that for an optimal input data memory with MSBs protected, the accuracy/privacy tradeoff can be significantly improved. For example, in the case where  $\sigma=2$  and  $\epsilon=8$ , if the memory failure rate is 0.23, without protection, the accuracy will be much less than any acceptable amount (i.e. within 1% of the error free system). By introducing the protection to 2 MSBs, at the same failure rate, the accuracy will be increased to >96%, which is within 1% of the fault free differentially private model. In the following section, based on the design guidelines, a low power memory will be designed to minimize power consumption while keeping an acceptable accuracy for the differentially private deep learning systems.

### 3.3.4. Integer Linear Programs (ILP) Model based Memory Design

Based on the above analysis, we propose an input data memory design technique to improve the prediction accuracy of differentially private deep learning systems. To optimize the dataset quality, the design problem becomes an energy-accuracy-cost tradeoff design problem. We apply the model for hybrid SRAM without bitcell integration cost (i.e., Model 2) in [47] to handle this problem. In the following we provide an independent brief introduction to the mathematical model.

Assume the data points  $y_1, y_2, \dots, y_n$  are stored in a memory, and each data point needs  $s$  memory bitcells to store. Then, the mean square error (MSE) of these data points is defined by

$$MSE = \frac{1}{n} \sum_{i=1}^n \left( y_i^{(D)} - y_i^{(O)} \right)^2,$$

where  $y_i^{(D)}$  and  $y_i^{(O)}$  are the degraded and original data values, respectively. The degradations are caused by hardware memory failures. The expected MSE can be calculated by

$$E(MSE) = \frac{1}{ns} \sum_{i=1}^n \sum_{k=1}^s 4^k q_{ik},$$

where  $q_{ik}$  is the given failure probability of the  $k^{th}$  bitcell of the  $i^{th}$  data [47]. Note that the general concept of MSE is widely used in data analytics and statistics, not only in image or video pixels.

Suppose we have  $r_1$  and  $r_2$  design options for 6T and 8T SRAM, respectively. Let  $r = r_1 + r_2$  be the total design options. In addition, define binary decision variable

$$x_{ikl} = \begin{cases} 1, & \text{if option } l \text{ is chosen for the } ik^{th} \text{ bitcell} \\ 0, & \text{otherwise} \end{cases}$$

$$(i = 1, \dots, n; k = 1, \dots, s; l = 1, \dots, r)$$

Then the following ILP model can be formulated to enable an optimal input data memory using 6T sizing techniques and 8T+6T hybrid design:

$$\min_x \sum_{i=1}^n \sum_{k=1}^s \sum_{l=1}^r 4^k q_{ikl} x_{ikl} \quad (16)$$

$$\text{s.t.} \quad \sum_{l=1}^r x_{ikl} \geq 1, \quad i = 1, \dots, n; k = 0, \dots, s \quad (17)$$

$$\sum_{i=1}^n \sum_{k=0}^s \sum_{l=1}^r s_{ikl} x_{ikl} \leq s_{total} \quad (18)$$

$$x_{ikl} \in \{0,1\}, \quad i = 1, \dots, n; k = 1, \dots, s; l = 1, \dots, r \quad (19)$$

The objective function (16) is to minimize the expected MSE of the whole data set. Constraint (17) is to guarantee that one can choose exactly one design option for each bit cell. Note that since this is a minimization problem, (17) is equivalent to  $\sum_{l=1}^r x_{ikl} = 1$ . The total area constraint (18) assures that the total area of the design cannot exceed the given limit  $s_{total}$ , where  $s_{ikl}$  is a known parameter indicating the area cost of the  $ik^{th}$  bitcell if it is selected to adopt the  $l^{th}$  design option. Finally, constraint (19) indicates that  $x_{ikl}$  is a binary decision variable. The following section will present the memory design and evaluate results in a 45nm CMOS technology based on this optimization model. It should be noted that the developed ILP model can be used for optimal memory design in different technologies.

### 3.4. Embedded Memory Design for Deep Learning

To evaluate the effectiveness of the proposed memory design technique, 0.4V and 0.5V are used in our analysis based on a 45nm CMOS technology to enable the maximum energy efficiency at near-threshold voltage [48-49]. The deep learning system was set up using a single convolutional layer, a learning rate of 0.05, a batch size of 600, and an L2-norm gradient bound of 4.0 for norm clipping. The total epochs for any given privacy level are calculated during training and are based on the privacy parameters  $\epsilon$  and  $\delta$ , and the noise parameter  $\sigma$ . For

example, with large target  $\epsilon$  (i.e. less privacy) and/or large  $\sigma$  (i.e. more noise), the network model can be trained for more epochs without violating the chosen privacy level.

### 3.4.1. Optimized Memory Design

Traditional low-power memories often utilize bitcell sizing or more than 6T bitcells to reduce memory failures induced by process variations, thereby achieving power savings at low voltages. This is because, at low voltages, memory failures are mainly caused by the intra-die variations in process parameters (e.g., variations in channel length, channel width, oxide thickness, threshold voltage, line-edge roughness, and random dopant fluctuations [RDF]) and the inter-die variations (i.e. different process corners including “typical NMOS and typical PMOS”, “fast NMOS and slow PMOS”, “slow NMOS and fast PMOS”, “slow NMOS and slow PMOS”, and “fast NMOS and fast PMOS”) [50-51],[11]. Among the different sources of intra-die variations, RDF-induced threshold voltage ( $V_{th}$ ) variations are the most significant in causing memory failures [38], which can be expressed by

$$\sigma V_{th} = \sigma V_{th0} \sqrt{\frac{W_{min} L_{min}}{W L}} \quad (20)$$

where  $\sigma V_{th0}$  is the standard deviation of  $V_{th}$ , and  $W$  and  $L$  represent the width and length of the transistor, respectively.  $\sigma V_{th}$  for an NMOS and PMOS transistor with  $W$  equal to the minimum LEFF in the 45nm predictive technology is 46.9mV and 41.8mV, respectively.

According to (20),  $\sigma V_{th}$  is inversely proportional to  $\sqrt{WL}$ , indicating that the deviation of  $V_{th}$  is reduced as  $W$  and  $L$  increase. Therefore, upsizing bitcells can reduce memory failures at low voltages due to the reduced intra-die threshold voltage ( $V_{th}$ ) variations.



Table 9. Memory Failure Rate

Memory bitcells	Height ( $\mu\text{m}$ )	Width ( $\mu\text{m}$ )	Area ( $\mu\text{m}^2$ )	Area ratio $s_k$	Failure rate	
					@0.4V	@0.5V
6T: C61	0.45	1.523	0.685	1	0.5897	0.3436
6T: C62	0.45	1.563	0.703	1.026	0.5341	0.3074
6T: C63	0.45	1.603	0.721	1.053	0.4803	0.2771
6T: C64	0.45	1.643	0.739	1.079	0.4342	0.2521
8T: C81	0.45	1.669	0.751	1.096	0.0121	0.00082
8T: C82	0.45	1.700	0.765	1.117	0.0043	0.00009
8T: C83	0.45	1.740	0.783	1.143	0.002	0.00002

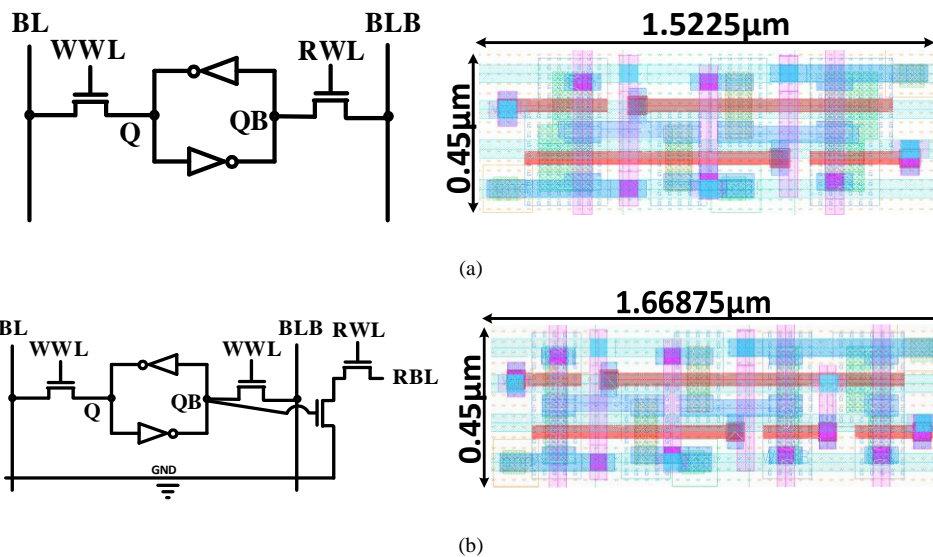


Figure 19. Different Memory Designs: (A) 6T SRAM Schematic and Minimum-Sized Layout Design in 45 nm Technology (C61) and (B) 8T SRAM Schematic and Minimum-Sized Layout Design (C81) in the Same 45 nm Technology.

In addition to upsizing bitcells, more than 6T bitcells can also mitigate process variation caused memory failures. Figure 19 shows the 6T bitcell and 8T bitcell width using 45 nm CMOS technology. As shown, 6T bitcells can achieve better area-cost performance while 8T can effectively reduce memory failures due to its decoupled read and write paths using two extra transistors. However, an 8T bitcell causes about 9.6% area overhead compared to 6T.

Memory failure rates are also strongly dependent on inter-die variations. Under inter-die variations, the dominant failures of 6T and 8T occur in read operations at “fs” (fast NMOS and slow PMOS) and in write operations at “sf” (slow NMOS and fast PMOS) process corners, respectively [11]. In our analysis, 10,000 Monte-Carlo simulations are performed with local intra-die threshold voltage variations (RDF effects) at the worst process corners for 6T and 8T bitcells. The failure rates are listed in Table 9. As shown, there are  $r=r_1+r_2=4+3=7$  total options (including 4 upsized 6T options and 3 upsized 8T options). As expected, upsizing bitcells and 8T options can both result in a lower failure rate with a larger bitcell area. Also, as the supply voltage decreases from 0.5V to 0.4V, the memory failure rate of the same bitcell increases accordingly.

Results of solving (16)-(19) for a variety of  $s_{total}$  values in the range [8.0, 9.1] using Gurobi solver (version 7.0.2) at both 0.4V and 0.5V are listed in Table 10. In the traditional design, all bitcells select the same option as discussed in [47]. It can be seen that in most design cases significant MSE improvement can be enabled with the optimal design, including over 99% MSE improvement for both 0.4V and 0.5V if the total area constraint is 8.5 or 8.7. Another interesting observation is that for two different voltages, the optimization solutions under the same total area constraint have the same tendency: when  $s_{total}$  is small (e.g.,  $< 8.3$ ), the most cost-efficient bitcell (C61) is usually selected to meet the area constraint. In the extreme case, with  $s_{total} = 8.0$ , all bitcells are C61, which is the only possible solution under such a strict area constraint. As  $s_{total}$  increases beyond 8.5, a larger number of different 8T bitcells are selected to optimize the quality.

Table 10. Results and Comparisons of Proposed Memory for Deep Learning

$S_{total}$	Optimal Design @ 0.5V									Traditional Scenario		MSE Improvement
	$MSE_{opt.}$	$S_7$	$S_6$	$S_5$	$S_4$	$S_3$	$S_2$	$S_1$	$S_0$	$MSE_{Trd.}$	$Des_{optn.}$	
8.0	12034.27	C61	C61	C61	C61	C61	C61	C61	C61	12034.27	C61	0.00%
8.1	3003.22	C81	C61	C61	C61	C61	C61	C61	C61	12034.27	C61	75.04%
8.3	200.77	C81	C81	C81	C61	C61	C61	C61	C61	10337.20	C62	98.06%
8.5	28.56	C81	C81	C81	C81	C81	C61	C61	C61	8997.34	C63	99.68%
<b>8.7</b>	<b>2.91</b>	<b>C83</b>	<b>C83</b>	<b>C82</b>	<b>C81</b>	<b>C81</b>	<b>C81</b>	<b>C61</b>	<b>C61</b>	<b>7944.11</b>	<b>C64</b>	<b>99.96%</b>
8.9	0.80	C83	C83	C82	C81	C81	C81	C81	C81	18.01	C81	95.56%
9.1	0.43	C83	C83	C83	C83	C83	C83	C82	C82	1.94	C82	77.84%
$S_{total}$	Optimal Design @ 0.4V									Traditional Scenario		MSE Improvement
	$MSE_{opt.}$	$S_7$	$S_6$	$S_5$	$S_4$	$S_3$	$S_2$	$S_1$	$S_0$	$MSE_{Trd.}$	$Des_{optn.}$	
8	26207.11	C61	C61	C61	C61	C61	C61	C61	C61	26207.11	C61	0.00%
8.1	6883.95	C81	C61	C61	C61	C61	C61	C61	C61	26207.11	C61	73.73%
8.3	735.63	C81	C81	C81	C61	C61	C61	C61	C61	22596.87	C62	96.74%
8.5	150.27	C83	C83	C82	C81	C61	C61	C61	C61	19329.43	C63	99.22%
<b>8.7</b>	<b>56.60</b>	<b>C83</b>	<b>C83</b>	<b>C82</b>	<b>C81</b>	<b>C81</b>	<b>C81</b>	<b>C61</b>	<b>C61</b>	<b>16710.36</b>	<b>C64</b>	<b>99.66%</b>
8.9	45.46	C83	C83	C83	C83	C82	C81	C81	C61	270.28	C81	83.18%
9.1	43.80	C83	C83	C83	C83	C83	C83	C82	C82	94.57	C82	53.69%

It should be noted that as  $S_{total} = 8.5$  or  $8.9$ , the optimal solutions for  $0.4V$  are different from the ones for  $0.5V$ . This is because, for different memory bitcells, the relationship between memory failure and voltage may not be linear [11].

### 3.4.2. Power Efficiency

We have also evaluated the power efficiency of the optimized memory design, as displayed in Table 11. All possible memory operations were considered for the total power estimation, including: read (i.e. read zero and read one), write (i.e. write zero to zero, zero to one, one to zero, and one to one), and hold (i.e. leakage power while holding zero and leakage power while holding one). As shown in Table 11, operating at  $0.4V$  enables significant power savings

as compared to the traditional supply voltage (1V). As the total area constraint,  $S_{total}$ , increases, the power consumption increases due to more 8T bitcells being included in the optimized design solution. If 8.7 is the target area constraint, then 74.82% and 86.11% power savings can be enabled at 0.5V and 0.4V compared to 1V, respectively.

### 3.4.3. Input Data Quality and Accuracy

We further evaluate the input data quality and prediction accuracy using the optimized memory. The results are listed in Table 12. The MNIST dataset [45], which was used as the original dataset for training the CNN model, displays almost no accuracy loss (0.01%) as compared to the fault free test samples. Additionally, the Fashion [52] and Kuzushiji-MNIST (KMNIST) [53] datasets are introduced to evaluate the efficiency of the proposed technique. The Fashion and KMNIST datasets are comprised of 28×28 grayscale images of 70,000 fashion product and Japanese characters, respectively, with each dataset containing samples from 10 categories. In both datasets the training set has 60,000 images and the test set has 10,000 images.

Table 11. Power Consumption of Optimized Memory at 45nm CMOS Technology @ 0.5V









































































$S_{total}$	Proposed optimal design		Traditional design			$P_{reduction}$ @ 0.4v (opt.) vs. 1v (Trd.)	$P_{reduction}$ @ 0.5v (opt.) vs. 1v (Trd.)
	$P_{opt.}$ (W) @ 0.4V	$P_{opt.}$ (W) @ 0.5V	$P_{Trd.}$ (W) @ 0.4V	$P_{Trd.}$ (W) @ 0.5V	$P_{Trd.}$ (W) @ 1.0V		
8	1.30E-06	2.07E-06	1.30E-06	2.07E-06	9.28E-06	86.03%	77.69%
8.1	1.41E-06	2.53E-06	1.30E-06	2.07E-06	9.28E-06	84.85%	72.74%
8.3	1.63E-06	3.01E-06	1.34E-06	2.15E-06	1.00E-05	83.74%	69.90%
8.5	1.74E-06	3.50E-06	1.38E-06	2.29E-06	1.16E-05	85.02%	69.83%
<b>8.7</b>	<b>1.96E-06</b>	<b>3.55E-06</b>	<b>1.42E-06</b>	<b>2.42E-06</b>	<b>1.41E-05</b>	<b>86.11%</b>	<b>74.82%</b>
8.9	2.07E-06	4.09E-06	2.18E-06	4.22E-06	1.02E-04	97.97%	95.99%
9.1	2.18E-06	3.85E-06	2.18E-06	3.87E-06	1.02E-04	97.86%	96.23%

$P_{opt.}$ : power consumption of the proposed memory;  $P_{Trd.}$ : power consumption of traditional memory design;  
 $P_{reduction}$ : power reduction

Both Fashion and KMNIST datasets serve as drop-in replacements for the MNIST dataset, as they share the same image sizes and number of classes. The complexity of the Fashion dataset is considered to be moderately more complex to classify than the MNIST dataset while KMNIST is significantly more complex. This level of dataset complexity is reflected in the classification accuracy results. When training a CNN model with the same architecture on the Fashion dataset, the proposed memory yields a negligible accuracy loss when voltage scaling to 0.5V (0.03%) or 0.4V (0.33%). When training a CNN model with the same architecture using the KMNIST dataset, a dataset that is significantly more difficult to classify, the proposed memory still yields negligible loss in classification accuracy when voltage scaling to 0.5V (0.15%) or 0.4V (0.59%).

The results in Table 12 are based on the specific privacy level where the maximum accuracy is enabled for the MNIST dataset (i.e.  $\sigma=2$ ,  $\epsilon=8$ ). With the same privacy level, using the proposed memory design, the accuracy almost remains the same for the Fashion and KMNIST datasets while the supply voltage is reduced from 1V to 0.4V.

Table 12. Input Data Quality and Accuracy

	No Error	1V Trd.	0.5V Trd.	This Work @ 0.5V	0.4V Trd.	This Work @ 0.4V
MNIST Dataset [45]						
						
						
						
Test Accuracy ( $\sigma = 2, \varepsilon = 8$ )	96.7%	96.67%	42.3%	96.69%	12.22%	96.6%
Fashion Dataset [52]						
						
						
						
Test Accuracy( $\sigma = 2, \varepsilon = 8$ )	87.1%	87.06%	31.75%	87.07 %	11.59 %	86.77%
KMNIST Dataset [53]						
						
						
						
Test Accuracy ( $\sigma = 2, \varepsilon = 8$ )	80.16%	79.87%	36.84%	80.01 %	14.99 %	79.57%

The Fashion and KMNIST datasets display higher accuracies for lower levels of  $\varepsilon$ , but still maintain high accuracy for varying levels of noise

### 3.4.4. Accuracy at Different Privacy Levels

To evaluate the impact of privacy levels on the effectiveness of the proposed memory technique, varying  $\sigma$  and  $\epsilon$  values are included in CNN model simulations. It shows that the privacy level has a noticeable impact on the inference accuracy of the differentially private deep learning systems. The MNIST, Fashion, and KMNIST datasets were used to determine the impact of the privacy level on the inference accuracy. In general, the higher the privacy level is, the lower the test accuracy becomes. This relationship can be seen in Table 13, which includes both high and low levels of privacy for comparison of test accuracy calculations. As displayed in Table 13, the proposed memory design at both 0.4V and 0.5V performs similarly to the 1V traditional design, and is capable of achieving inference accuracy within 1% of the fault free model at both low and high privacy levels. Therefore, the proposed memory can be a preferable solution for implementing power-efficient differently private deep learning systems.

Table 13. At Particular Privacy Parameters, the Impact of Privacy Level on Test Accuracy

Dataset	Privacy Parameters	Privacy/Noise Level	1V Trd.	0.5V Trd.	This Work @ 0.5V	0.4V Trd.	This Work @ 0.4V
MNIST	$\sigma = 4, \epsilon = 5$	High	95.89%	35.36%	95.91%	13.66%	95.74%
	$\sigma = 2, \epsilon = 10$	Low	96.52%	48.49%	96.39%	14.15%	96.34%
Fashion	$\sigma = 4, \epsilon = 5$	High	86.33%	27.53%	86.4%	11.25%	86.1%
	$\sigma = 2, \epsilon = 10$	Low	87.54%	20.14%	87.64%	10.33%	87.16%
KMNIST	$\sigma = 4, \epsilon = 5$	High	81.38%	25.14%	81.46%	11.11%	81.29%
	$\sigma = 2, \epsilon = 10$	Low	83.01%	36.13%	82.98%	13.89%	82.69%

## **4. FLEXIBLE LOW-COST POWER-EFFICIENT VIDEO MEMORY WITH ECC-ADAPTATION**

### **4.1. Introduction**

In this chapter, a flexible power-efficient video memory is presented that can dynamically adjust the strength of error-correction-code (ECC), thereby enabling power-quality trade-off based on application requirements. Specifically, we utilize the bit significance characteristics of video data to develop a low-cost parity storage scheme that supports both hamming code-74 (ECC74) and hamming code-1511 (ECC1511). Based on this, we design a flexible memory with three dynamic power-quality adaptation schemes (i.e., ECC74, ECC1511, and no ECC) to meet different video application requirements, which includes an integrated ECC encoder/decoder that handles both ECC74 and ECC1511 while reducing area overhead. The proposed memory results in significant power reduction without noticeable video quality degradation.

The main contributions of this work are as follows: a) runtime adaptation of ECC to improve video quality and b) ECC memory design with encoder and decoder integration to minimize area overhead.

### **4.2. State-of-the-Art**

Existing as critical hardware-building blocks for today's approximate computing platforms, video memories show application resilience to approximations with a trade-off between a "good enough" output and additional power savings. State-of-the-art, power-efficient video-specific memory can be broadly classified into two categories: design-time fixed quality or run-time adjustable quality.



#### **4.2.1. Video-Specific Memory with Design-Time Fixed Quality**

During the past decade, low-voltage video memories were widely investigated in the literature, and most existing solutions are designs with design-time fixed quality. For example, Chang et al. [8] presented a hybrid 6T+8T SRAM to achieve quality-power optimization. In [7], a heterogeneous sizing scheme was presented to reduce the failure probability of conventional 6T bitcells. In [9], the correlation between MSBs was utilized to design a hybrid 8T+10T memory for power savings. In [11], advanced data-mining techniques were used to identify useful video data characteristics (e.g., data association) for hardware design. At the same time, several recent works for analyzing the quality of videos, such as viewer experience, have recently been shown to outperform the traditional mean squared error (MSE) and PSNR [54]. Those video-specific memory designs enhanced power efficiency with a reduced implementation cost when compared to general-purpose memories [47]; however, the quality of those designs is fixed during design-time, so they lack run-time adaptation.

#### **4.2.2. Adaptive Memory with Dynamic Power-Quality Management**

There are several recent attempts to enable adaptive video memory with dynamic power-quality management. For example, the video memory presented in [55] used the least significant bits (LSBs) of video data to store the MSBs' error-correction-code (ECC). In [56], a video content-aware memory technique for power-quality trade-off was developed from viewers' perspectives, based on the influence of video macroblock characteristics on viewer experience. Additionally, in [57], a data-dependent reconfigurable conditional pre-charge (CP) SRAM was designed to utilize statistical dependencies present in the binary values. This paper presents a new low-cost adaptive-ECC video memory with dynamic power-quality trade-off. Our proposed adaptive-ECC video memory is orthogonal to existing viewer-aware or data-dependent adaptive

memories [56, 57], and therefore can be simultaneously utilized to further optimize power efficiency.

### 4.3. Proposed Low-Cost ECC Storage Scheme

#### 4.3.1. Traditional ECC

According ECC is a very popular technique to enhance the reliability of memory systems [58]. There are various types of ECCs that provide various levels of trade-offs between error correction capability and implementation cost. This paper utilizes the cost-effective hamming code-74 (ECC74) and hamming code-1511 (ECC1511) [55], detailed in Table 14, due to area constraints of the main use-case, video memory. Traditional ECC74 provides protection for 4 message bits, by requiring 3 parity bits to identify a faulty bit location, where each parity bit is generated using 3 message bits. Alternatively, ECC1511 protects 11 message bits with 4 parity bits, where each parity bit is generated using 7 message bits. For ECC74 and ECC1511, only 1 faulty message bit can be detected and corrected. If there are multiple faulty message bits, these two ECC algorithms cannot determine that, and may incorrectly “correct” a message bit.

To provide more context for the ECC74 and ECC1511 algorithms, using Table 14 as a visual aid, M0-M14 are message bits and P1-P4 are parity bits. Parity bits are placed on the  $2^n$  ( $n=0,1..$ ) positions [58]. The calculation of parity bits and error correction bits are based on a specific Hamming code sequence, as expressed below:

$$P_{1,74} = 3, 5, 7 = M2 \oplus M4 \oplus M6 \quad (21)$$

$$P_{2,74} = 3, 6, 7 = M2 \oplus M5 \oplus M6 \quad (22)$$

$$P_{3,74} = 5, 6, 7 = M4 \oplus M5 \oplus M6 \quad (23)$$

The three parity bits for ECC74 are generated by performing XOR ( $\oplus$ ) operations according to Equations (1)-(3), and utilizing even parity. To determine a faulty bit after memory storage for ECC74, the calculation of error correction bits are expressed in Equations (4)-(6), where  $P_N$  is the previously calculated parity stored and then read back, and  $P_{N\_74}$  is the recalculated parity from the read back message data bits:

$$E_{1\_74} = P1 \oplus P_{1\_74} \quad (24)$$

$$E_{2\_74} = P2 \oplus P_{2\_74} \quad (25)$$

$$E_{3\_74} = P3 \oplus P_{3\_74} \quad (26)$$

If the binary to decimal conversion of  $(E_{3\_74}, E_{2\_74}, E_{1\_74})$  is evaluated as zero, then there is no error; otherwise, the bit-flip error position shall be determined by the evaluated decimal number. For example, suppose  $E_{3\_74} = '1'$ ,  $E_{2\_74} = '1'$ , and  $E_{1\_74} = '0'$ , then the faulty bit position is  $110_2$  which corresponds to the 6<sup>th</sup> bit, M5, having a bit-flip error; hence, M5 is toggled to correct the error.

A major disadvantage with traditional ECCs is their significant cost overhead, caused mainly by additional bitcells required for storing parity bits. For ECC74, protecting 4 message bits requires 3 parity bits: a 75% silicon area overhead. For ECC1511, protecting 11 message bits requires 4 parity bits: a 36% silicon area overhead.

Table 14. ECC Sequence and Message Bit Placement for Traditional ECC74 and ECC1511

Traditional ECC74							
$2^3$	7	6	5	$2^2$	3	$2^1$	$2^0$
N/A	M6	M5	M4	P3	M2	P2	P1
Traditional ECC1511							
$2^3$	7	6	5	$2^2$	3	$2^1$	$2^0$
P4	M6	M5	M4	P3	M2	P2	P1
$2^4$	15	14	13	12	11	10	9
N/A	M14	M13	M12	M11	M10	M9	M8

Therefore, this excessive overhead eliminates traditional ECCs for data integrity solutions in embedded memory designs, as the use-case requires optimized resource allocation. Instead of additional parity bits for message protection, if we can identify a use-case where the parity bits can be embedded within the message bits, then the area overhead requirements of ECCs can be eliminated. One such use-case is videos, where we can replace legitimate message bits with parity bits, and by doing so trade-off video quality.

For example, suppose we directly apply the traditional ECC scheme in Table 14 to the memory system in Table 15. As shown in Table 15, S0 to S15 are the bit positions for bytes 1 and 2. S0 (Least Significant Bit [LSB]) to S7 (Most Significant Bit [MSB]) are the bit positions of the 1<sup>st</sup> byte and S8 (LSB) to S15 (MSB) are the bit positions of the 2<sup>nd</sup> byte of the memory array, respectively. The three parity bits for ECC74 are stored in S3, S1, and S0. The four parity bits for ECC1511 are stored in S7, S3, S1, and S0. Figure 20 (a), (b), and (c) present the original video frame, encoded video frame with traditional ECC74, and encoded video frame with traditional ECC1511, respectively. Peak Signal-to-Noise Ratio (PSNR) is a widely adopted video quality evaluation metric, where a higher PSNR value translates to better video frame quality.

Storing the parity bits using the traditional ECC schemes results in significant video quality degradation, due in part to video data carrying more quality weight in the MSBs. The PSNR of the encoded frames with traditional ECC74 and ECC1511 are 27.75dB and 8.27dB, respectively. One observes from Figure 20 (c), ECC1511 encoding results in larger video quality loss as opposed to ECC74 due to its 4<sup>th</sup> parity bit being stored in the MSB of the first byte. Consequently, to ensure the least amount of video quality degradation, video data bit significance characteristics should be considered, such that LSBs are favored for parity storage.



(a) Original Frame      (b) Encoded ECC74      (c) Encoded ECC1511

Figure 20. Video Output Quality with Traditional ECC. (A) Original Frame, (B) ECC74 Parity Bits Stored with PSNR = 27.75 dB, and (C) ECC1511 Parity Bits Stored with PSNR = 8.27 dB.

Table 15. Impact of Traditional ECC (Parity Bit Position) on Video Storage

Traditional ECC74: Byte 1								
Memory bits	S7	S6	S5	S4	S3	S2	S1	S0
Data	M7	M6	M5	M4	P3	M2	P2	P1
Traditional ECC1511: Byte 1								
Memory bits	S7	S6	S5	S4	S3	S2	S1	S0
Data	P4	M6	M5	M4	P3	M3	P2	P1
Traditional ECC1511: Byte 2								
Memory bits	S15	S14	S13	S12	S11	S10	S9	S8
Data	M15	M14	M13	M12	M11	M10	M9	M8

### **4.3.2. Bit Significance Characteristics of Video Data and Proposed Storage Scheme for Parity Bits**

As opposed to typical fault-tolerant applications, video data has bit significance characteristics where MSBs have a greater contribution to output quality than LSBs. According to recent literature, the video memory presented in [55] uses the LSBs to store the MSBs' parity bits, thereby effectively reducing video quality degradation overhead; this is what we use as a basis for comparison. As a caveat however, only ECC1511 was considered in [55]. In this paper, we propose a flexible memory with three dynamic power-quality adaptation schemes to meet the various requirements of video applications, i.e., ECC74, ECC1511, and no ECC. We also design an integrated ECC encoder/decoder that handles both ECC74 and ECC1511, which further reduces area overhead.

Our proposed storage scheme is applied for two-byte memory, as illustrated in Table 16. If ECC1511 is selected, then there would be four parity bits stored in the LSBs (i.e., P1, P2, P3, P4) and eleven protected MSB message bits (i.e., M7 to M2, M15 to M11). Alternatively, if ECC74 is selected, then there would be three parity bits stored in the LSBs (i.e., P1, P2, P3) and four protected MSB message bits (i.e., M6, M7, M14, M15).

Table 16. Proposed ECC (Message Bit and Parity Bit Placement)

Proposed ECC74: Byte 1								
memory bits/sequence	S7	S6	S5	S4	S3	S2	S1	S0
	3	6	9	11	13	15	2 <sup>1</sup>	2 <sup>0</sup>
Data	M7	M6	M5	M4	M3	M2	P2	P1
Proposed ECC74: Byte 2								
memory bits/sequence	S15	S14	S13	S12	S11	S10	S9	S8
	5	7	10	12	14	16	8	2 <sup>2</sup>
Data	M15	M14	M13	M12	M11	M10	M9	P3
Proposed ECC 1511: Byte 1								
memory bits/sequence	S7	S6	S5	S4	S3	S2	S1	S0
	3	6	9	11	13	15	2 <sup>1</sup>	2 <sup>0</sup>
Data	M7	M6	M5	M4	M3	M2	P2	P1
Proposed ECC 1511: Byte 2								
memory bits/sequence	S15	S14	S13	S12	S11	S10	S9	S8
	5	7	10	12	14	16	2 <sup>3</sup>	2 <sup>2</sup>
Data	M15	M14	M13	M12	M11	M10	P4	P3

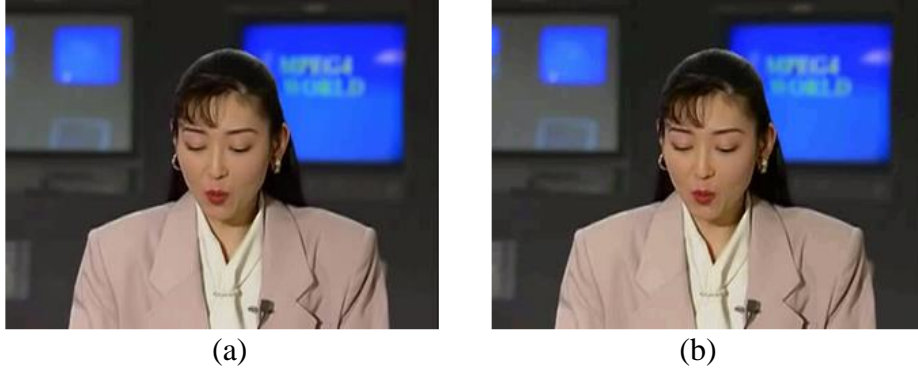


Figure 21. Encoded Video Frame (Akiyo) with (a) ECC74 Where Parity was Stored in the LSBs with PSNR = 41.2582 and (b) ECC1511 Where Parity Bits were Stored in LSBs with PSNR = 39.8426 dB.

$$P_1 = 3, 5, 7, 9, 11, 13, 15 \quad (27)$$

$$= M7 \oplus M15 \oplus M14 \oplus M5 \oplus M4 \oplus M3 \oplus M2$$

$$P_2 = 3, 6, 7, 10, 11, 14, 15 \quad (28)$$

$$= M7 \oplus M6 \oplus M14 \oplus M13 \oplus M4 \oplus M11 \oplus M2$$

$$P_3 = 5, 6, 7, 12, 13, 14, 15 \quad (29)$$

$$= M15 \oplus M6 \oplus M14 \oplus M12 \oplus M3 \oplus M11 \oplus M2$$

$$P_4 = 9, 10, 11, 12, 13, 14, 15 \quad (30)$$

$$= M5 \oplus M13 \oplus M4 \oplus M12 \oplus M3 \oplus M11 \oplus M2$$

Equations (7) to (10) calculate the parity bits for the proposed scheme. The symbols in Equations (7) to (10), (3, 5, 7..),  $M$ , and  $P$ , indicate the ECC sequence, message bit, and parity bit, respectively.

Since ECC74 doesn't require the 4th parity bit,  $P_4$ , this is disabled when using ECC74. Furthermore, to reduce circuit area overhead, the encoders and decoders were developed for ECC1511 then reused for ECC74 by only using the green colored bits in Equations (7) to (9) to calculate  $P_1$ - $P_3$  when ECC 74 is selected. The encoder/decoder design is detailed in Section 4.5.



Figure 21 (a) and (b) present the video output quality using the proposed ECC storage scheme from Table 15. Figure 21 (a) shows the video quality PSNR metric for ECC74 as 41.26dB with 2 parity bits stored in the 2 LSBs of the 1st byte and 1 parity bit stored in the LSB of the 2nd byte. Figure 21 (b) shows the video quality PSNR metric for ECC1511 as 39.84dB with 2 parity bits stored in the 2 LSBs of both the 1st and 2nd bytes. Since ECC1511 needs to sacrifice one extra LSB to store its parity bits, it results in a lower PSNR value than ECC74. Hence, the proposed ECC parity storage scheme from Table 16, as shown in Figure 21, which supports both ECC74 and ECC1511, has much better potential to significantly improve video quality compared to the traditional ECC scheme from Table 15, as shown in

Figure 20. Based on this parity storage scheme, an adaptive ECC mechanism is proposed to meet various requirements of video applications, as discussed in the next section.

#### **4.4. ECC Adaptation Based on Requirements and Failure Rate Based on Voltage Scaling**

In this section, an adaptive ECC mechanism is presented, which supports three ECC conditions, no ECC, ECC74, and ECC1511. First, SRAM failure characteristics were studied using a 45 nm CMOS technology. Then, the impact of memory failures on video quality was analyzed, including failures in parity bits. Finally, a memory failure based adaptive ECC mechanism was developed.

##### **4.4.1. Failure Characteristics of 6T SRAM**

Figure 22 graphs the failure rate of a 45 nm 6T SRAM bitcell at an increasing voltage range between 500mV and the technology's 1.0V nominal supply voltage, with 5mV increments. The failure rate was measured with 10,000 Monte Carlo simulations at the worst process corner for 6T bitcells, fast NMOS and slow PMOS (FS). As expected, the failure rate increased rapidly as the supply voltage was reduced. The failure rate was about 7.87% at 500mV; and when the

supply voltage was scaled-up to 685mV, the failure rate was about 0.48%. There were no errors in the memory system when the supply voltage was 795mV or above. Next, we studied the impact of memory failures on video output quality.

#### 4.4.2. Errors Injected, Including in Parity Bits

We first analyzed the impact of memory failures on video quality with failures injected in all bits, including parity bits, at a uniform random distribution of 0.1%, on the well-known video sequence – Akiyo.

Figure 23 illustrates the error mapping and quality of a video that was stored in a 65536 word  $\times$  16bit SRAM array, with supply voltage at 665mV, using our proposed ECC74 scheme.

Figure 23 (a) and (b) demonstrate the error distribution of the SRAM memory with 0.1% failures injected in the original image. In Figure 23 (a), the dots, two of which are circled in blue as an example, represent the error positions in the SRAM memory array.

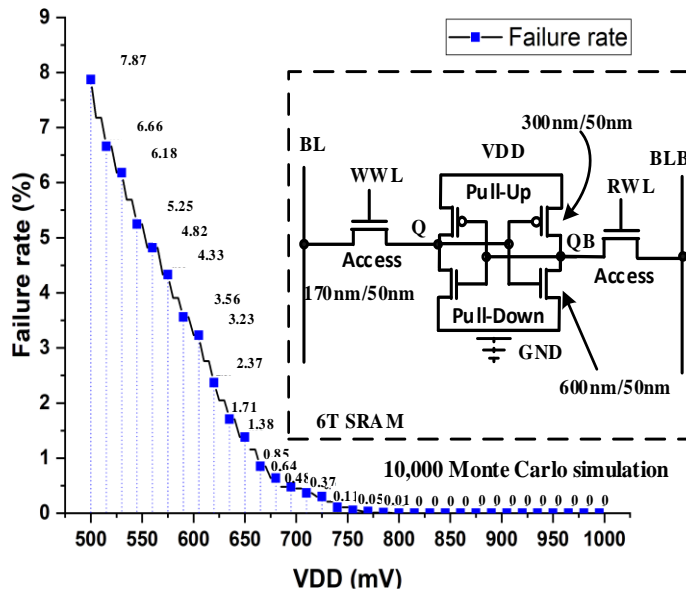


Figure 22. Relation Between Supply Voltage (VDD) and SRAM bitcell Failure Rate in a 45nm CMOS Technology.

As can be seen, the memory failures were distributed uniformly in the MSBs and LSBs, and the PSNR was 31.4524dB, as shown in Figure 23 (b). In Figure 23 (c), the parity bits of ECC74 were stored in the LSBs of the SRAM array, based on the proposed parity storage scheme. Specifically, the two LSBs of the 1st byte and one LSB of the 2nd byte were utilized to store the parity bits. The video output quality with ECC encoding is illustrated in Figure 23 (d), having a PSNR of 41.2582dB. After storing the parity bits in the memory, the exact same number and position of errors from Figure 23 (a) were injected into the memory in Figure 23 (e), resulting in a PSNR of 30.0148dB. Finally, after decoding, ECC74 could correct one error in either of the 2 MSBs of every two sequential bytes of the SRAM memory array, as shown in Figure 23 (g), resulting in a 7.57dB PSNR improvement compared to the original image without ECC74 (i.e., Figure 23 (b) vs. (h)). Note that the error map in Figure 23 (g) shows some new errors, circled in purple, due to injected memory failures in the parity bits, which result in an incorrect ECC correction.

Figure 24 illustrates the error map and the video output quality for ECC1511. It can be seen from Figure 24 (a) that the additional LSB parity bit caused a slight image quality degradation compared to ECC74 (i.e., Fig. 24 (b) with a PSNR of 39.8426dB vs. Figure 24 (d) with a PSNR of 41.2582dB). Figure 24 (c) and (d) used the same 0.1% error map used for the previous ECC74 analysis, resulting in the PSNR being slightly degraded to 30.8653dB, due to the extra parity bit. Figure 24 (e) shows that the decoder circuit corrected most of the injected errors in the 11 protected MSBs, resulting in a PSNR of 39.2536dB in Figure 24 (f), which is slightly higher than when using ECC74 (i.e., Figure 23 (h)). Hence, even though ECC1511 sacrifices one extra LSB for parity, its stronger error-correction ability improves video quality by correcting more MSB errors.

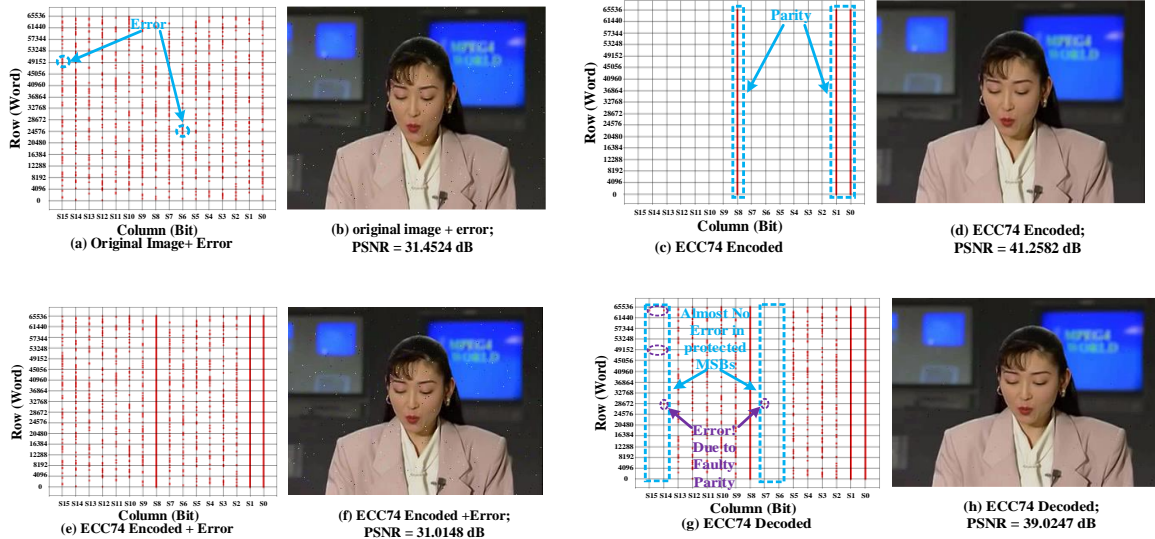


Figure 23. Error Map and Stored Video Frame with Proposed ECC74 Under 0.1% Faulty Memory Bitcells.

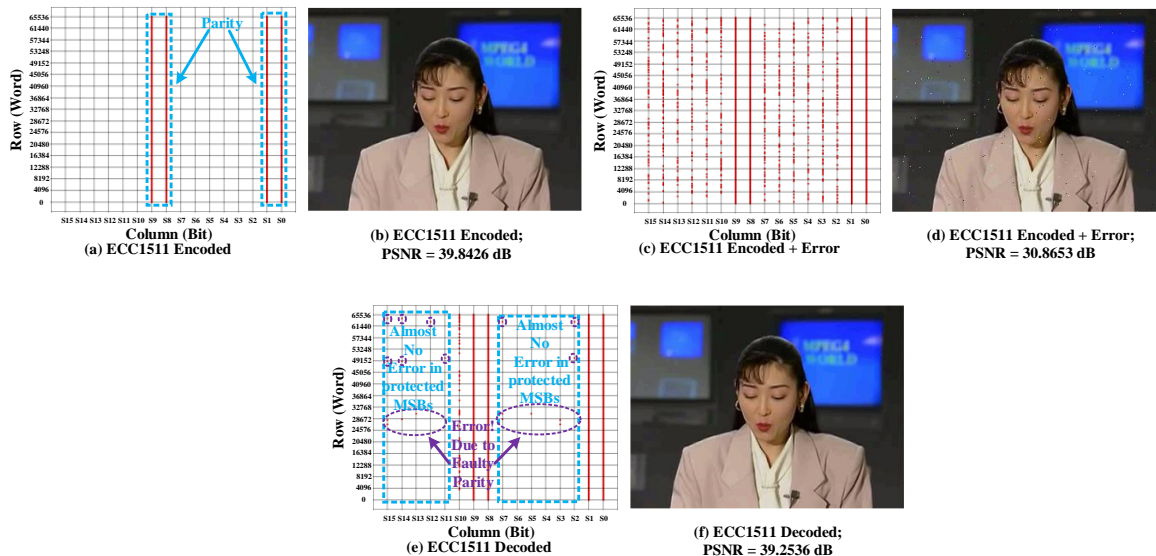


Figure 24. Error Map and Stored Video Frame with Proposed ECC1511 Under 0.1% Faulty Memory Bitcells.

So far, we analyzed the proposed ECC scheme at a low 0.1% failure rate; in the next subsection we continue analysis of its performance at different failure rates.

#### 4.4.3. ECC Under Various Failure Rates

To further analyze the performance of our proposed ECC schemes at various failure rates with different Bit videos, 100 videos were randomly selected from YouTube-8M [59] for evaluation.

The output quality of the same randomly selected frame from each video was tested for a range of failure rates between 0.01% to 1%, as shown in Figure 25. It can be seen that both ECC schemes significantly increase video quality compared to without ECC, except for when the failure rate is less than or equal to 0.01%, since for very low failure rates, the message bits replaced by parity bits cause more PSNR loss than PSNR gain due to faulty bit corrections. When the memory failure rate is between 0.01% and 0.05%, ECC74 performs best, since the additional message bit replaced by ECC1511's 4<sup>th</sup> parity bit causes more PSNR loss than PSNR gain due to additional faulty bit corrections. When the memory failure rate is greater than 0.05% and less than 0.6%, ECC1511 performs best, since the PSNR gained by its increased faulty bit correction outweighs the PSNR loss due to its extra parity bit. However, for failure rates over 0.6% ECC74 is best due to its reduced possibility of multi-bit errors compared to ECC1511. As expected, video quality degrades as memory failure rate increases, even for the ECC schemes, since they can only correct a single error bit for every 2-bytes of memory. As the failure rate increases, the likelihood of multiple bit errors per 2-bytes also increases, which cannot be corrected with either ECC74 or ECC1511. Hence, the proposed ECC scheme performs best when the error rate is within an acceptable range, where the likelihood of multi-bit errors is small.

Next, a memory failure based ECC scheme is proposed to enable runtime adaptation.

#### **4.4.4. Proposed Runtime ECC Adaptation Scheme**

According to [35], video quality is deemed acceptable when PSNR is 30dB or higher. Since ECC74 has a PSNR of 29.88755dB at ~1%, this was within the acceptable range. Hence, if the memory works at its nominal supply voltage or the error rate is lower than 0.01%, no ECC is needed. For failure rates between 0.01% and 0.05% ECC74 should be selected.

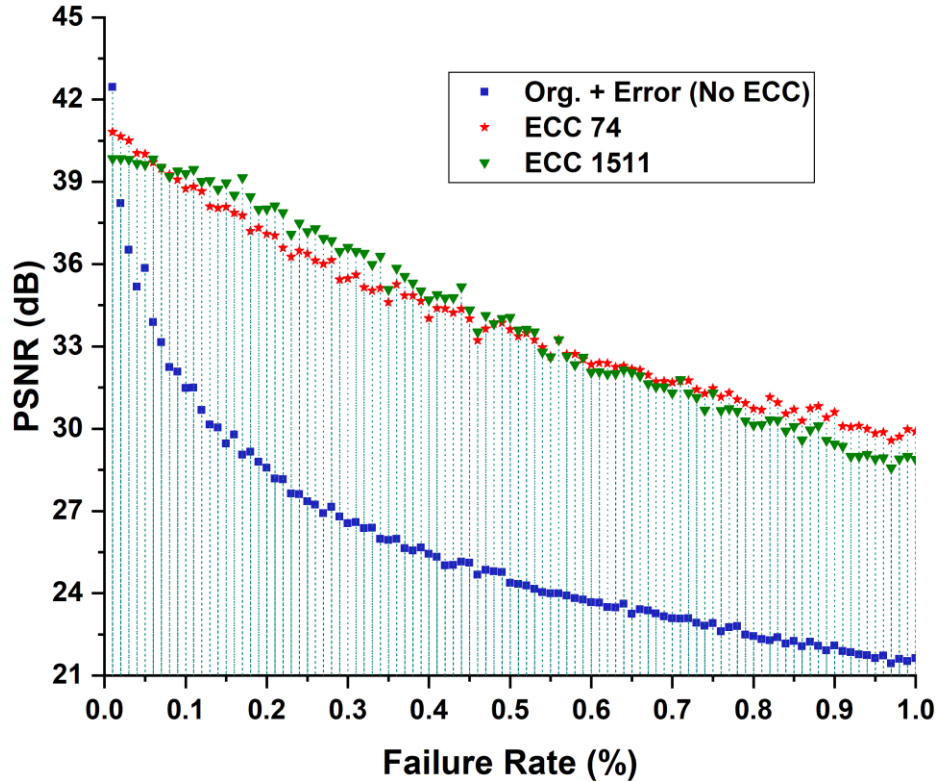


Figure 25. ECC Adaptation Based on Failure Rate and Corresponding PSNR.

As the failure rate continues to increase (between 0.05% and 0.6%), ECC1511 should be utilized due to its stronger error correction ability. And, when the failure rate is above 0.6%, both ECC74 and ECC1511 cannot correct multiple bit errors, but ECC74 should be selected since it has fewer parity bits that can have errors that cause incorrect ECC correction, and therefore performs better. The next section describes the hardware implementation of this proposed runtime ECC adaptation scheme.

#### 4.5. Proposed Memory

Figure 26 presents the architecture of the proposed adaptive ECC memory with its bitcells organized in four 1024×16 bit sub-blocks. Based on the traditional memory structure, ECC Encoder/Decoder, Correction Unit, and Output MUX were needed to enable ECC adaption. For each read/write operation, a 4-to-1 multiplexer with two control signals (S1 and S0) is used

to select the correct operation as follows: (i) when S1 and S0 are “00”, ECC1511 is activated; (ii) when S1 and S0 are “10”, ECC74 is selected; and (iii) when S1 and S0 are “11”, no ECC is selected and a normal read/write operation is executed. An S1 and S0 of “01” is invalid and will not occur in a properly operating system; however, if this does occur for some reason, a normal read/write operation without ECC will be executed. As shown in Figure 26, the input data [15:0], excluding M10, is provided to the encoder to generate the parity bits or pass the original LSB message data, depending on the control signals, *S0* and *S1*. Then, the data is sent to the memory for storing. During this process, the first two LSBs of both pixels/bytes may be replaced with parity bits after encoding, depending on which of the 3 ECC schemes is selected (i.e., if ECC74 is selected, M0, M1, and M8 are replaced with P1, P2, and P3, respectively; if ECC1511 is selected, in addition to the M0, M1, and M8 replacements, M9 is also replaced with P4; and if no ECC is selected, then no message bits are replaced). When reading from the memory, the data is sent to the decoder and correction unit circuitry to check for, and correct a faulty bit if needed, respectively. If either ECC scheme was selected, then the Output MUX selects the final output from the Correction Unit; otherwise, it selects the memory output as the final output. To

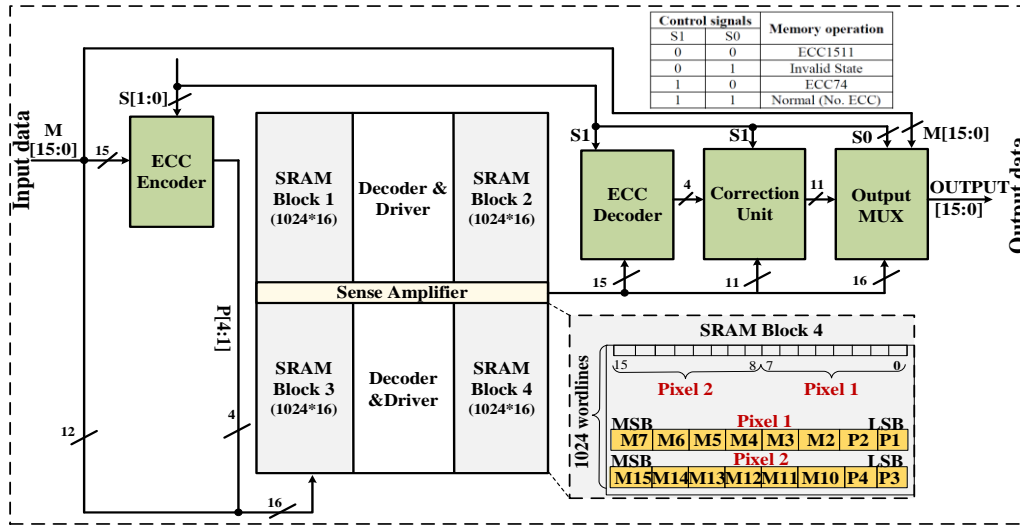


Figure 26. Proposed Adaptive ECC Memory.

minimize implementation cost, the ECC encoder/decoder were designed to reuse circuitry for both ECC1511 and ECC74, which is discussed next.

#### 4.5.1. Reusable ECC Encoder for ECC1511 and ECC74

Figure 27 shows the integrated ECC Encoder for ECC1511 and ECC74. There is a 15-bit message input (M[15:0], excluding M10) and two control signals (S0 and S1) for ECC selection; and the encoder generates four parity bits (P[4:1]) for calculation of error bit detection and correction, only if ECC is selected. If ECC is selected (i.e., S0 = 0), then Vdd1! is enabled to supply the ECC74 encoding circuitry (i.e., the first 2 XOR gates in the first 3 XOR chains); if S[1:0] = "00", then Vdd2! is also enabled to supply the additional circuitry needed for ECC1511 encoding (i.e., the 4th XOR chain and the rest of the XOR gates in the first 3 XOR chains); otherwise (i.e., S[1:0] = "01" or "11") both Vdd1! and Vdd2! are kept at ground so that the encoder circuitry is inactive, therefore conserving power, since ECC is not being utilized. The 4 output MUXes then select which parity bit or original message bit to store in the



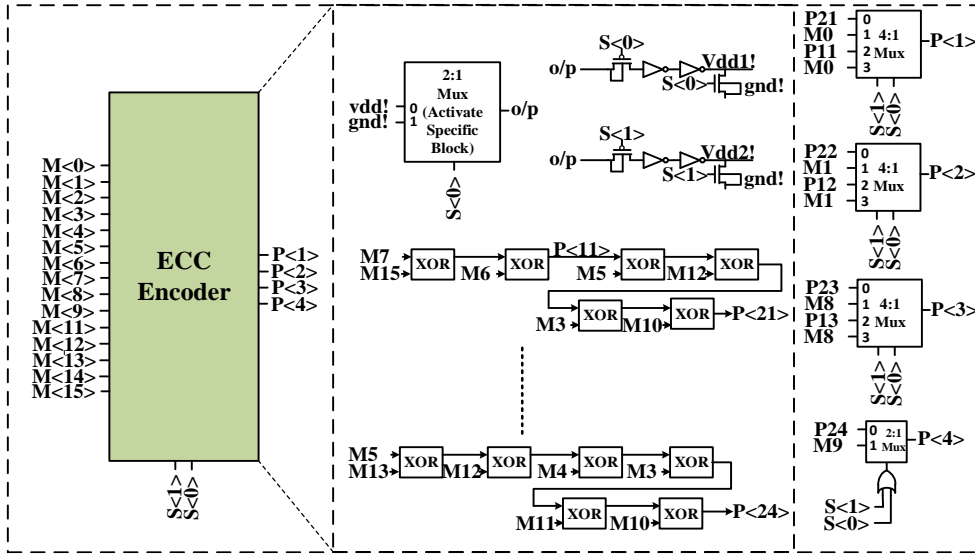


Figure 27. ECC Encoder.

2 LSBs of both bytes of memory. As an example, if ECC74 was activated, then P11, P12, P13, and M9 would be stored in P1, P2, P3, and P4, respectively.

#### 4.5.2. Reusable ECC Decoder for ECC1511 and ECC74

Figure 28 shows the integrated ECC Decoder design. Its input signals include data signals (M2...M7) and (M11...M15), parity bits (P1...P4), and one control signal, *SI*. It generates seven internal signals (E13, E12, E11 for ECC74 and E24, E23, E22, E21 for ECC1511), which are then grouped into a 4-bit number, E[4:1], which represents the error bit position, if any, in the 2 memory bytes. For example, if E[4:1] = “0111”, then message bit 14 is faulty according to Table 16 (i.e., 7 corresponds to M14), and would be toggled in the subsequent Correction Unit. Similar to the Encoder, Vdd1! is used to supply the XOR gates that generate E13, E12, E11 and the output MUXes that generate E[4:1], and Vdd2! is used to supply the additional XOR gates needed to generate E24, E23, E22, E21, in order to conserve power.

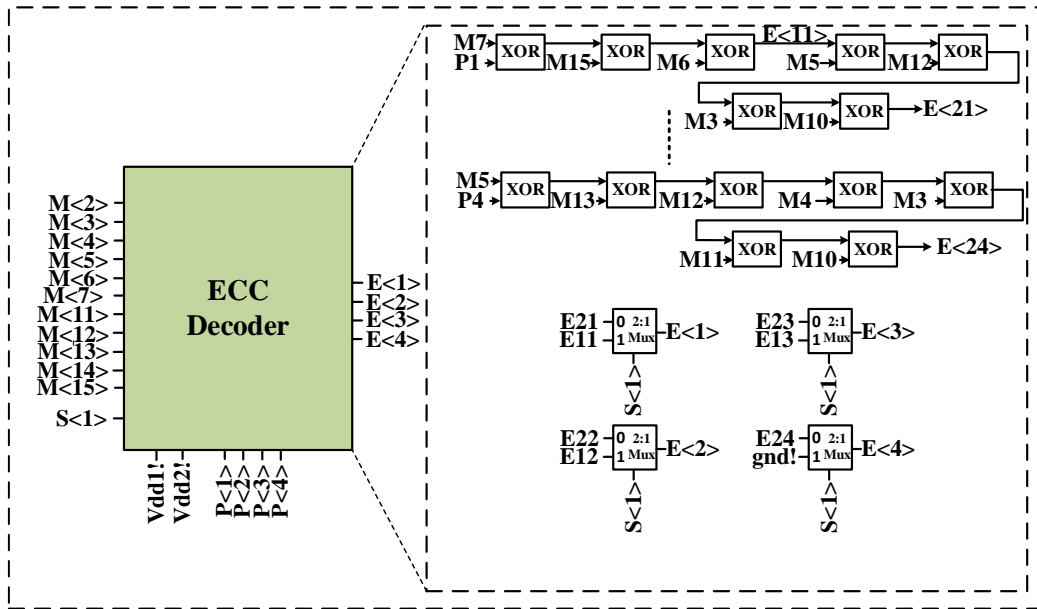


Figure 28. ECC Decoder.

### 4.5.3. Correction Unit

Figure 29 presents the error bit correction unit, which flips a message bit identified as faulty by  $E[4:1]$ . An active high 4-to-16 decoder is used to select a specific faulty bit location by asserting  $In$ , which is input to an XOR gate along with its corresponding message bit, such that the message bit is flipped when  $In$  is asserted. For example, the top most  $In$  in Figure 29 is asserted when  $E[4:1] = "1111"$ , which according to Table 16 corresponds to  $M2$ ; hence, its corresponding XOR gate input is  $Out(2)$ , and that XOR gate's output is  $D\_out(2)$ . Note that message bits  $M10$ ,  $P1$ ,  $P2$ ,  $P3$ , and  $P4$  can never be corrected; hence,  $out[10:8]$  and  $out[1:0]$  are passed directly to  $D\_out[10:8]$  and  $D\_out[1:0]$ , respectively.

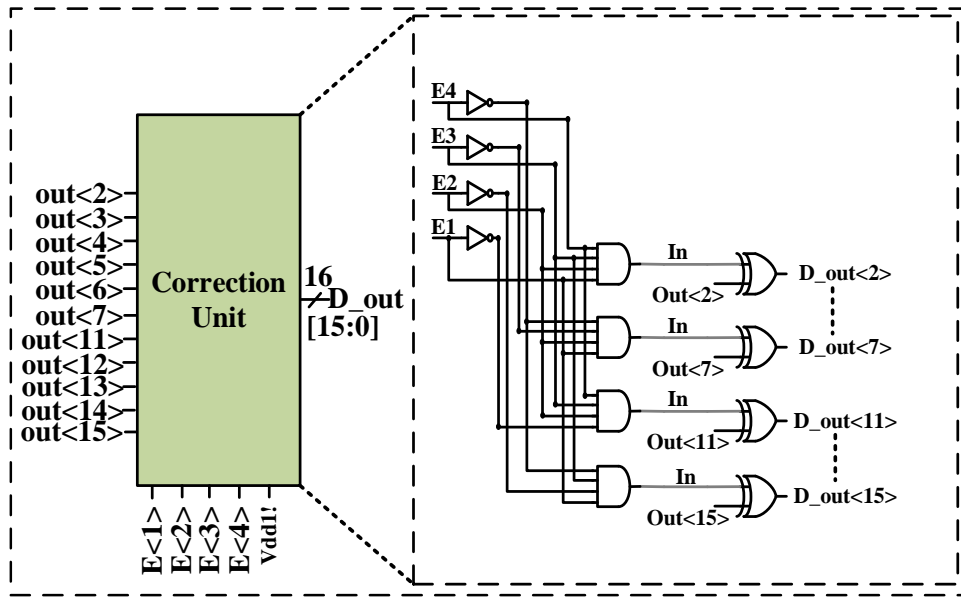


Figure 29. Correction Unit.

#### 4.5.4. Output MUX

Figure 30 shows the output MUX, which selects between the potentially ECC corrected bits and the original SRAM bits for bit positions 15-11 and 7-2, depending on whether ECC was selected or not. Bit positions 10-8 and 1-0 are always the original SRAM bits, as mentioned above, so no MUX is required for these.

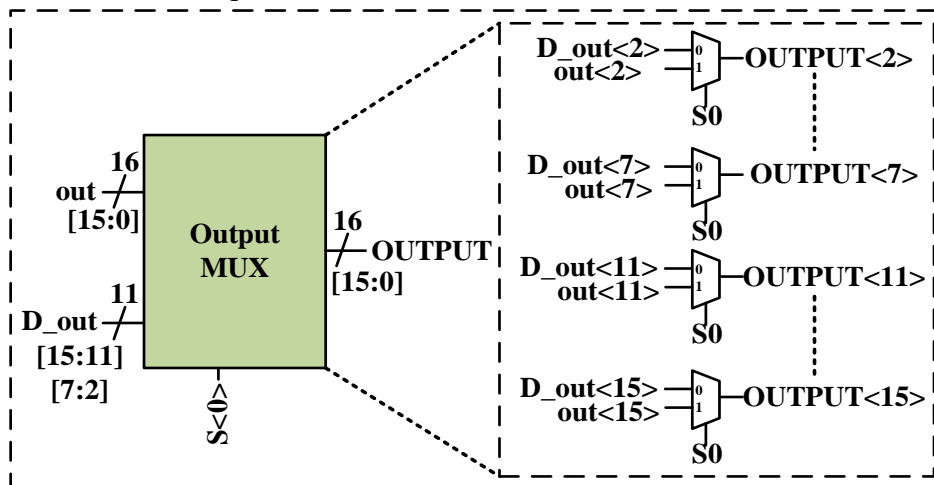


Figure 30. Output MUX.

## 4.6. Results

### 4.6.1. Timing Diagram

Figure 31 presents the timing diagram for the proposed memory, showing three segments of simulation waveforms, No ECC (i.e., normal memory operation), ECC74, and ECC1511. Specifically, Figure 31 shows (a) the input data, (b) the data after encoding, (c) the error correction bit information, and (d) the data after decoding.

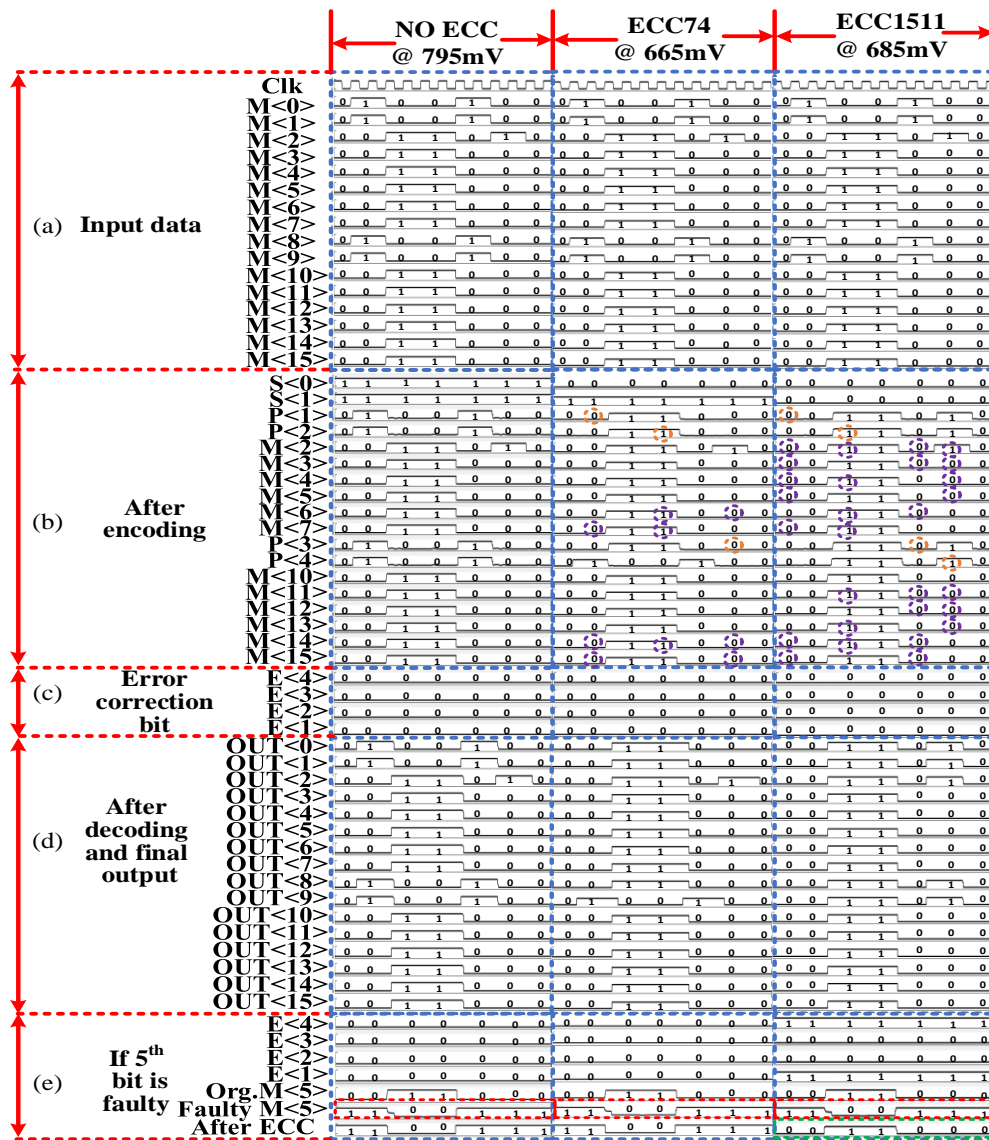


Figure 31 Timing Diagram.

At first, the input data was applied to the memory and after the ECC encoder, the generated parity bits were sent to the memory block to be stored as the LSBs. As shown in Figure 31 (b), the purple marked bits were the combination that generated parity bits (orange marked) for that specific operation. To calculate parity for ECC1511 and ECC74, seven and three input bits were needed, respectively. Finally, the memory checked the error correction bit positions in Figure 31 (c), and if all zeros, then no error was detected. Otherwise, the correction unit toggled the faulty bit. Since Figure 31 (c) is always all zeros, there were no faulty bits, such that Figure 31 (b) and (d) are the same. Figure 31 (e) illustrates the case when one faulty bit was stored in place of the original message bit, M5. Since neither No ECC nor ECC74 protects M5, their error correction E[4:1] is all zeros, and their M5 output after ECC is the same as the inserted faulty M5 bit. However, for ECC 1511,  $E[4:1] = 1001_2 = 9_{10}$ , which corresponds to M5 according to Table 16. Hence, the faulty M5 is flipped resulting in M5 after ECC being the same as the original M5.

#### **4.6.2. Power Efficiency**

For power analysis, we utilized the same 45nm CMOS process discussed in Section 4.4.2. For each testcase shown in Figure 32, the average power consumption was measured for writing  $FF00_{16}$  to a random word in a  $128 \text{ word} \times 16 \text{ bit}$  memory bank, which was initialized to  $A5A5_{16}$ , followed immediately by reading  $FF00_{16}$  from the same word, such that all read/write memory operations were equally included (i.e., reading ‘0’ and ‘1’, and writing ‘0’ to ‘0’, ‘0’ to ‘1’, ‘1’ to ‘0’, and ‘1’ to ‘1’). Our baseline design is the traditional SRAM without any of the additional ECC circuitry, operating at 795mV, the lowest voltage that did not induce errors, which required an average power of  $2.94E-1\text{mW}$ .

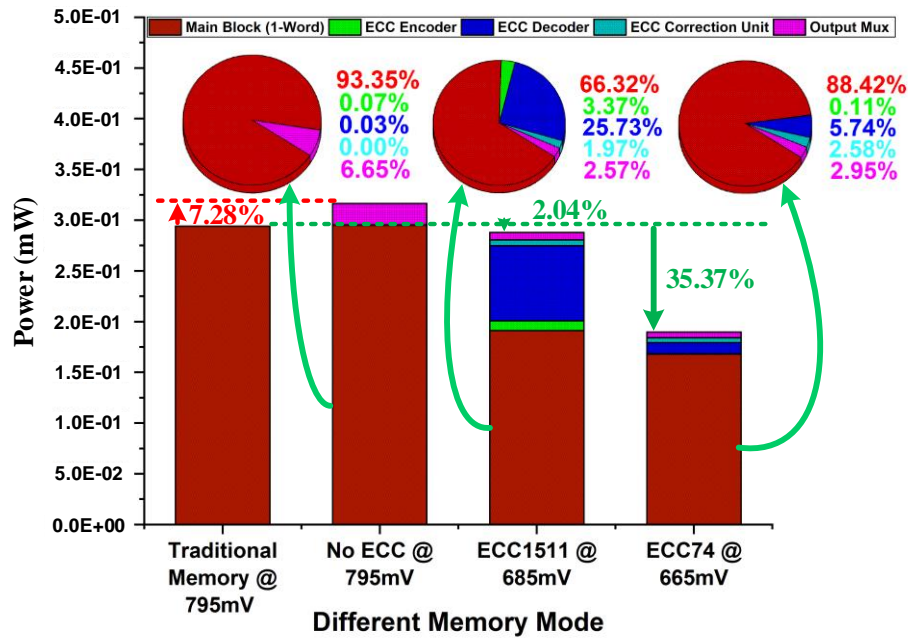


Figure 32. Power Comparison of Proposed ECC Memory with Traditional Memory.

Our first testcase is the proposed ECC memory operating at 795mV, where No ECC is selected, which shows that the added ECC circuitry only requires an additional 7.28% power when not being utilized. Our next testcase is the proposed ECC memory operating at 685mV, the lowest voltage where ECC1511 would be invoked, which consumed 2.04% less power than the baseline design. Our final testcase is the proposed ECC memory operating at 665mV, the lowest voltage where the failure rate would still be less than 1%. In this case, ECC74 was invoked, which resulted in a 35.37% power reduction compared to the baseline design. The tradeoff for this reduction in power is slightly decreased video quality (i.e., PSNR of 34.32dB for the ECC1511 case and 32.28dB for the ECC74 case, both of which are well above the minimally acceptable 30dB).

### 4.6.3. Video Quality

To evaluate the quality of videos with our proposed method, 100 different videos were simulated with various modes of ECC under 0.1% failure rates. Those videos were downloaded from YouTube-8M [59].

As shown in Figure 33, if no ECC was applied with 0.1% error, PSNR ranged between 32dB to 33dB. With ECC1511 enabled, PSNR improved by approximately 24.90%. If no ECC was utilized with 0.9% error, PSNR ranged between 22dB to 23dB, and improved by approximately 33.04% with ECC1511 enabled. Furthermore, applying ECC74 to the 0.9% error case even further increased PSNR improvement, as Section 4.4.3 determined that ECC74 was better than ECC1511 for higher failure rates.

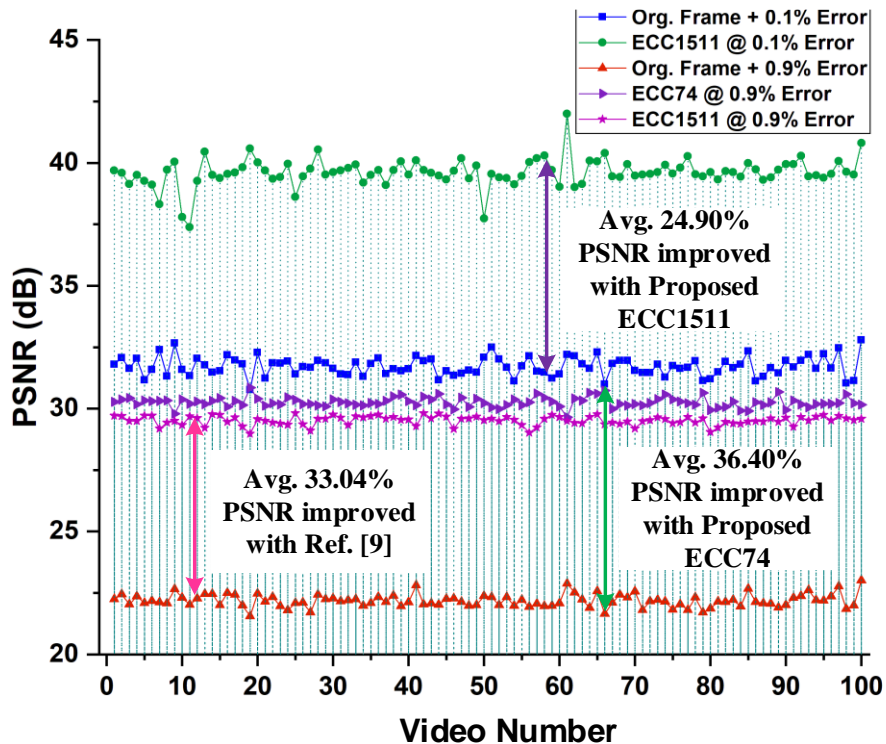


Figure 33. PSNR Values of 100 Videos at 0.1% and 0.9% Failure Rates.

## 5. CONCLUSIONS AND FUTURE WORK

I have conducted research on multiple projects related to memory (SRAM, DRAM) design for video processing and deep learning systems. All the projects include designing novel circuitry, implementing in Cadence, and verifying with HSPICE and python.

In Chapter 2, novel mathematical models for optimizing memory design using nonlinear programs and integer linear programs have been developed. Different memory designs, such as (a) alternative bitcells and transistor sizing technique and (b) hybrid SRAM and DRAM designs, were considered. The results of the numerical studies show that the developed method can significantly reduce the expected mean square error of video storage. It is worthy to emphasize that although the models are developed for video applications, they can be easily adapted to a variety of data-intensive applications, such deep learning systems [11, 51]. Based on the developed modeling framework, future investigations will include adding other constraints, such as performance and power, into the models.

Chapter 3 focused on analyzing the power efficiency, accuracy, and privacy characteristics of differentially private deep learning systems, and presented a memory design for the input data consisting of upsized devices and 8T+6T hybrid bitcells, to achieve power efficiency and accuracy optimization for different privacy levels. It concluded that the memory design that achieves the optimal quality of the input data, can provide the highest prediction accuracy with different privacy levels. To enable the presented design technique, a mean squared error (MSE) based Integer Linear Programs (ILP) model was developed for optimal memory design with different silicon area constraints in differentially private deep learning systems, which significantly saved design time as compared with traditional time-consuming and laborious ASIC design processes. Simulation results demonstrate significant reduction in power



consumption under different silicon area design constraints, with less than 1% degradation in classification accuracy for different privacy levels. Future investigations would include extension of the proposed optimal memory design to deal with activation private data storage in partitioned deep learning systems (e.g., [60]).

In Chapter 4, a flexible power-efficient video memory was presented that can dynamically adjust the strength of error-correction-code (ECC), thereby enabling power-quality trade-off to achieve considerable power savings (up to 35.5%) without a noticeable degradation in video quality. To minimize the implementation overhead, the following two techniques have been developed: (i) a new parity storage scheme that utilizes the bit significance characteristics of video data for both ECC74 and ECC1511, and (ii) an integrated ECC encoder/decoder hardware design to support both ECC74 and ECC1511 that automatically shuts down part or all of the ECC circuitry when ECC74 or No ECC is selected, respectively.

Since parity bit errors caused by memory failures resulted in the ECC decoder incorrectly flipping bits, which caused an increase in video quality degradation, future work will consider hardening specific bits, such as parity bits, to provide better video quality, with the trade-off being increased area overhead. Additionally, other multi-bit error correcting codes, besides ECC74 and ECC1511 used in this work, could also be considered.

## REFERENCES

- [1] Chia-Ping Lin et al., "A 5mW MPEG4 SP encoder with 2D bandwidth-sharing motion estimation for mobile applications," 2006 *IEEE International Solid State Circuits Conference - Digest of Technical Papers*, San Francisco, CA, 2006, pp. 1626-1635, doi: 10.1109/ISSCC.2006.1696217.
- [2] A. P. Chandrakasan, S. Sheng and R. W. Brodersen, "Low-power CMOS digital design," in *IEEE Journal of Solid-State Circuits*, vol. 27, no. 4, pp. 473-484, April 1992, doi: 10.1109/4.126534.
- [3] (2017, Sep.) Cisco Systems, Inc. [Online]. Available: <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html>
- [4] (2012) The Digital Universe in 2020: Big Data, bigger Digital Shadows, and Biggest Growth in the Far East. December 2012. [Online]. Available: <https://www.emc.com/collateral/analystreports/idc-digital-universe-united-states.pdf>
- [5] J. Wang, P. Chang, T. Tang, J. Chen and J. Guo, "Design of Subthreshold SRAMs for Energy-Efficient Quality-Scalable Video Applications," in *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 1, no. 2, pp. 183-192, June 2011, doi: 10.1109/JETCAS.2011.2158345.
- [6] T Liu et al., "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.
- [7] J. Kwon, I Lee, and J Park, "Heterogeneous SRAM Cell Sizing for Low Power H.264 Applications," *IEEE Trans. on Circuits and Systems I*, vol. 99, no. 2, pp. 1-10, Feb. 2012.
- [8] 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.
- [9] N Gong, S Jiang, A Challapalli, S Fernandes, and R Sridhar, "Ultra-Low Voltage Split-data-aware Embedded SRAM for Mobile Video Applications," *IEEE Trans. on Circuits and Systems II*, vol. 59, no. 12, pp. 883-887, 2012.
- [10] A Kazimirsky, A Teman, N Edri, and A Fish, "A 0.65-V, 500-MHz Integrated Dynamic and Static RAM for Error Tolerant Applications," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 9, pp. 2411-2418, Sep. 2017.
- [11] J Edstrom, Y Gong, D Chen, J Wang, and N Gong, "Data-Driven Intelligent Efficient Synaptic Storage for Deep Learning," *IEEE Trans. on Circuits and Systems II*, vol. 64, no. 12, pp. 1412-1416, 2017.
- [12] FreePDK45. [Online]. Available: <http://www.eda.ncsu.edu/wiki/FreePDK45:Contents>.

- [13] A. T. Do, Z. C. Lee, B. Wang, I. Chang, X. Liu and T. T. Kim, "0.2 V 8T SRAM With PVT-Aware Bitline Sensing and Column-Based Data Randomization," in *IEEE Journal of Solid-State Circuits*, vol. 51, no. 6, pp. 1487-1498, June 2016, doi: 10.1109/JSSC.2016.2540799.
- [14] N. Gong et al., "Hybrid-Cell Register Files Design for Improving NBTI Reliability," *Microelectronics Reliability*, vol. 52, no. 9, pp. 1865-1869, 2012.
- [15] J. Liu, B. Jaiyen, R. Veras, and O. Mutlu, "RAIDR: Retention-aware Intelligent DRAM Refresh," in *Proc. 39th Annual International Symposium on Computer Architecture*, 2012, pp. 1–12
- [16] ME Sinangil and AP Chandrakasan, "Application-specific SRAM design using output prediction to reduce bit-line switching activity and statistically gated sense amplifiers for up to  $1.9\times$  lower energy/access," *IEEE J. Solid-State Circuits*, vol. 49, no. 1, 2014.
- [17] A. Chin and A. Klinefelter, "Differential Privacy as a Response to the Reidentification Threat: The Facebook Advertiser Case Study," *North Carolina Law Review*, vol. 90, no. 5, 2012.
- [18] J. Tang, A. Korolova, X. Bai, X. Wang and X. Wang, "Privacy Loss in Apple's Implementation of Differential Privacy on MacOS 10.12," *ArXiv*, 2017.
- [19] Ú. Erlingsson, V. Pihur and A. Korolova, "RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, Scottsdale, 2014.
- [20] M. Barbaro and T. Zeller, "A Face Is Exposed for AOL Searcher No. 4417749," *The New York Times*, New York, 2006.
- [21] D. Jackson, "The Netflix Prize: How a \$1 Million Contest Changed Binge-Watching Forever," *Thrillist.com*, 2017.
- [22] A. Narayanan and V. Shmatikov, "Robust De-anonymization of Large Sparse Datasets," in *IEEE Symposium on Security and Privacy*, Oakland, 2008.
- [23] R. Wang, Y. F. Li, X. Wang, H. Tang and X. Zhou, "Learning Your Identity and Disease from Research Papers: Information Leaks in Genome Wide Association Study," in *Proceedings of the 16th ACM Conference on Computer and Communications Security*, Chicago, 2009.
- [24] W. M. Holt, "Security and Privacy Weaknesses of Neural Networks," *Provo*, 2017.
- [25] (2019). Apple Watch Series. Accessed: May 22, 2019. [Online]. Available: <https://www.apple.com/apple-watch-series-4/health/>

- [26] A. C. Valdez and M. Ziefle, "The users' perspective on the privacy-utility trade-offs in health recommender systems," *Int. J. Human-Comput. Studies*, vol. 121, pp. 108–121, Jan. 2019.
- [27] (Jan. 2019). CarePredict Launches AI-Powered Platform for Seniors Aging at Home, at CES 2019, CarePredict. Accessed: May 22, 2019. [Online]. Available: <https://www.carepredict.com/news/carepredictlaunches-ai-powered-platform-for-seniors-aging-at-home-at-ces-2019/>
- [28] C. Song, T. Ristenpart and V. Shmatikov, "Machine Learning Models that Remember Too Much," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, Dallas, 2017.
- [29] G. Ateniese, L. V. Mancini, A. Spognardi, A. Villani, D. Vitali and G. Felici, "Hacking Smart Machines with Smarter Ones: How to Extract Meaningful Data from Machine Learning Classifiers," *International Journal of Security and Networks*, vol. 10, no. 3, pp. 137-150, 2015.
- [30] R. Shokri, M. Stronati, C. Song and V. Shmatikov, "Membership Inference Attacks Against Machine Learning Models," in *IEEE Symposium on Security and Privacy*, San Jose, 2017.
- [31] C. Dwork and A. Roth, "The Algorithmic Foundations of Differential Privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3-4, pp. 211-407, 2014.
- [32] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," *Proc. IEEE*, vol. 105, no. 12, pp. 2295–2329, Dec. 2017.
- [33] T. Chen et al., "DianNao: A small-footprint high-throughput accelerator for ubiquitous machine-learning," in *Proc. ASPLOS*, Mar. 2014, pp. 269–284.
- [34] Fabio Frustaci, Mahmood Khayatzadeh, David Blaauw, Dennis Sylvester, and Massimo Alioto, 'SRAM for ErrorTolerant Applications With Dynamic Energy-Quality Management in 28nm CMOS', *IEEE Journal of Solidstate circuits*, vol.50, no.5, pp 1310-1312, MAY 2015
- [35] F. Frustaci, D. Blaauw, D. Sylvester and M. Alioto, "Better-than-voltage scaling energy reduction in approximate SRAMs via bit dropping and bit reuse," *2015 25th International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS)*, Salvador, 2015, pp. 132-139, doi: 10.1109/PATMOS.2015.7347598
- [36] J. Hsu, M. Gaboardi, A. Haeberlen, S. Khanna, A. Narayan, B. C. Pierce and A. Roth, "Differential Privacy: An Economic Method for Choosing Epsilon," in *IEEE 27th Computer Security Foundations Symposium*, Vienna, 2014.

- [37] N. Papernot, M. Abadi, Ú. Erlingsson, I. Goodfellow and K. Talwar, "Semi-Supervised Knowledge Transfer for Deep Learning from Private Training Data," in *5th International Conference on Learning Representations*, Toulon, 2017.
- [38] X. Zhang, S. Ji and T. Wang, "Differentially Private Releasing via Deep Generative Model," *ArXiv*, 2018.
- [39] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar and L. Zhang, "Deep Learning with Differential Privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and*, Vienna, 2016.
- [40] A. D. Sarwate and K. Chaudhuri, "Signal Processing and Machine Learning with Differential Privacy," *IEEE Signal Processing Magazine*, pp. 86-94, September 2013.
- [41] V. Sze, Y.-H. Chen, J. Emer, A. Suleiman and Z. Zhang, "Hardware for Machine Learning Challenges and Opportunities," in *IEEE Custom Integrated Circuits Conference*, Austin, 2017.
- [42] Google, "TensorFlow Lite," Google, 2018. [Online]. Available: <https://www.tensorflow.org/lite/>. [Accessed 3 December 2018].
- [43] L. Yang and B. Murmann, "Approximate SRAM for Energy-Efficient, Privacy-Preserving Convolutional Neural Networks," in *IEEE Computer Society Annual Symposium on VLSI*, Bochum, 2017.
- [44] Google, "TensorFlow™," Google, [Online]. Available: <https://www.tensorflow.org/>. [Accessed 11 11 2018].
- [45] Y. LeCun, C. Cortes and C. J. Burges, "THE MNIST DATABASE of handwritten digits," 1998. [Online]. Available: <http://yann.lecun.com/exdb/mnist/>. [Accessed 20 March 2019].
- [46] "Internet of Things (IoT) connected devices installed base worldwide from 2015 to 2025 (in billions)," [Online]. Available: <https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/>. [Accessed August 2019].
- [47] Y. Xu, H. Das, Y. Gong and N. Gong, "On Mathematical Models of Optimal Video Memory Design," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 30, no. 1, pp. 256-266, Jan.
- [48] N. Gong, J. Edstrom, D. Chen and J. Wang, "Data-Pattern Enabled Self-Recovery Multimedia Storage System for Near-Threshold Computing," in *IEEE International Conference on Computer Design (ICCD'16)*, Scottsdale, 2016.
- [49] 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," *IEEE Trans. On Computers*, vol. 65, no. 3, pp. 755-769, 2016.

- [50] S. Mukhopadhyay, H. Mahmoodi and K. Roy, "Modeling of Failure Probability and Statistical Design of SRAM Array for Yield Enhancement in Nanoscaled CMOS," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 24, no. 12, p. 1859–1880, 2005.
- [51] S. Gopalakrishnan, 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 *2016 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, Dresden, 2016.
- [52] H. Xiao, K. Rasul and R. Vollgraf, "Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms," 28 August 2017. [Online]. Available: <https://arxiv.org/abs/1708.07747>. [Accessed 3 April 2019].
- [53] T. Clanuwat, M. Bober-Irizar, A. Kitamoto, A. Lamb, K. Yamamoto and D. Ha, "Deep Learning for Classical Japanese Literature," 3 December 2018. [Online]. Available: <http://www.arxiv.org/pdf/1812.0118.pdf>. [Accessed 10 August 2019].
- [54] D. Chen, J. Edstrom, Y. Gong, P. Gao, L. Yang, M. McCourt, J. Wang and N. Gong, "Viewer-Aware Intelligent Efficient Mobile Video Embedded Memory," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 4, pp. 684-696, 2018.
- [55] F. Frustaci, M. Khayatzadeh, D. Blaauw, D. Sylvester and M. Alioto, "SRAM for Error-Tolerant Applications With Dynamic Energy-Quality Management in 28nm CMOS," *IEEE J. Of Solid-State Circuits*, vol. 50, no. 5, pp. 1310-1323, 2015.
- [56] J. Edstrom, Y. Gong, A. Haidous, B. Humphrey, M. McCourt, Y. Xu, J. Wang and N. Gong, "Content-Adaptive Memory for Viewer-Aware Energy-Quality Scalable Mobile Video Systems," *IEEE Access*, vol. 7, pp. 47479-47493, 2019.
- [57] C. Duan, A. J. Gotterba, M. E. Sinangil and A. P. Chandrakasan, "Energy-Efficient Reconfigurable SRAM: Reducing Read Power Through Data Statistics," *IEEE Journal Of Solid-State Circuitis*, vol. 52, no. 10, pp. 2703-2711, 2017.
- [58] R. W. Hamming, "Error detecting and error correcting codes," *Bell Syst. Tech. J.*, vol. 29, no. 2, pp. 147–160, Apr. 1950.
- [59] " YouTube-8M Dataset.," 2017. [Online]. Available: <https://research.google.com/youtube8m/>.
- [60] J. Wang, J. Zhang, W. Bao, X. Zhu, C. B and P. Yu, "Not Just Privacy: Improving Performance of Private Deep Learning in Mobile Cloud," in *KDD*, 2018.