Abstract—Conventional low-power static random access memories (SRAMs) reduce read energy by decreasing the bit-line swings uniformly across the bit-line columns. This is because the read energy is proportional to the bit-line swings. On the other hand, bit-line swings are limited by the need to avoid decision errors especially in the most significant bits. We propose an information-theoretic approach to determine optimal non-uniform bit-level swings by formulating convex optimization problems. For a given constraint on mean squared error (MSE) of retrieved words, we consider criteria to minimize energy (for low-power SRAMs), maximize speed (for high-speed SRAMs), and minimize energy-delay product. These optimization problems can be interpreted by classical water-filling, ground-flattening and water-filling, and sand-pouring and water-filling, respectively. By leveraging these interpretations, we also propose greedy algorithms to obtain optimized discrete swings. Numerical results show that energy-optimal swing assignment reduces energy consumption by half at a peak signal-to-noise ratio of 30dB for an 8-bit accessed word. The energy savings increase to four times for a 16-bit accessed word.

I. INTRODUCTION

In von Neumann architectures that separate the computing and memory units, it is well known that frequent data access consumes enormous energy. Since data access requires more energy than arithmetic operations [1], it dominates the total energy consumption especially for recent big data applications. Modern information processing systems leverage several levels of local memory hierarchy to reduce the energy cost for data access [2]–[4]. Although dynamic random access memories (DRAMs) consume much more energy for data access than static random access memories (SRAMs), several architectural techniques can minimize DRAM access. Hence, SRAM access energy is the largest part of the total energy consumption in many information processing systems [3]–[7].

Thus, it is critical to reduce the energy consumption of SRAM access. The basic way to reduce the read access energy is to decrease either supply voltages or bit-line (BL) swings, which increases vulnerability to variations and noise. If we reduce supply voltages or BL swings across all BL columns [8], [9], then bit error rates (BERs) of all bit positions increase equally.

In many applications including signal processing and machine learning tasks, however, the impact of bit errors depends on bit position. For example, errors in the most significant bits (MSB) of image pixels degrade overall image quality much more than errors in the least significant bits (LSB). Likewise, an MSB error can cause a catastrophic loss in the inference accuracy of ML applications.

Until now, the following techniques have been proposed to address the different impacts of each bit position:

1) Storing the MSBs in more robust bit cells and the LSBs in less robust cells [10]–[12].
2) Applying higher supply voltage for the MSBs and lower supply voltage for the LSBs [13]–[15].
3) Unequal error protection (UEP) by error control coding (ECC) [16]–[19].
4) LSB dropping (dropping the LSBs at the cost of reduced arithmetic precision) [18]–[20].

The first approach requires modifying the bit cell array at design time, which results in prohibitively high design cost. Also, it is unable to dynamically track the time-varying fidelity requirement and correspondingly minimize energy [19]. The second approach needs different supply voltages for each bit position, which significantly complicates the power routing network. Practical implementations only allow a few supply voltage levels [14], [15]. Fine-grained UEP requires complicated hardware implementations and dynamic change of protection is limited. LSB dropping enables dynamic fidelity control by changing the number of dropped LSBs. In [18], selective ECC was proposed by combining UEP and LSB dropping. In [21], coding techniques applying selective ECC adaptively according to the stored data's magnitude were proposed. Note that LSB dropping and selective ECCs allow two levels of granularity (dropped/undropped or protected/unprotected) for each bit position.

This paper presents an information-theoretic approach to determine the optimal bit-level swing assignments under Gaussian noise assumptions. For a given constraint on mean squared error (MSE) of retrieved word, we formulate convex optimization problems whose objectives are as follows:

C1. Minimize energy for low-power SRAMs,
C2. Minimize maximum delay (i.e., maximize speed) for high-speed SRAMs,
C3. Minimize energy-delay product (EDP).
Solutions to these convex problems yield optimal performance theoretically attainable. We investigate the fundamental trade-offs between physical resources (energy, delay, and EDP) and a fidelity (MSE) constraint in this end-to-end transmission problems.

In addition, we provide generalized water-filling interpretations for our optimal solutions. This follows since accessing a $B$-bit word is equivalent to communicating information through $B$ parallel channels. In classical water-filling, the ground represents the noise levels of parallel channels \[22\], \[23\]. On the other hand, the importance of each bit position determines the ground level in our optimization problems. Each optimization problem has its own interpretation depending on its objective function: water-filling (Criterion 1), ground-flattening and water-filling (Criterion 2), and sand-pouring and water-filling (Criterion 3), respectively. We also observe interesting connections between our problems and variants on water-filling such as constant-power water-filling \[24\], \[25\] and mercury/water-filling \[26\]. Also, we show that the proposed optimization techniques can be extended to a wide range of sources and noise models.

Furthermore, we propose an SRAM circuit architecture to assign non-uniform bit-level swings. The proposed architecture separates the data for each bit position in different SRAM subarrays by interleaving. The proposed architecture enables fine-grained and dynamic control of bit-level swings depending on time-varying fidelity requirements with little circuit complexity overhead. Also, we propose greedy algorithms to optimize swing values drawn from a discrete set due to circuit implementation limitations. Generalized water-filling interpretations and Karush-Kuhn-Tucker (KKT) conditions are leveraged to develop these discrete optimization algorithms.

The rest of this paper is organized as follows. Section II introduces key metrics of energy, delay, and fidelity. Section III formulates the convex optimization problems to determine the optimum bit-level swings and provides generalized water-filling interpretations. Section IV shows that the proposed optimization techniques can be extended to various source and noise models. Section V investigates the SRAM architecture and develops greedy algorithms to optimize discrete swings. Section VI gives numerical results and Section VII concludes.

II. SRAM METRICS FOR RESOURCE AND FIDELITY

The total energy in an SRAM read access is given by

$$E_{\text{total}} = E_{\text{array}} + E_{\text{peri}} + E_{\text{leakage}}$$  \hspace{1cm} (1)

where $E_{\text{array}}$ and $E_{\text{peri}}$ denote the dynamic energy consumption from the SRAM bit cell array and the peripheral circuitry, respectively, and $E_{\text{leakage}}$ represents the energy loss due to leakage.

$E_{\text{array}}$ is the largest component of energy consumption in high-density SRAMs. For normal operating conditions, $E_{\text{array}}$ is more than half of $E_{\text{total}}$ \[9\], \[27\], \[28\] and $E_{\text{array}}$ is given by

$$E_{\text{array}} \propto N_{\text{BL}} N_{\text{WL}} C_{\text{bit}} V_{\text{dd}} \Delta$$  \hspace{1cm} (2)

where $N_{\text{BL}}$ and $N_{\text{WL}}$ are the numbers of bit-lines (BLs) and word-lines (WLs) in a memory bank, respectively. $C_{\text{bit}}$ is the

![Fig. 1. A typical SRAM block where $N_{\text{BL}} = 6$ and $N_{\text{WL}} = 5$.](image)

BL capacitance per bit cell and $V_{\text{dd}}$ is the supply voltage. Also, $\Delta$ denotes the BL voltage swing in read access. As shown in Fig. 1, the voltage swing $\Delta$ is the voltage difference between BL and BL-bar (BLB). This voltage difference occurs because either BL or BLB is discharged according to the stored bit. A sense amplifier detects which line (BL or BLB) has the higher voltage and decides whether the corresponding bit cell stores 1 or 0.

The swing $\Delta$ can be controlled by changing the WL pulse-width (i.e., WL activation time) $T_{\text{WL}}$ since

$$\Delta = \frac{I_c}{N_{\text{WL}} C_{\text{bit}}} \cdot T_{\text{WL}}$$  \hspace{1cm} (3)

where $I_c$ is the discharge current of BL corresponding to the accessed bit cell \[9\]. From (2) and (3), we can observe that $E_{\text{array}}$ is directly proportional to $T_{\text{WL}}$. Also, $T_{\text{WL}}$ has a direct impact on the read access time \[9\], \[29\].

Since larger voltage swing $\Delta$ improves noise margin, there are trade-off relations between reliability, energy, and delay. These relations will be explained in the following subsections.

A. Resource Metrics for Accessing $B$-bit Word: Energy, Delay, and EDP

We define resource metrics for energy, delay, and EDP for accessing a $B$-bit word. First, read energy can be defined as follows.

**Definition 1:** The read energy to access a $B$-bit word is

$$E(\Delta) = \sum_{b=0}^{B-1} \Delta_b = 1^T \Delta$$  \hspace{1cm} (4)

where 1 denotes the all-one vector and the superscript $^T$ denotes transpose. Note that $\Delta = (\Delta_0, \ldots, \Delta_{B-1})$ where $\Delta_b$ denotes the swing for the $b$th bit position in a $B$-bit word.

**Definition 2:** The maximum swing corresponding to a $B$-bit word is given by

$$\rho = \max(\Delta) = \max \{\Delta_0, \ldots, \Delta_{B-1}\}.$$  \hspace{1cm} (5)
If we allot non-uniform swings for each bit position, the access time for a B-bit word depends on $T_{\text{max}}$, which is the pulse-width corresponding to the maximum swing $\rho$ because of (3). Hence, the maximum swing $\rho$ is a proper metric for read speed performance and should be minimized to maximize read speed.

The EDP is widely used to consider both energy and speed [30], [31]. We define the EDP for accessing a B-bit word based on Definitions 1 and 2.

Definition 3: The EDP to access a B-bit word is

$$
\text{EDP}(\Delta) = E(\Delta) \cdot \rho = 1^T \Delta \cdot \rho. \tag{6}
$$

B. Fidelity Metric for Accessing B-bit Word: MSE

We will define a fidelity metric for accessing a B-bit word. Suppose that a B-bit word $x = (x_0, \ldots, x_{B-1})$ is stored in SRAM cells, where $x_0$ and $x_{B-1}$ are the LSB and MSB, respectively. Note that $x$ can be represented by

$$
x = \sum_{b=0}^{B-1} 2^b x_b \tag{7}
$$

where $x_b \in \{0, 1\}$ and $x \in [0, 2^B - 1]$ (for integers $i$ and $j$ such that $i \leq j$, $[i, j] = \{i, i+1, \ldots, j\}$). Also, $\hat{x} = (\hat{x}_0, \ldots, \hat{x}_{B-1})$ denotes the retrieved B-bit word. A decision error flips the original bit $x_b$ as follows:

$$
\hat{x}_b = x_b \oplus \epsilon_b \tag{8}
$$

where $\oplus$ denotes XOR operator and $\epsilon_b = 1$ denotes a bit error in $b$th bit position. The decimal representation of the retrieved word is $\hat{x} = \sum_{b=0}^{B-1} 2^b \hat{x}_b$. The decimal error $e$ is given by

$$
e = \hat{x} - x = \sum_{b=0}^{B-1} 2^b \epsilon_b \tag{9}
$$

where $\epsilon_b = \hat{x}_b - x_b \in \{-1, 0, 1\}$.

Remark 4: The decimal error $e = (\epsilon_0, \ldots, \epsilon_{B-1})$ depends on $x_b$ as well as $\epsilon_b$. Suppose that $e = (1, 0, 0, 1)$. If $x = (1, 0, 0, 1) = 9$, then $\hat{x} = (0, 0, 0, 0) = 0$, i.e., $e = (-1, 0, 0, -1) = -9$. If $x = (0, 1, 1, 0) = 6$, then $\hat{x} = (1, 1, 1, 1) = 15$ and $e = (1, 0, 0, 1) = 9$.

Since major noise sources of SRAMs are well modeled as Gaussian distributions [32]-[35], the error probability of the $b$th bit position is given by

$$p_b = \Pr (\epsilon_b = 1) = Q\left(\frac{\Delta_b}{\sigma}\right) \tag{10}
$$

where $\Delta_b$ and $\sigma^2$ denote the swing of $b$th bit position and the noise variance in the corresponding BL, respectively. Note that $Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{t^2}{2}\right) \, dt$. By increasing $\Delta_b$ in (10), we can reduce $p_b$. However, larger $\Delta_b$ implies more energy consumption and slower speed (see Definitions 1 and 2).

To measure memory retrieval reliability, bit error probability (10) is not appropriate for many applications, since it does not distinguish the differential impact of MSB and LSB errors. Hence, we use the MSE as a fidelity metric.

Definition 5: The MSE of $x$ is given by

$$\text{MSE}(x) = E[(\hat{x} - x)^2] = E[e^2]. \tag{11}
$$

### Table I: Comparison of Resource and Fidelity Metrics for Single-Bit and B-bit Word

|                | Single bit | B-bit word | Remarks |
|----------------|------------|------------|---------|
| Energy         | $\Delta$   | $E(\Delta) = 1^T \Delta$ | Definition 1 |
| Delay          | $\rho = \max(\Delta)$ | $E(\Delta) \cdot \rho$ | Definition 2 |
| EDP            | $\Delta^2$ | $E(\Delta) = E(\Delta) \cdot \rho$ | Definition 3 |
| Fidelity       | $\rho = Q\left(\frac{\Delta}{\sigma}\right) \sum_{b=0}^{B-1} 4^b Q\left(\frac{\Delta_b}{\sigma}\right)$ | Lemma 6 |

Lemma 6: For a uniformly distributed $x$, MSE($x$) is given by

$$
\text{MSE}(x) = \text{MSE}(\Delta) = \sum_{b=0}^{B-1} 4^b Q\left(\frac{\Delta_b}{\sigma}\right). \tag{12}
$$

Proof: If $x$ is uniformly distributed, $x_b$s are independent and identically distributed (i.i.d) and follow the Bernoulli distribution Ber (1/2). The MSE of $x$ is given by

$$
\text{MSE}(x) = E\left[\sum_{b=0}^{B-1} 4^b e_b^2\right] \tag{13}
$$

$$
= \sum_{b=0}^{B-1} 4^b E[e_b^2] = \sum_{b=0}^{B-1} 4^b p_b \tag{14}
$$

where (14) follows from $E[e_b^2] = E[e_b] = p_b$ and $E[e_b e_j] = 0$ since $e_b$s are independent and $E[e_b] = 0$ for $x_b \sim \text{Ber}(1/2)$ [10]. In addition, (15) follows from (10). Because MSE($x$) is a function of $\Delta$, we set $\text{MSE}(x) = \text{MSE}(\Delta)$.

Note that MSE($x$) is the nonnegative weighted sum of bit error probabilities. The weight of $4^b$ represents differential importance of each bit position. We show that MSE($x$) is convex.

Lemma 7: MSE($\Delta$) is a convex function of $\Delta$.

Proof: $Q(x)$ is convex for $x \geq 0$ because

$$
\frac{d^2 Q(x)}{dx^2} = \frac{x}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) \geq 0. \tag{16}
$$

Since $\Delta_b \geq 0$ and MSE($\Delta$) is the nonnegative weighted sum of $Q(\Delta_b/\sigma)$, MSE($\Delta$) is convex.

A signed number $x$ can be represented by

$$
x = -x_{B-1} \cdot 2^{B-1} + \sum_{b=0}^{B-2} 2^b x_b \tag{17}
$$

whose MSE($x$) is the same as (12).

Table I compares key resource and fidelity metrics for single-bit and B-bit word accesses.

III. OPTIMAL BIT-LEVEL SWINGS

We formulate convex optimization problems to determine the optimum swings. For a given constraint on MSE, we attempt to (1) minimize energy for low-power SRAMs, (2) maximize speed for high-speed SRAMs, and (3) minimize EDP. Also, we provide generalized water-filling interpretations of these optimization problems based on KKT conditions.
The optimal solution (19) is derived from L1 and the corresponding KKT conditions. The details of the proof are given in Appendix A.

The optimal solution (19) can be interpreted as classical water-filling or reverse water-filling as shown in Fig. 2. Each bit position can be regarded as an individual channel among B parallel channels. In the water-filling interpretation (see Fig. 2(a)), the ground levels depend on the importance of bit positions. We flood the bins to the water level of $\log \nu$. Since the MSB has the lowest ground level and the LSB has the highest ground level, larger swings are assigned to more significant bit positions. For a bit position $b$ such that $\nu > \frac{\sqrt{2\pi\sigma}}{2}$, we can readily obtain the following equation (see Appendix A):

$$ \log \nu = \log \frac{\sqrt{2\pi\sigma}}{4b} + \Delta_b^2 \frac{2\sigma^2}{\pi} $$

where $\log \nu$, $\log \frac{\sqrt{2\pi\sigma}}{2b}$, and $\Delta_b^2 \frac{2\sigma^2}{\pi}$ represent the water level, the ground level, and the water depth, respectively. The water level $\log \nu$ depends on $\nu$ in (18).

Fig. 2(b) illustrates a reverse water-filling interpretation of (19). For a bit position $b$ such that $\frac{1}{\nu} < \frac{4b}{\sqrt{2\pi\sigma}}$, by modifying (21), we can readily obtain

$$ \log \frac{4b}{\sqrt{2\pi\sigma}} = \log \frac{1}{\nu} + \Delta_b^2 \frac{2\sigma^2}{\pi} $$

where $\log \frac{4b}{\sqrt{2\pi\sigma}}$ and $\log \frac{1}{\nu}$ denote the reverse ground level and the reverse water level, respectively. The reverse ground level implies the importance of each bit position. We allocate positive swings only for bit positions whose reverse ground levels are greater than the reverse water level.

Although we are dealing with the weighted bit error probabilities $4bQ\left(\frac{\Delta_b}{\sigma}\right)$ rather than capacities (for water-filling) or rate distortion functions (for reverse water-filling), we still obtain water-filling and reverse water-filling interpretations.

Remark 9 (LSB dropping and constant-power water-filling): Constant-power water-filling activates the same subset of parallel channels but with a constant power allocation [24], [25]. The constant-power water-filling in communication theory is equivalent to the LSB dropping in circuit theory [18]–[20] since LSB dropping allocates uniform swings for undropped bit positions.

B. Maximize Speed

For high-speed SRAMs, we should maximize the speed of read access for a given constraint on MSE. The maximum speed can be achieved by minimizing $\rho$ of (5) since $\rho$ is proportional to the maximum pulse-width $T_{\max}$.
By introducing an additional variable \( \xi \), we can reformulate (23) as

\[
\begin{align*}
\text{minimize} & \quad \Delta \xi \\
\text{subject to} & \quad \sum_{b=0}^{B-1} 4^b Q \left( \frac{\Delta_b}{\sigma} \right) \leq V \\
& \quad 0 \leq \Delta_b \leq \xi, \quad b = 0, \ldots, B - 1
\end{align*}
\]

This reformulated optimization problem is also convex. From KKT conditions, we show that \( \xi \) is the same as \( \rho \) (see Appendix B).

**Theorem 10:** The optimal swing \( \Delta^* \) of (23) is given by

\[
\Delta_b^* = \rho = \xi = \sigma \sqrt{2 \log \left( \frac{4^B - 1}{3\sqrt{2\pi} \sigma} \cdot \nu \right)}
\]

for all \( b \in [0, B - 1] \). Note that \( \nu \) is a dual variable.

**Proof:** We define the Lagrangian \( L_2(\Delta, \xi, \nu, \lambda, \eta) \) associated with problem (24) as

\[
L_2(\Delta, \xi, \nu, \lambda, \eta) = \xi + \nu \left( \sum_{b=0}^{B-1} 4^b Q \left( \frac{\Delta_b}{\sigma} \right) - V \right) - \sum_{b=0}^{B-1} \lambda_b \Delta_b - \sum_{b=0}^{B-1} \eta_b (\Delta_b - \xi)
\]

where \( \nu, \lambda = (\lambda_0, \ldots, \lambda_{B-1}) \) and \( \eta = (\eta_0, \ldots, \eta_{B-1}) \) are dual variables. The optimal solution (25) can be derived from \( L_2 \) and corresponding KKT conditions. The details of the proof are given in Appendix B.

The optimal solution (25) can be interpreted by ground-flattening and water-filling. For any \( b \in [0, B - 1] \), we derive the following equation (see Appendix B).

\[
\log \nu = \log \frac{\sqrt{2\pi} \sigma}{4^b} + \log \eta_b + \frac{\Delta_b^2}{2\sigma^2}
\]

where \( \log \nu, \log \frac{\sqrt{2\pi} \sigma}{4^b}, \log \eta_b, \) and \( \frac{\Delta_b^2}{2\sigma^2} \) represent the water level, the ground level, the ground-flattening term, and the water depth, respectively. Compared with (21), we observe that (27) has an additional ground-flattening term \( \log \eta_b \). By solving KKT conditions, we show that

\[
\log \eta_b = \log \frac{3}{4^B - 1} \cdot 4^b < 0.
\]

Hence, the flattened ground level (i.e., the sum of the ground level and the ground flattening term) is given by

\[
\log \frac{\sqrt{2\pi} \sigma}{4^b} + \log \eta_b = \log \frac{3\sqrt{2\pi} \sigma}{4^B - 1}.
\]

Since the unequal ground levels are flattened by the flattening terms, the water depths of all bit positions are identical after water-filling (see Fig. 3(b)).
minimize $\Delta \cdot \xi$
subject to $\sum_{b=0}^{B-1} 4^b Q \left( \frac{\Delta_b}{\sigma} \right) \leq V$
$0 \leq \Delta_b \leq \xi, \quad b = 0, \ldots, B - 1$

which is derived by taking into account (6) and (24). We show that $\xi$ is equal to $\rho$ (see Appendix C).

**Theorem 13:** The optimal swing $\Delta^*$ of (32) is given by

$$\Delta^*_b = \begin{cases} 0, & \text{if } \log \frac{\nu}{\rho} \leq \log \frac{\sqrt{2\pi} \sigma}{4^b}, \\ \rho, & \text{if } \log \frac{\nu}{\rho} \geq \log \frac{\sqrt{2\pi} \sigma}{4^b} + \frac{\Delta_b^2}{2\sigma^2}, \end{cases}$$

where $\nu$ is a dual variable.

**Proof:** We define the Lagrangian $L_3(\Delta, \xi, \nu, \lambda, \eta)$ associated with problem (32) as

$$L_3(\Delta, \xi, \nu, \lambda, \eta) = 1^T \Delta \cdot \xi + \nu \left( \sum_{b=0}^{B-1} 4^b Q \left( \frac{\Delta_b}{\sigma} \right) \right) - \nu \sum_{b=0}^{B-1} \lambda_b \Delta_b$$

$$+ \sum_{b=0}^{B-1} \eta_b (\Delta_b - \xi)$$

where $\nu, \lambda = (\lambda_0, \ldots, \lambda_{B-1})$, and $\eta = (\eta_0, \ldots, \eta_{B-1})$ are dual variables. The optimal solution (33) can be derived from $L_3$ and corresponding KKT conditions. The details of the proof are given in Appendix C.

The optimal solution of (33) can be interpreted by sand-pouring and water-filling as shown in Fig. 5(a). For $\log \frac{\nu}{\rho} > \log \frac{\sqrt{2\pi} \sigma}{4^b}$, we derive the following equation (see Appendix C):

$$\log \frac{\nu}{\rho} = \log \frac{\sqrt{2\pi} \sigma}{4^b} + \log \left( 1 + \frac{\eta_b}{\rho} \right) + \frac{\Delta_b^2}{2\sigma^2}$$

where $\log \frac{\nu}{\rho}, \log \frac{\sqrt{2\pi} \sigma}{4^b}, \log \left( 1 + \frac{\eta_b}{\rho} \right)$, and $\frac{\Delta_b^2}{2\sigma^2}$ represent the water level, the ground level, the sand depth, and the water depth, respectively. Pouring sand suppresses the maximum water depth (i.e., the maximum swing) and water-filling allocates swings by taking into account energy efficiency.

The following corollary shows the relation between the sand depth and other metrics.

**Corollary 14:** The sand depth $s_b$ is given by

$$s_b = \log \left( 1 + \frac{\eta_b}{\rho} \right)$$

where

$$\eta_b = \begin{cases} 0, & \text{if } 0 \leq \Delta_b < \rho, \\ > 0, & \text{if } \Delta_b = \rho. \end{cases}$$
Hence, \( s_b = 0 \) for \( 0 \leq \Delta_b < \rho \) and \( s_b > 0 \) for \( \Delta_b = \rho \). Also, the amount of sand is given by

\[
\sum_{b=0}^{B-1} \exp(s_b) = \frac{E(\Delta)}{\rho} + B. \tag{38}
\]

**Proof:** See Appendix C. \( \blacksquare \)

We observe that the amount of sand depends on the energy and the maximum swing.

Suppose that sand is poured in only the MSB position, i.e., \( \Delta_{B-1} = \rho \) and \( \Delta_b < \rho \) for \( b \in [0, B - 2] \). Then,

\[
\eta_{B-1} = \sum_{b=0}^{B-1} \eta_b = \sum_{b=0}^{B-1} \Delta_b = E(\Delta) \tag{39}
\]

which follows from \( 37, 83 \) (in Appendix C), and Definition 1. Hence,

\[
s_{B-1} = \log \left( 1 + \frac{E(\Delta)}{\rho} \right) = \log \left( 1 + \frac{B}{\text{PAPR}(\Delta)} \right) \tag{40}
\]

where the peak-to-average ratio (PAPR) of swings is given by

\[
\text{PAPR}(\Delta) = \frac{\rho}{\exp(s_{B-1}) - 1}. \tag{41}
\]

We also note that (40) takes a similar form as the Gaussian channel’s capacity. By (40) and (41), we obtain

\[
\text{PAPR}(\Delta) = \frac{B}{\exp(s_{B-1}) - 1} \tag{42}
\]

which shows that more sand reduces the PAPR of swings.

Fig. 4 illustrates the ground-flattening and reverse water-filling interpretations. From (35), we can obtain

\[
\log \left( \frac{4^b}{\sqrt{2\pi}\sigma} \right) + \log \frac{\rho}{\rho + \eta_b} = \log \frac{\rho}{\nu} + \frac{\Delta^2_b}{2\sigma^2} \tag{43}
\]

where the negative flattening term \( \log \frac{\rho}{\rho + \eta_b} \) suppresses the maximum swing and reverse water-filling up to the reverse water level \( \log \frac{\rho}{\nu} \) optimizes energy efficiency.

**Remark 15 (Sand-pouring and mercury-filling):** Sand-pouring and water-filling has a connection to mercury/water-filling \([26]\) because both are explained by two-level filling. In the mercury/water-filling case, the mercury is poured before water-filling to fill the gap between an ideal Gaussian signal and practical signal constellations, hence, each mercury depth depends only on the corresponding signal constellation. On the other hand, sand-pouring depends on the ground level and sand depths are correlated with each other since sand-pouring attempts to flatten the ground. Also, the amount of poured sand depends on water-filling as shown in Corollary 14 whereas the amount of mercury is not related to water-filling.

**Remark 16 (Ground-flattening and Sand-pouring):** The names of ground-flattening and sand-pouring come from analogies with hydrodynamics. In hydrodynamics, flattening ground levels increases the flow speed by reducing wetted perimeter \([1, 36]\). In our optimization problems, ground-flattening terms in (27) maximize the read speed by achieving perfectly even ground levels. The sand-pouring of (35) limits the speed performance degradation by partially flattening the ground levels.

Table II summarizes water-filling and reverse water-filling interpretations for our optimization problems. Notice the duality between ground-flattening and sand-pouring.

### IV. Non-uniform Sources and Non-Gaussian Noises

In this section, we study how to extend our optimization problems to non-uniformly distributed sources and to non-Gaussian noise models.

**A. Non-uniform Sources**

In Lemma 6, we considered the MSE of a uniformly distributed source. For a non-uniformly distributed source \( x = \sum_{b=0}^{B-1} x_b \) of \( 7 \), the MSE is derived in the following proposition.

**Proposition 17:** The MSE of \( x \) is given by

\[
\text{MSE}(x) = \sum_{b=0}^{B-1} 4^b p_b + 2 \sum_{b=1}^{B-1} \sum_{b'=0}^{b-1} 2^{b+b'} p_bp_{b'} \phi(b, b') \tag{44}
\]

where \( \phi(b, b') = \Pr(x_b = x_{b'}) - \Pr(x_b \neq x_{b'}) \), \( p_b = Q \left( \frac{\Delta_b}{\sigma} \right) \), and \( p'_b = Q \left( \frac{\Delta'_{b'}}{\sigma} \right) \).

**Proof:** From \([13]\), the MSE of \( x \) is given by

\[
\text{MSE}(x) = \mathbb{E} \left[ \sum_{b=0}^{B-1} 2^b e_b \right]^2 \tag{45}
\]

\[
= \sum_{b=0}^{B-1} 4^b p_b + 2 \sum_{b=1}^{B-1} \sum_{b'=0}^{b-1} 2^{b+b'} \mathbb{E} [e_be_{b'}] \tag{46}
\]

where \( \mathbb{E} [e_be_{b'}] \) for \( b \neq b' \) is given by

\[
\mathbb{E} [e_be_{b'}] = \sum_{x, \hat{x}} p(x)p(\hat{x} | x)e_be_{b'} \tag{47}
\]

\[
= p_0 p_b \left\{ \Pr(x_b = b_{b'}) - \Pr(x_b \neq b_{b'}) \right\} \tag{48}
\]

\[
= p_0 p_b \phi(b, b'). \tag{49}
\]

If \( x \) is a uniformly distributed, \( \phi(b, b') = 0 \) because \( \Pr(x_b) = \frac{1}{2} \) for any \( b \in [0, B - 1] \).

Note that (44) is not convex since the \( p_bp_{b'} \) values are not convex and \( \phi(b, b') \) can be negative. Fortunately, (44) can be approximated to (12) because the second term in the right side of (44) is much smaller than the first term as shown in the following claim.

\( \star \)

\( \star \)

\( \star \)
Claim 18: If \( p_0 \ll \frac{1}{2} \), then \((44)\) can be approximated as \((12)\).  

Proof: We can rewrite \((44)\) as follows:

\[
\text{MSE}(x) = p_0 + \sum_{b=1}^{B-1} (4^b + c_b)p_b \tag{50}
\]

where \( c_b = 2^{b+1} \sum_{b'=0}^{b-1} 2^{b'} p_{b'} \phi(b, b') \). Hence,

\[
|c_b| \leq 2^{b+1} \sum_{b'=0}^{b-1} 2^{b'} p_{b'} |\phi(b, b')| \tag{51}
\]

\[
\leq 2^{b+1} \sum_{b'=0}^{b-1} 2^{b'} p_{b'} \tag{52}
\]

\[
\leq 2^{b+1} p_0 \sum_{b'=0}^{b-1} 2^{b'} \tag{53}
\]

\[
= 2^{b+1}(2^b - 1)p_0 \tag{54}
\]

where \((52)\) follows from \(|\phi(b, b')| \leq 1\). Also, \((53)\) follows from the fact that \( p_0 \geq p_b \) for \( b \in [1, B-1] \) in our optimization problems. If \( 4^b \gg 2^{b+1}(2^b - 1)p_0 \) for every \( b \in [1, B-1] \), then we can neglect the MSE difference between a uniformly distributed source and non-uniformly distributed sources, which is satisfied by the condition \( p_0 \ll \frac{1}{2} \).

We observe that \((44)\) is very close to \((12)\) in many cases even if \( p_0 \approx \frac{1}{2} \) (see Table III in Section VI). The reason is that the second term of \((44)\) cancels out due to sign changes of \( \phi(b, b') \).

B. Non-Gaussian Noise Models

Although SRAM noise is well-modeled as a Gaussian distribution, the proposed optimization problems can be extended to non-Gaussian noise models. We show that the convexity of proposed optimization problems is maintained if the noise is unimodal and symmetric with zero mean.

Claim 19: If the noise is modeled as a unimodal and symmetric distribution with zero mean, then MSE(\(\Delta\)) is convex.

Proof: Suppose that the noise distribution is \( f(t) \), which is a unimodal and symmetric distribution with zero mean. Then, the bit error probability is given by \( p_b = \int_{\Delta_b}^{\infty} f(t) dt \). Note that

\[
\frac{d^2 p_b}{d\Delta_b^2} = -\frac{df(\Delta_b)}{d\Delta_b} \geq 0. \tag{55}
\]

which follows from \( \frac{df(\Delta_b)}{d\Delta_b} \leq 0 \) for \( \Delta_b \geq 0 \). Since the MSE is the nonnegative weighted sum of bit error probabilities, the MSE is also convex.

Hence, the optimization problems to minimize energy, delay, and EDP for a given constraint on MSE are convex if the noise distribution is unimodal, symmetric, and has zero mean.

V. ARCHITECTURE AND DISCRETE SWINGS

In the previous section, we determined the optimized swings assuming that any real value can be assigned to bit-level swings. However, current SRAM architectures and circuits do not support fine-grained bit-level swing assignments. In this section, we propose an SRAM architecture to enable bit-level swing control. Also, we provide algorithms to optimize discrete-valued swings rather than continuous-valued swings.

A. Proposed Architecture

In [9], an SRAM architecture that allocates different swings for each memory instance (array or sub-array) was introduced. The fine-grained swings were achieved by WL pulse-width control with little overhead. This architecture attempts to compensate for the impact of spatial variations by applying different pulse-widths to each sub-array.

By tweaking the architecture of [9], we propose an architecture that controls bit-level swings in an efficient manner. We can separate the data for each bit position in different sub-arrays by interleaving (see Fig. 6). Note that interleaving is already used in most SRAMs for soft-error immunity [37]. [38]. Hence, our architecture does not incur additional overhead, compared to the architecture in [9].

The proposed architecture enables fine-grained bit-level swing control by adjusting pulse-width for each sub-arrays. In addition, dynamic swing control depending on the time-varying fidelity requirement can be achieved by pulse-width control in Fig. 6.

Since pulse-width control is usually implemented by cascaded logic gates [9], the swing granularity depends on logic gates response time, which is a finite value. Hence, we present optimization algorithms for discrete swings in the following subsection.

B. Optimization of Discrete Swings: Discrete Water-filling

By leveraging graphical interpretations from Section III, we propose optimization algorithms for discrete swings. For Criterion 1 (minimize energy) and Criterion 2 (maximize speed), our algorithm approximates the Levin–Campello algorithm [39]–[41]. The optimization problem of Criterion 3 (minimize EDP) cannot be solved by the Levin–Campello algorithm, hence we develop an algorithm based on sand-pouring and water-filling interpretation and its KKT conditions.

Suppose that \( \beta \) is the granularity in swings. Our discrete water-filling algorithm (Algorithm 1) attempts to obtain the discrete swings minimizing energy or maximizing speed by a greedy approach. The basic idea is to fill the water from the bit position whose temporal water level is the lowest.
We can consider choosing $b$ which is equivalent to line 4 of Algorithm 1.

As Discrete water-filling for (18) and (23) Algorithm 1

ground level and sand depth. We increase level of each bit position is the sum of the corresponding
the lowest sand level as shown in line 5. Note that the sand
cannot be handled by the Levin–Campello algorithm. By
Campello algorithm by replacing line 4 in Algorithm 1 with
$g_b$ set the ground level as $b$ for

\[ \beta \ni \frac{\Delta_b}{\sigma} \]

Lemma 6

As $b \rightarrow 0$, (57) converges to

\[ \beta \cdot 4^b \cdot \frac{\partial Q}{\partial \Delta_b} = -\beta \cdot 4^b \sqrt{2\pi\sigma} \exp\left(-\frac{0^2}{2\sigma^2}\right). \]  

We can consider choosing $b$ that minimizes (58) as follows:

\[ b = \arg \min \left\{ -4^b \sqrt{2\pi\sigma} \exp\left(-\frac{\Delta_b^2}{2\sigma^2}\right) \right\} \]

\[ = \arg \min \left\{ \log 4^b + \Delta_b^2 \frac{\sigma}{2\sigma^2} \right\}, \]

which is equivalent to line 3 of Algorithm 1.

Numerical results in Section VI show that the discrete swings obtained by Algorithm 1 are almost identical to the solutions by the Levin–Campello algorithm.

We present an algorithm to obtain discrete swings to minimize EDP in Algorithm 2. The Levin-Campello algorithm cannot solve this problem since the $\rho = \max \{\Delta\}$ in EDP cannot be handled by the Levin-Campello algorithm. By leveraging the sand-pouring and water-filling interpretation of Fig. 5 and KKT conditions, Algorithm 2 attempts to pour sand and fill water judiciously.

At each iteration, Algorithm 2 first pours more sand from the lowest sand level as shown in line 5. Note that the sand level of each bit position is the sum of the corresponding ground level and sand depth. We increase $\eta_b$ by $b$ in line 6 and $\Delta_b$ by $\beta$ in line 11 at each iteration to satisfy the optimal condition $\sum \eta_b = \sum \Delta_b$ (see (83) in Appendix C).

After increasing $\eta_b$, the sand depth $s_b$ of each bit position is calculated by Corollary 14 which indicates the increased amount of sand. Afterwards, water is filled from the bit position whose water level is the lowest. Note that the sand depth $s_b$ affects the water level unlike Algorithm 1 (Compare line 4 of Algorithm 1 and line 10 of Algorithm 2).

Numerical results in Section VI show that the EDP loss due to discrete swings of Algorithm 2 is negligible for moderate granularity $\beta$.

VI. NUMERICAL RESULTS

We evaluate the solutions of the three optimization problems for both continuous and discrete swings. Note that the solution of maximizing speed is equivalent to the conventional uniform swing as noted in Remark 11.

Fig. 7 compares the read energy consumption $E(\Delta)$ as in Definition 1 for a given constraint of peak signal-to-noise ratio (PSNR). The PSNR depends on the MSE as

\[ \text{PSNR} = 10 \log_{10} \left( \frac{(2B-1)^2}{\text{MSE}(\Delta)} \right). \]

At PSNR = 30dB, the optimal solution of (18) (i.e., minimizing energy) reduces the energy consumption by half for $B = 8$, compared to uniform swing (i.e., maximizing speed). For $B = 16$, the energy consumption of energy-optimal swing will be only quarter, compared to the uniform swing. Note that energy consumption of EDP-optimal swing is slightly worse than that of energy-optimal swing.

Fig. 8 compares the maximum delay $\rho$ as in Definition 2 for a given PSNR. The conventional uniform swing minimizes the maximum delay, hence it is the speed-optimal solution. The swings minimizing energy achieve significant energy savings at the cost of speed (e.g., the maximum delay increase of 20% at PSNR = 30dB). The EDP-optimal swings increase only 8% of maximum delay at PSNR = 30dB.

Fig. 9 compares the EDP for a given PSNR. As formulated, the swings minimizing EDP show the best results. The EDP can be reduced by 45% for $B = 8$ at PSNR = 30dB. The
EDP improvement will be much more for $B = 16$, e.g., 75% EDP saving at PSNR = 30dB. Note that slight loss of speed performance can result in significant energy and EDP savings.

Fig. 10 shows optimal solutions to (a) minimize energy, (b) minimize maximum delay, and (c) minimize EDP. As shown in Fig. 10(a), we should allocate larger swings for more significant bits. Also, we observe that the swings for several LSBs can be zero depending on PSNR, e.g., $\Delta_0 = \Delta_1 = \Delta_2 = 0$ at PSNR = 30dB, a refined kind of LSB dropping. These numerical solutions confirm Theorem 8 and its water-filling interpretation in Fig. 2. Fig. 10(b) shows the solutions minimizing maximum delay. As we showed in Theorem 10, uniform swings minimize the maximum delay. The optimized swings in Fig. 10(c) minimize the EDP. Although the EDP-optimal swings are similar to the energy-optimal swings, we observe that $\Delta_6 = \Delta_7 = \rho$ at PSNR = 30dB. It is because these two bit positions are filled with sand to suppress the maximum delay as shown in Theorem 13 and its graphical interpretation in Fig. 5.

Table III compares the PSNRs for uniformly distributed source to real image data (non-uniformly distributed sources) from [42]. Although $p_0 = \frac{1}{2}$ at PSNR = 20dB (see Fig. 11(a) and (c)), we can observe that their PSNRs are almost the same as the PSNRs of uniformly distributed sources as discussed in Section IV-A.

Fig. 11 shows that the energy penalty due to discrete swings obtained by our Algorithm 1 is almost the same as the Levin–Campello algorithm as explained in Corollary 20. We can observe that the energy penalty due to discrete swings is smaller for larger $B$.

Fig. 12 compares the EDP by optimal swings of Theorem 13 and discrete swings by Algorithm 2. By comparing Fig. 11 to Fig. 12, we observe that the EDP is more sensitive to $\beta$ than the energy. The reason is that the EDP is perturbed by the discretization of $\rho$ as well as the discretization of energy. Nonetheless, the EDP penalty at PSNR = 30dB is very little for moderate granularity such as $\beta = 1$. Since the Levin–Campello algorithm cannot solve the EDP optimization problem, it is absent in Fig. 12.

VII. CONCLUSION

SRAM is a critical component for information processing systems. In this paper, we addressed the optimal bit-level swings of SRAMs for applications with fidelity dependent on bit position. We formulated convex optimization problems to determine the optimal swings for the objective functions of energy, maximum delay, and EDP. The optimized bit-level swings can achieve significant energy (50% for 8-bit word and 75% for 16-bit word) and EDP (45% for 8-bit word and 75% for 16-bit word) savings at PSNR of 30dB compared to the conventional uniform swings.

By treating each bit position as an individual channel, we cast bit-level swing optimization problems as generalizations...
TABLE III
COMPARISON OF PSNRs [dB] OF UNFORMLY DISTRIBUTED SOURCES AND REAL IMAGE DATA

| PSNR of uniform source | PSNR of Airport | PSNR of Fishing Boat | PSNR of Man |
|------------------------|-----------------|----------------------|-------------|
| 20                     | 19.99           | 20.05                | 20.19       |
| 24                     | 24.05           | 24.01                | 24.10       |
| 28                     | 28.04           | 28.01                | 28.06       |
| 32                     | 32.00           | 32.03                | 32.04       |
| 36                     | 36.00           | 36.00                | 36.03       |
| 40                     | 40.01           | 39.95                | 40.02       |

The KKT conditions of (18) are as follows:

\[
\sum_{b=0}^{B-1} 4^b Q \left( \frac{\Delta_b}{\sigma} \right) \leq \mathcal{V}, \quad \nu \geq 0, \quad (61)
\]

\[
\nu \cdot \left\{ \sum_{b=0}^{B-1} 4^b Q \left( \frac{\Delta_b}{\sigma} \right) - \mathcal{V} \right\} = 0, \quad (62)
\]

\[
\Delta_b \geq 0, \quad \lambda_b \geq 0, \quad \lambda_b \Delta_b = 0 \quad (63)
\]

for \( b \in [0, B - 1] \). From \( \frac{\partial L}{\partial \Delta_b} = 0 \), \( \Delta_b \) is given by

\[
\lambda_b = 1 - \nu \cdot \frac{4^b}{\sqrt{2\pi\sigma}} \exp \left( -\frac{\Delta_b^2}{2\sigma^2} \right) \geq 0. \quad (64)
\]

By (63) and (64), we obtain

\[
\Delta_b \left\{ 1 - \nu \cdot \frac{4^b}{\sqrt{2\pi\sigma}} \exp \left( -\frac{\Delta_b^2}{2\sigma^2} \right) \right\} = 0. \quad (65)
\]

If \( \nu = 0 \), then \( \lambda_b = 1 \) and \( \Delta_b = 0 \) for any \( b \in [0, B - 1] \) because of (63) and (64). Since \( \Delta = 0 \) is a trivial solution, we claim that \( \nu \neq 0 \), which results in

\[
\sum_{b=0}^{B-1} 4^b Q \left( \frac{\Delta_b}{\sigma} \right) = \mathcal{V}. \quad (66)
\]

If \( \nu \leq \frac{\sqrt{2\pi\sigma}}{4} \), then \( \Delta_b > 0 \) is impossible because it would imply \( \lambda_b = 0 \) and \( \nu = \frac{\sqrt{2\pi\sigma}}{4} \exp \left( \frac{\Delta_b^2}{2\sigma^2} \right) \), which contradicts the condition \( \nu \leq \frac{\sqrt{2\pi\sigma}}{4} \). Hence, \( \Delta_b = 0 \) for \( \nu \leq \frac{\sqrt{2\pi\sigma}}{4} \).

If \( \nu > \frac{\sqrt{2\pi\sigma}}{4} \), then \( \Delta_b = 0 \) is impossible because it would imply \( \nu = \frac{\sqrt{2\pi\sigma}}{4} \exp \left( \frac{\Delta_b^2}{2\sigma^2} \right) = \frac{\sqrt{2\pi\sigma}}{4} \), which contradicts the condition \( \nu > \frac{\sqrt{2\pi\sigma}}{4} \). We claim that \( \Delta_b > 0 \) and \( \lambda_b = 0 \), which results in and (21) for \( \nu > \frac{\sqrt{2\pi\sigma}}{4} \). Thus, the optimal solution \( \Delta^\star \) of (18) can be derived from (19).

APPENDIX B
PROOF OF THEOREM 10

The KKT conditions of (24) are as follows:

of water-filling that may involve sand-pouring and ground-flattening. Also, we developed optimization algorithms for discrete swings by leveraging water-filling interpretations and KKT conditions. The discrete swings obtained by proposed algorithms achieve almost the same energy and EDP savings as the continuous swings for moderate granularity.
Fig. 10. Optimal solutions (a) minimizing energy, (b) maximizing speed, and (c) minimizing EDP.

\[
E(\Delta) / B \leq \begin{cases} 0 & \text{if } \nu \geq 0, \\
\nu \cdot \left\{ \sum_{b=0}^{B-1} 4^b \left( \frac{\Delta_b}{\sigma} \right) - \nu \right\} & \text{if } 0 \leq \Delta_b \leq \xi, \quad \lambda_0 \geq 0, \quad \lambda_b \Delta_b = 0, \\
\eta_0 & \text{if } \eta_0(\Delta_b - \xi) = 0. 
\end{cases}
\]

(67)

\[
\nu > 0, \quad \sum_{b=0}^{B-1} 4^b Q \left( \frac{\Delta_b}{\sigma} \right) = \nu, \quad \nu \geq 0.
\]

(68)

For \( \nu = 0 \), the EDP in (70) implies that \( \eta_0(\Delta_b - \xi) = 0 \) for all \( b \in [0, B - 1] \). Hence, we claim that

\[
\nu > 0, \quad \sum_{b=0}^{B-1} 4^b Q \left( \frac{\Delta_b}{\sigma} \right) = \nu. \quad \nu \geq 0.
\]

(69)

From (71), \( \nu \leq \eta_0 \cdot \frac{\sqrt{2\pi} \sigma}{\Delta_b} \exp \left( \frac{\Delta_b^2}{2\sigma^2} \right) \). If \( \nu \geq \eta_0(\Delta_b - \xi) = 0 \) from (72), both \( \eta_0(\Delta_b) = 0 \) and \( \eta_0(\Delta_b - \xi) = 0 \) result in \( \eta_0 = 0 \) for any \( b \in [0, B - 1] \), which violates (72). Hence, we claim that

\[
\eta_b = \nu \cdot \frac{4^b}{\sqrt{2\pi} \sigma} \exp \left( -\frac{\Delta_b^2}{2\sigma^2} \right) \Delta_b = 0. \quad \nu > 0, \quad \nu \geq 0.
\]

(70)

If \( \nu = 0 \), the EDP in (71) implies that \( \eta_0(\Delta_b - \xi) = 0 \) for all \( b \in [0, B - 1] \). Hence, we claim that

\[
\nu > 0, \quad \sum_{b=0}^{B-1} 4^b Q \left( \frac{\Delta_b}{\sigma} \right) = \nu. \quad \nu \geq 0.
\]

(71)

From (71), \( \nu \leq \eta_0 \cdot \frac{\sqrt{2\pi} \sigma}{\Delta_b} \exp \left( \frac{\Delta_b^2}{2\sigma^2} \right) \). If \( \nu \leq \eta_0(\Delta_b - \xi) = 0 \) from (72), both \( \eta_0(\Delta_b) = 0 \) and \( \eta_0(\Delta_b - \xi) = 0 \) result in \( \eta_0 = 0 \) for any \( b \in [0, B - 1] \), which violates (72). Hence, we claim that

\[
\eta_b = \nu \cdot \frac{4^b}{\sqrt{2\pi} \sigma} \exp \left( -\frac{\Delta_b^2}{2\sigma^2} \right) \Delta_b = 0. \quad \nu > 0, \quad \nu \geq 0.
\]

(72)

From (71), \( \nu \leq \eta_0 \cdot \frac{\sqrt{2\pi} \sigma}{\Delta_b} \exp \left( \frac{\Delta_b^2}{2\sigma^2} \right) \). If \( \nu \leq \eta_0(\Delta_b - \xi) = 0 \) from (72), both \( \eta_0(\Delta_b) = 0 \) and \( \eta_0(\Delta_b - \xi) = 0 \) result in \( \eta_0 = 0 \) for any \( b \in [0, B - 1] \), which violates (72). Hence, we claim that

\[
\eta_b = \nu \cdot \frac{4^b}{\sqrt{2\pi} \sigma} \exp \left( -\frac{\Delta_b^2}{2\sigma^2} \right) \Delta_b = 0. \quad \nu > 0, \quad \nu \geq 0.
\]

(73)

Fig. 11. Energy consumption of discrete swings obtained by Algorithm 1 and the Levin–Campello algorithm for (a) \( B = 8 \) and (b) \( B = 16 \).
for all \( b \in [0, B - 1] \). From \( \frac{\partial L_A}{\partial q_b} = 0 \) and \( \frac{\partial L_A}{\partial x} = 0 \), we obtain the following equations:

\[
\xi + \eta_b = \lambda_b + \nu \cdot \frac{4^b}{\sqrt{2\pi\sigma}} \exp \left( -\frac{\Delta_b^2}{2\sigma^2} \right), \quad (82)
\]

\[
\sum_{b=0}^{B-1} \Delta_b = \sum_{b=0}^{B-1} \eta_b, \quad (83)
\]

Suppose that \( \nu = 0 \), then \( \xi + \eta_b = \lambda_b \) for all \( b \in [0, B - 1] \), which implies \( (\xi + \eta_b) \Delta_b = 0 \) because of (80). For \( b \) such that \( \Delta_b \neq 0 \), \( \eta_b = 0 \) because of \( \xi + \eta_b = 0 \), \( \lambda_b \geq 0 \) and \( \xi \geq 0 \). For \( b \) such that \( \Delta_b = 0 \), \( \eta_b = 0 \) because of (81). Hence, if \( \nu = 0 \), then \( \eta_b = 0 \) for all \( b \in [0, B - 1] \), which implies \( \Delta_b = 0 \) for all \( b \in [0, B - 1] \) due to \( \Delta_b \geq 0 \) and (83). Thus, we claim that

\[
\nu > 0, \quad \sum_{b=0}^{B-1} 4^b Q \left( \frac{\Delta_b}{\sigma} \right) = \mathcal{V} \quad (84)
\]

which is the same as (74).

By (80) and (82),

\[
\lambda_b \Delta_b = \nu \left\{ \frac{\xi + \eta_b}{\nu} - \frac{4^b}{\sqrt{2\pi\sigma}} \exp \left( -\frac{\Delta_b^2}{2\sigma^2} \right) \right\} \Delta_b = 0 \quad (85)
\]

where \( \frac{\nu}{\xi + \eta_b} \leq \frac{\sqrt{2\pi\sigma}}{4^b} \exp \left( \frac{\Delta_b^2}{2\sigma^2} \right) \) because of \( \lambda_b \geq 0 \). If \( \frac{\nu}{\xi + \eta_b} \leq \frac{\sqrt{2\pi\sigma}}{4^b} \), then \( \Delta_b = 0 \), which implies \( \eta_b = 0 \) by (81). Hence, we claim that

\[
\Delta_b = 0, \quad \eta_b = 0, \quad \text{if} \quad \frac{\nu}{\xi} \leq \frac{\sqrt{2\pi\sigma}}{4^b}. \quad (86)
\]

If \( \frac{\nu}{\xi + \eta_b} > \frac{\sqrt{2\pi\sigma}}{4^b} \), then \( \Delta_b > 0 \) and

\[
\nu = \frac{\sqrt{2\pi\sigma}}{4^b} \cdot \eta_b \cdot \exp \left( \frac{\rho^2}{2\sigma^2} \right) \quad (77)
\]

which is equivalent to (27). From (72) and (77), we obtain (25) and (28).

**APPENDIX C**

**PROOF OF THEOREM 13 AND COROLLARY 14**

The KKT conditions of (32) are as follows:

\[
\sum_{b=0}^{B-1} 4^b Q \left( \frac{\Delta_b}{\sigma} \right) \leq \mathcal{V}, \quad \nu \geq 0, \quad (78)
\]

\[
\nu \cdot \left\{ \sum_{b=0}^{B-1} 4^b Q \left( \frac{\Delta_b}{\sigma} \right) - \mathcal{V} \right\} = 0, \quad (79)
\]

0 \leq \Delta_b \leq \xi, \quad \lambda_b \geq 0, \quad \lambda_b \Delta_b = 0, \quad (80)

\eta_b \geq 0, \quad \eta_b (\Delta_b - \xi) = 0 \quad (81)

\[
\frac{\nu}{\xi + \eta_b} = \frac{\nu}{\xi - \lambda_b} = \frac{\sqrt{2\pi\sigma}}{4^b} \exp \left( \frac{\Delta_b^2}{2\sigma^2} \right), \quad (90)
\]

By (87) and (90),

\[
\frac{\nu}{\xi + \eta_b} = \frac{\nu}{\xi - \lambda_b} = \frac{\sqrt{2\pi\sigma}}{4^b} \exp \left( \frac{\Delta_b^2}{2\sigma^2} \right), \quad (91)
\]
for \(0 < b \Delta_b < \xi, \xi + \eta_b = \xi - \lambda_b \) (i.e., \(\eta_b = -\lambda_b \)) means \(\eta_b = \lambda_b = 0\) because of \(\eta_b \geq 0\) and \(\lambda_b \geq 0\). Hence, we claim that

\[
\nu = \frac{\sqrt{2\pi\sigma}}{4b} \exp\left(-\frac{b^2 \sigma^2}{2\sigma^2}\right), \quad \eta_b = \lambda_b = 0
\]

(92)

For \(\xi < \frac{\sqrt{2\pi\sigma}}{4b} \exp\left(-\frac{b^2 \sigma^2}{2\sigma^2}\right)\), there should exist \(\eta_b > 0\) for \(b \in [0, B - 1]\) to make \(\sum_{b=0}^{B} \Delta_b > 0\). Hence, there exists \(\Delta_b = \xi\) due to (81), which implies \(\rho = \max(\Delta) = \xi\). From (86), we can obtain the solution \(\Delta^*\) as (35).

Note that \(s_b > 0\) for \(\Delta_b = \rho\) and \(\lambda_b = 0\). In this case, (82) can be modified into

\[
\rho + \eta_b = \nu \frac{4b}{\sqrt{2\pi\sigma}} \exp\left(-\frac{\rho^2}{2\sigma^2}\right)
\]

(93)

As shown in Fig. 5(a), the sand depth \(s_b\) is given by

\[
s_b = \log_{\frac{\nu}{\rho}} \frac{\log_{\frac{\nu}{\rho}} \left(\frac{\sqrt{2\pi\sigma}}{4b} + \frac{\rho^2}{2\sigma^2}\right)}{\rho - \eta_b}
\]

(94)

\[
= \frac{\nu}{\rho} \log_{\frac{\nu}{\rho}} \left(1 + \frac{\eta_b}{\rho}\right)
\]

(95)

\[
= \log_{\frac{\nu}{\rho}} \left(1 + \frac{\eta_b}{\rho}\right)
\]

(96)

where (95) follows from (93). If \(0 \leq \Delta_b < \rho\), then \(\eta_b = 0\) as shown in (86) and (92). Hence, \(s_b = 0\) for \(0 \leq \Delta_b < \rho\). Hence, (36) in Corollary 14 is proved. Also, (38) in Corollary 14 is derived from (83) and (96).

REFERENCES

[1] M. Horowitz, “Computing’s energy problem (and what we can do about it),” in Proc. IEEE Int. Solid-State Circuits Conf. Dig. Tech. Pap. (ISSCC), Feb. 2014. pp. 10–14.

[2] Y. Chen et al., “DaDianNao: A machine-learning supercomputer,” in Proc. 47-th Annu. IEEE/ACM Int. Symp. Microarchitecture (MICRO), 2014, pp. 609–622.

[3] Y. H. Chen, J. Emer, and V. Sze, “Eyeriss: A spatial architecture for energy-efficient dataflow for convolutional neural networks,” in Proc. ACM/IEEE 43rd Int. Symp. Comput. Architecture (ISCA), Jun. 2016, pp. 367–379.

[4] V. Sze, Y.-H. Chen, T.-J. Yang, and J. Emer, “Efficient processing of deep neural networks: A tutorial and survey,” arXiv preprint arXiv:1703.09039, Mar. 2017. [Online]. Available: http://arxiv.org/abs/1703.09039

[5] C.-P. Lin, P.-C. Tseng, Y.-T. Chiu, S.-S. Lin, C.-C. Cheng, H.-C. Fang, W.-M. Chao, and L.-G. Chen, “A 5mW MPEG4 SP encoder with 2D bandwidth-sharing motion estimation for mobile applications,” in Proc. IEEE Int. Solid-State Circuits Conf. Dig. Tech. Pap. (ISSCC), Feb. 2006, pp. 1626–1635.

[6] S. Han, X. Liu, H. Mao, J. Pu, A. Pedram, M. A. Horowitz, and W. J. Dally, “EIE: Efficient inference engine on compressed deep neural network,” in Proc. ACM/IEEE 43rd Int. Symp. Comput. Architecture (ISCA), Jun. 2016, pp. 243–254.

[7] Y. H. Chen, T. Krishna, J. S. Emer, and V. Sze, “Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks,” IEEE J. Solid-State Circuits, vol. 52, no. 1, pp. 127–138, Jan. 2017.

[8] B. Zhao, D. Blaauw, D. Sylvester, and S. Hanson, “A Sub-200mV 6T SRAM in 0.13 µm CMOS,” in Proc. IEEE Int. Solid-State Circuits Conf. Dig. Tech. Pap. (ISSCC), Feb. 2007, pp. 332–606.

[9] M. H. Abu-Rahma, M. Anis, and S. S. Yoon, “Reducing SRAM power using fine-grained wordline pulsedwidth control,” IEEE Trans. VLSI Syst., vol. 24, no. 6, pp. 2397–2401, Jun. 2016.

[10] X. Li, “Maximum-information storage system: Concept, implementation and application,” in Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD), Nov. 2010, pp. 39–46.
[34] M. Abu-Rahma and M. Anis, *Nanometer Variation-Tolerant SRAM: Circuits and Statistical Design for Yield*. Springer Publishing Company, 2012.

[35] B. S. Leibowitz, J. Kim, J. Ren, and C. J. Madden, “Characterization of random decision errors in charged comparators,” in *Proc. IEEE Custom Integrated Circuits Conf. (CICC)*, Sep. 2008, pp. 691–694.

[36] D. Knighton, *Fluvial Forms and Processes: A New Perspective*. London, UK: Arnold, 1998.

[37] J. Maiz, S. Hareland, K. Zhang, and P. Armstrong, “Characterization of multi-bit soft error events in advanced SRAMs,” in *IEEE Int. Electron Devices Meeting (IEDM) Tech. Dig.*, Dec. 2003, pp. 21.4.1–21.4.4.

[38] K. Osada, Y. Saitoh, E. Ibe, and K. Ishibashi, “16.7-fA/cell tunnel-leakage-suppressed 16-Mb SRAM for handling cosmic-ray-induced multiterrors,” *IEEE J. Solid-State Circuits*, vol. 38, no. 11, pp. 1952–1957, Nov. 2003.

[39] B. Fox, “Discrete optimization via marginal analysis,” *Manag. Sci.*, vol. 13, no. 3, pp. 210–216, 1966.

[40] J. Campello, “Optimal discrete bit loading for multicarrier modulation systems,” in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Aug. 1998, p. 193.

[41] ———, “Practical bit loading for DMT,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 1999, pp. 801–805.

[42] “Usc-sipi image database, univ. southern california.” [Online]. Available: http://sipi.usc.edu/database/?volume=misc