

# eDRAM-Based Tiered-Reliability Memory with Applications to Low-Power Frame Buffers

Kyungsang Cho<sup>†‡</sup> Yongjun Lee<sup>†‡</sup> Young H. Oh<sup>†</sup> Gyoo-cheol Hwang<sup>†</sup> Jae W. Lee<sup>‡</sup>

<sup>†</sup>Samsung Electronics  
Hwaseong, Korea

<sup>‡</sup>Sungkyunkwan University  
Suwon, Korea

{kyungsang.cho, yonjun80.lee, gchwang}@samsung.com

{loias, yongjunlee, garion9013, jaewlee}@skku.edu

## ABSTRACT

Embedded DRAM (eDRAM) is becoming more and more popular as a low-cost alternative to on-chip SRAM. eDRAM is particularly attractive for frame buffers in video applications with ever increasing screen resolutions. However, eDRAM suffers short retention time and high refresh power, which prevents its widespread adoption. To save the refresh power of eDRAM-based frame buffers, we propose *Tiered-Reliability Memory* (TRM), where the frame buffer is divided into multiple segments with different refresh periods and hence different error rates. By allocating most-significant bits to the most reliable segment, our four-tier TRM reduces refresh power by 48% without degrading user experience.

## Categories and Subject Descriptors

B.3.1 [Hardware]: Memory Structures—*Semiconductor Memories*

## Keywords

eDRAM; Frame buffer; Refresh; Error tolerance; Low power

## 1. INTRODUCTION

Many complex system-on-chip's (SoCs) require large on-chip buffers, which often become a determining factor of chip area and power consumption. Recently, Embedded DRAM (eDRAM) has emerged as a low-cost replacement for conventional on-chip SRAM. For example, several multicore processors have adopted eDRAM-based last-level caches integrated with processing elements either on the same die [15] or in the same package [16]. Compared to conventional SRAM-based caches and register files, eDRAM-based caches and register files have an advantage of much higher density with lower leakage power [9, 17].

The eDRAM-based on-chip storage is attractive not only for caches but also for frame buffers in display driver and video processing SoCs. With ever increasing screen resolutions, the frame buffer organization will have an even greater impact on chip area and power. As shown in Figure 1, the

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or to publish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).  
ISLPED'14, August 11–13, 2014, La Jolla, CA, USA.

Copyright is held by the owner/author(s). Publication rights licensed to ACM.

ACM 978-1-4503-2975-0/14/08 ...\$15.00.

<http://dx.doi.org/10.1145/2627369.2627626>.

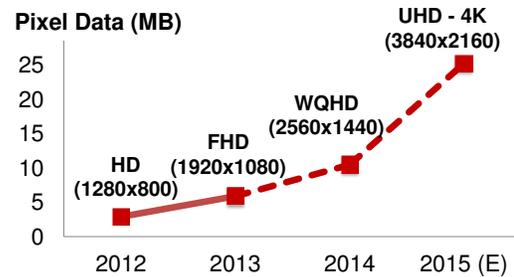


Figure 1: Trends of graphics display resolutions [3]

ultra high definition (UHD) resolution video will become commonplace in the near future for which a 24MB buffer is required to store a single video frame.

However, unlike SRAM, eDRAM cells need to be periodically refreshed to preserve stored bits, which becomes the main source of power consumption [9]. In the context of the commodity off-chip DRAM, researchers have proposed techniques to extend DRAM refresh period by using error correcting code (ECC) [11, 35], retention-aware DRAM page allocation [32] and refresh [22], and memory access-aware refresh [12]. However, the cell capacitance of eDRAM is much smaller than that of the commodity DRAM, and the refresh power problem is of a greater concern for eDRAM. Even worse, with technology scaling the cell capacitance will continue to shrink only to exacerbate this problem.

One nice property of the video and display applications is that they can tolerate errors in the pixel data. Besides, the human visual system (HVS) is known to be more sensitive to a change in the higher-order bits of a pixel value than the lower-order bits. Leveraging these properties we can effectively trade the accuracy of the pixel data for power savings in frame buffers with minimal degradation of video quality.

To exploit these opportunities, there are proposals for heterogeneous SRAM-based frame buffers, which consist of multiple segments with different error rates. This heterogeneity is realized by controlling transistor sizing [20], operating voltage [10], or number of transistors per cell [8, 13]. By allocating higher-order bits to more reliable segments power consumption can be significantly reduced with an (almost) undetectable degradation of the video quality. However, they are still based on expensive SRAMs and lack flexibility since the segment configuration, specified by the number of segments and their error rates, is fixed at design time.

This paper proposes Tiered-Reliability Memory (TRM) to effectively exploit power-accuracy tradeoffs in eDRAM-based frame buffers. In TRM, the memory array is divided into multiple segments that can be independently refreshed with different periods. Unlike previous DRAM refresh control schemes targeted for general-purpose computing plat-

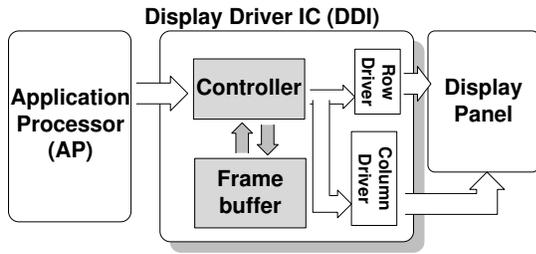


Figure 2: Mobile display sub-system

forms [22, 23], TRM is specialized for display applications and assigns data criticality at a sub-word (i.e., sub-pixel) granularity. By allocating the highest-order bits of a pixel to the most reliable segment, TRM achieves significant savings of refresh power without compromising display quality. Moreover, unlike heterogeneous SRAM-based frame buffers [8, 10, 13, 20], the segment configuration can be adjusted in the field, depending on the application and user preference.

## 2. BACKGROUND

### 2.1 Display Sub-system

Figure 2 illustrates a mobile display sub-system, which consists of an application processor (AP), a display driver IC (DDI) and a display panel. The DDI has an on-chip frame buffer that keeps pixel data and sends them to the display panel periodically, say at 60 Hz. Each pixel consists of three color components representing red (R), green (G), and blue (B), and each component typically takes one byte. The display sub-system is the most power-consuming block in a mobile device [7, 28] whose power efficiency is highly desirable to reduce the overall system power.

In DDI, the frame buffer accounts for a dominating fraction of total chip area and power consumption [5, 19]. SRAM is a traditional choice for the frame buffer, but eDRAM is becoming more and more popular for its cost and power benefits. With ever increasing screen resolutions, the organization and power efficiency of the frame buffer will have a profound impact on the overall system cost and power.

### 2.2 Embedded DRAM

Recently, eDRAM has emerged as a low-cost alternative to SRAM in organizing large-capacity memories on a chip. SRAM is still the most popular technology for on-chip storage due to its low access time and compatibility with the standard CMOS logic process. However, SRAM is based on 6-transistor (6T) cells (or enhanced 8T or 10T cells [25]), which incur significant area overhead and high leakage power. Unlike SRAM, eDRAM uses much fewer transistors per cell, leading to higher density and lower leakage power.

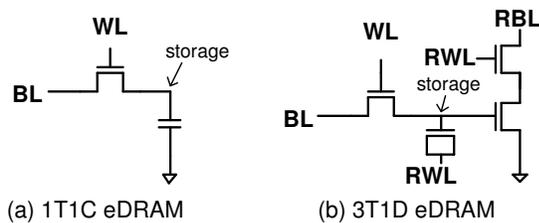


Figure 3: eDRAM cell schematics

	SRAM (6T)	eDRAM	
		1T1C [6]	3T1D [24]
Cell size	1×	0.22×	0.64×
Latency	Good	Poor	Good
Process	Logic compatible	Trench capacitor	Logic compatible
Leakage	Poor	Good	Good
Retention	$\infty$	40 $\mu$ s	200 $\mu$ s

Table 1: Comparison of SRAM and eDRAM at 65nm

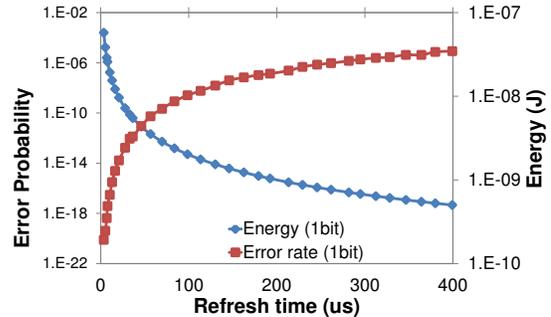


Figure 4: Tradeoff between bit error rate and refresh energy

There are two popular variants of eDRAM among others: 1T1C (trench capacitor) eDRAM and 3T1D (gain cell) eDRAM. Table 1 compares the two types of eDRAM with SRAM, and Figure 3 illustrates their cell structures. The 1T1C cell has the same structure with that of the commodity DRAM and is about 4-5× smaller than the 6T SRAM cell. However, it requires additional process masks to embed trench capacitors and has slow access time due to destructive reads [21]. In contrast, the 3T1D eDRAM uses additional transistors to overcome these limitations [24]. The 3T1D cell is based on the 3T DRAM cell in 1970s, and a gated diode is added to dynamically amplify the storage capacitance. Compared to the 1T1C eDRAM, the 3T1D eDRAM has advantages of low fabrication cost by obviating the needs for additional masks and fast access time comparable to SRAM with non-destructive reads.

Regardless of the types of eDRAM, the eDRAM cells need to be periodically refreshed to preserve stored bits, and the refresh power dominates the overall memory power consumption [35]. eDRAM has a smaller cell capacitance than the commodity DRAM. Hence, the retention time becomes shorter, and the refresh operation should be performed much more frequently. This leads to higher refresh power, which is a serious concern for eDRAM-based on-chip memories.

Both bit error rate and refresh power are functions of the refresh period, and there is a tradeoff between the two. Figure 4 illustrates this tradeoff. The refresh power, whose model is detailed in Section 4, is inversely proportional to the refresh period, whereas the eDRAM bit error rate increases exponentially as the refresh period increases [23, 35].

In video and display applications, which can tolerate bit errors (i.e., pixel inaccuracies), we can exploit power-accuracy tradeoffs by controlling refresh period. With *data criticality-aware non-uniform refreshment*, we can save a significant fraction of the refresh power with minimal degradation of the video quality.

## 3. TIERED-RELIABILITY MEMORY

TRM enables data criticality-aware non-uniform refreshment to save refresh power in eDRAM-based frame buffers.

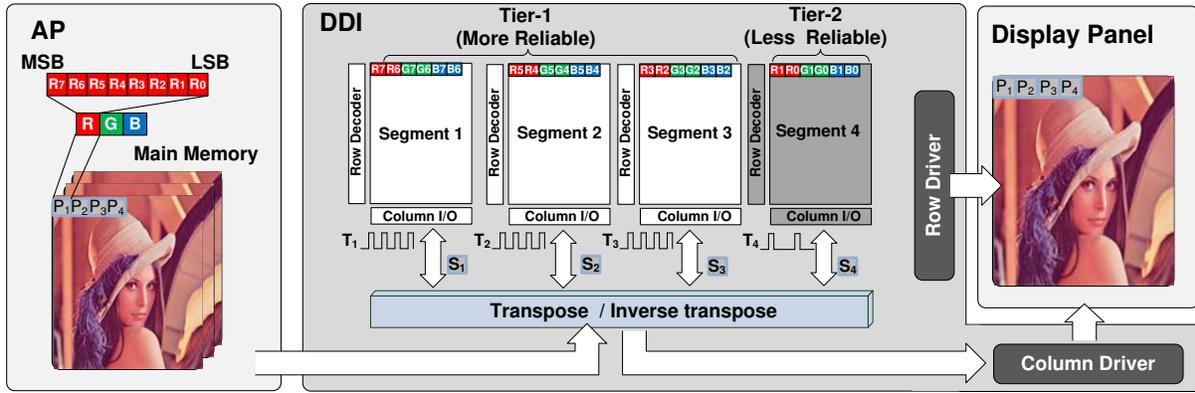


Figure 5: Mobile display sub-system augmented with eDRAM-based Tiered-Reliability Memory (TRM)

To realize this, TRM combines the following three components synergistically:

- (Multiple segments) The frame buffer is partitioned into multiple *segments* that can be refreshed independently with different periods.
- (Bit transpose) The incoming stream of pixel bits are rearranged at a sub-pixel granularity and allocated to segments according to their criticality. The MSBs of a color component are allocated to the most reliable segment, and the LSBs to the least reliable segment.
- (Optimal refresh period vector) The refresh period of each segment should be set optimally to balance refresh power savings with video quality degradation.

### 3.1 TRM Organization

Figure 5 illustrates a TRM-based display sub-system with a 4-segment frame buffer. The refresh period of each segment can be set independently from other segments to control its error rate. The refresh period of Segment  $i$  is denoted by  $T_i$  and represented as a multiple of the *nominal* refresh period ( $T_0$ ), which is the refresh period for reliable operation. We assume that, if  $T_i \leq 1$  (i.e., refresh period is equal to or shorter than the nominal refresh period), the error rate of Segment  $i$  will be zero. The nominal refresh period is determined by cell retention time.

Table 2 summarizes TRM parameters. The refresh period vector ( $\mathbf{V}$ ) is a vector of length  $N$  containing  $T_1, T_2, \dots, T_N$ . The number of tiers can be up to the number of segments. For example, the DDI in Figure 5 illustrates a two-tier TRM with  $\mathbf{V} = \langle 1, 1, 1, 2 \rangle$ , where Segments 1, 2, and 3 constitute more reliable Tier 1 with a refresh period of  $T_0$  and Segment 4 constitutes less reliable Tier 2 with a refresh period of  $2T_0$ .

Parameter	Descriptions
$N$	Number of segments
$T_0$	Nominal refresh period
$T_i$	Refresh period of Segment $i$
$\mathbf{V} = \langle T_1, T_2, \dots, T_N \rangle$	Refresh period vector

Table 2: TRM parameters

Independent refresh operation from each segment requires per-segment row and column access circuitry. However, this hardware overhead is minimal since most DDIs have already

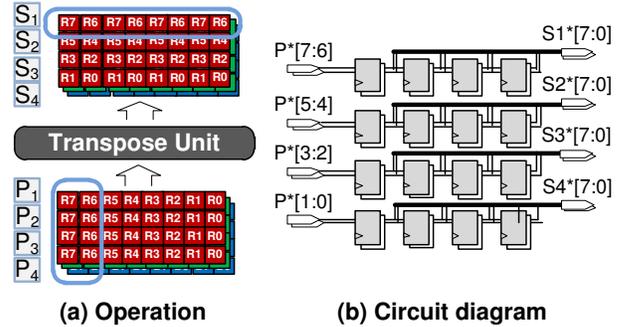


Figure 6: Transpose unit (\* denotes either R, G, or B)

adopted multi-bank frame buffers to reduce the internal operating frequency by accessing multiple banks simultaneously. In such a case, supporting programmable refresh periods is the only required modification to the bank structure.

To make the best use of the frame buffer with multiple reliability tiers, we should sort incoming pixel data by their criticality and allocate the most critical data to the most reliable segment. By doing this refresh power is allocated to data proportionally to their criticality. In video applications, the MSBs of a pixel (or color component) are more critical for human visual perception, so they should be allocated to more reliable memory (e.g., Tier 1 in Figure 5), and LSBs to less reliable memory (e.g., Tier 2).

With the transpose unit in Figure 6, we can cluster pixel bits by their criticality, hence eliminating criticality fragmentations. In the DDI, pixel data arrive in a raster scan order and are stored across multiple banks in the frame buffer by interleaving them at a *pixel* granularity. The transpose unit in Figure 6 gathers the 7th and 6th bits (R7 and R6) of the four pixels (P1 through P4) and forwards them to Segment 1 (S1) as a group. For each set of four pixels, the three color components (R, G, and B) are processed in parallel. This circuit requires 48 flip-flops for each of transpose and inverse transpose, respectively, whose area overhead is negligible compared to the frame buffer.

### 3.2 TRM Parameter Selection

This section discusses various aspects to consider in selecting the following two TRM parameters: number of segments ( $N$ ) and refresh period vector ( $\mathbf{V}$ ).

**Number of Segments ( $N$ ):** More segments enable more fine-grained control of refresh periods at the cost of additional per-segment hardware such as row decoder and refresh control logic. In the frame buffer, the maximum number of  $N$  is practically upper bounded by the number of bits for each color component, which is 8 in our setup.

**Refresh Period Vector ( $\mathbf{V}$ ):** A careful selection of  $\mathbf{V}$  is necessary to balance power savings and video quality degradation.  $\mathbf{V}$  controls the error rate of each segment and can be adjusted/tuned in the field. Since the display panel reads (and refreshes) pixel data from the frame buffer periodically, the frame rate sets an upper bound on the refresh period. For example, for the frame rate of 60Hz, this upper bound is  $1/60=16.7$  ms. If an image in the frame buffer does not change indefinitely, the pixel data in those segments with  $T_i > 1$  will be eventually broken, hence degrading the image quality. To prevent this, we assume a fresh copy of the image is sent from the AP to the DDI every 5 minutes (300 seconds) even if the image being displayed does not change. Evaluation with varying  $\mathbf{V}$  will be presented in Section 5.

## 4. MODELS AND METRICS

In this section we first describe the error rate model and power model that we use for evaluation, as functions of the refresh period. They are followed by the two metrics to quantify the degradation of video quality with TRM.

### Error Injection Methodology

We use as baseline the eDRAM bit error model introduced by Wilkerson et al. [35], which is reproduced in Figure 4. We assume bit retention failures are distributed randomly throughout the frame buffer. By applying their methodology to a 24MB UHD frame buffer with the same target failure rate, we obtain the nominal refresh period ( $T_0$ ) of  $44\mu s$ . Our error model assumes zero error rate for a refresh period shorter than  $T_0$ ; otherwise, the bit error rate will follow the curve in Figure 4.

For evaluation the TRM simulator injects a random bit error for all bits in the frame buffer with a probability of the bit error rate corresponding to the refresh period. For each bit this injection process is repeated by the maximum number of refreshes that can happen without updating the frame buffer. Once error injection is finished for the entire frame, we compare the error-injected frame with the original frame to quantify the degradation of image quality.

### Refresh Power Model

Based on the refresh power model of Tran et al. [31], the refresh power of TRM can be represented as follows:

$$P_{refresh} = \sum_{i=1}^N \frac{R_i \cdot E_{refresh}}{T_i} + P_C \approx \sum_{i=1}^N \frac{R_i \cdot E_{refresh}}{T_i}$$

where  $P_{refresh}$  is a sum of the refresh power of all segments and  $R_i$  is the fraction of the size of Segment  $i$  in the frame buffer;  $T_i$  is the refresh period of Segment  $i$ ;  $E_{refresh}$  is per-refresh energy consumption;  $P_C$  is the constant power for refresh operation, which typically accounts for less than 10% of refresh power [18].

### Video Quality Metrics

To quantify image/video quality, we use the following two metrics: Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity index (SSIM). Assuming  $X$  and  $Y$  are the original

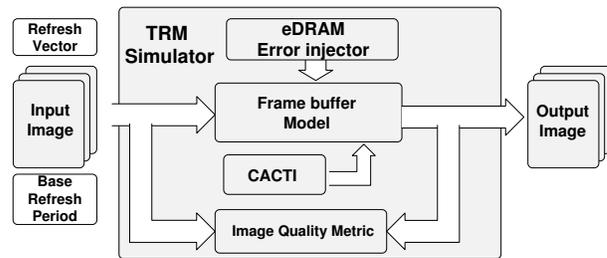


Figure 7: TRM Simulator

and error-injected images, respectively, PSNR is defined as follows:

$$PSNR(dB) = 20 \log\left(\frac{255^2}{MSE}\right), MSE = \sum \frac{(X_{i,j} - Y_{i,j})^2}{n}$$

where  $n$  is the total number of pixels, and mean square error (MSE) is a sum of pixel differences divided by pixel numbers.  $X_{i,j}$  ( $Y_{i,j}$ ) denotes the pixel value of  $X$  ( $Y$ ) at the  $i^{th}$  row and the  $j^{th}$  column. If PSNR is higher than 40-45 dB, a human cannot tell the difference between the two images. Some vendors call it *visually lossless*.

SSIM is more sensitive to the structure information like the human visual perception system [33] and defined as follows:

$$SSIM = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

where  $\mu_x$  is the average pixel value of the original image and  $\mu_y$  is that of the inaccurate image with TRM.  $\sigma_x$ ,  $\sigma_y$ ,  $\sigma_{xy}$  denote variance of  $X$ , variance of  $Y$ , and covariance of  $X$  and  $Y$ , respectively.  $C_1$  and  $C_2$  are 2.55 and 7.56 for the RGB color space. Based on the SSIM score whose range is from 0 to 1, the frame quality is classified into high (0.98-1), medium (0.96-0.98), or low (0.94-0.96).

## 5. EVALUATION

### 5.1 Methodology

To evaluate TRM, we have built a TRM simulator as shown in Figure 7. The simulator takes as input an input image, refresh period vector and nominal refresh period to output an inaccurate output image. CACTI [30] is used to model the power consumption. We assume a 4-segment TRM as shown in Figure 5 and a fresh frame sent to the frame buffer from the AP every 5 minutes (if not more frequently on the user's demand) as discussed in Section 3.2.

We choose input images from multiple sources for wide coverage of use cases, such as USC-SIPI [34], video clips [4], compress test image set [1], Kodak image set [2], and web pages (e.g., Google homepage). If an input image is not provided in the UHD resolution, we resize the image using the NumPy package [26]. To be visually lossless, PSNR should be greater than 45 dB, and SSIM greater than 0.98.

### 5.2 Results

An optimal refresh period vector allows us to achieve maximum power savings while staying visually lossless. To find one, we use 5 representative images taken from 5 different sources to measure PSNR, SSIM, and refresh power with varying refresh period vectors. Figure 8 shows the results,

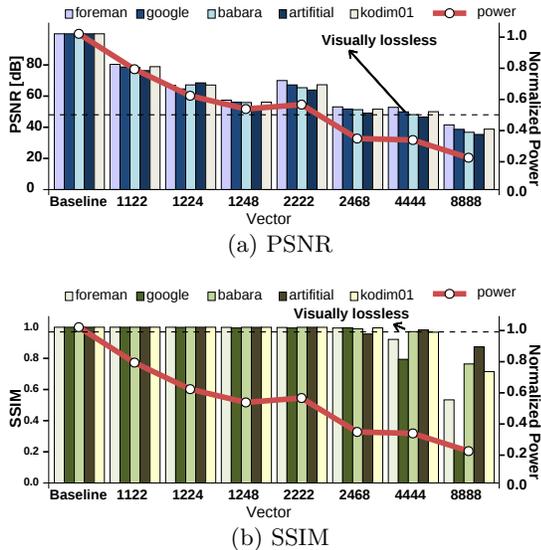


Figure 8: Video quality and refresh power with varying refresh period vectors for TRM

where the baseline is  $\mathbf{V}=\langle 1,1,1,1 \rangle$ . All measured values are normalized to the corresponding values of the baseline. From this simulation, we obtain  $\langle 1,2,4,8 \rangle$  as an optimal vector, which has the lowest power (i.e., 48% reduction) while being visually lossless.

Figure 8 shows that most refresh period vectors are visually lossless when  $T_1$  is 1; that is, as long as 2 MSBs are preserved, we cannot detect the degradation of image quality. This justifies criticality-aware power allocation to pixel bits. If only  $T_1$  is one and the other  $T_i$ 's are greater than one, as time goes on, all eDRAM cells except those in Segment 1 will eventually be discharged. This image represents the worst-case degradation of the image quality, and we call it *lower-bound* image.

Figure 9 shows the image quality of a wide variety of test images with  $\mathbf{V}=\langle 1,2,4,8 \rangle$ . With TRM all test images are visually loseless while consuming much less refresh power. Still, the image quality is much better than the lower bound. To bound the image quality degradation with TRM, the AP periodically resends the frame buffer with a fresh copy of the image. With this support, PSNR and SSIM stay above the cut-offs to be visually loseless while TRM achieves significant power savings.

Figure 10 shows the image quality (SSIM) with varying frame update intervals. The frame update interval is the time interval between adjacent frames sent from the AP. If the interval is shorter than 0.5 seconds (like playing video),  $\mathbf{V}=\langle 8,8,8,8 \rangle$  is the optimal choice to minimize power consumption while remaining visually loseless. For an interval longer than 2 seconds, this vector starts to become visually lossy, and  $\langle 4,4,4,4 \rangle$  is the optimal choice, and so on. This result demonstrates that the optimal vector differs depending on the usage scenario. For a scenario displaying a still image (like reading an e-book) with a long frame update interval (say, >2 minutes),  $\mathbf{V}=\langle 1,2,4,8 \rangle$  becomes optimal. Although  $\langle 1,2,4,8 \rangle$  and  $\langle 2,2,2,2 \rangle$  are comparable in terms of image quality, the former consumes about 7% less refresh power than the latter.

In summary, the human visual system is tolerant to errors in the low-order bits of the pixel data, but completely discarding the low-order bits would make the image visu-

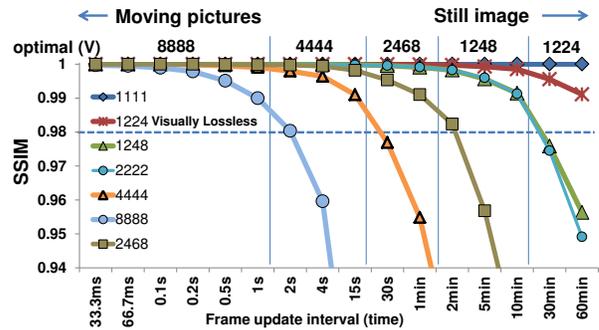


Figure 10: Frame update interval and SSIM with varying refresh vectors for TRM

ally lossy. With prolonged refreshes to those bits and periodic resubmission of the fresh frame from the AP, TRM can achieve significant power savings without degrading user experience.

## 6. RELATED WORK

**Extending DRAM refresh period:** Both Wilkerson et al. [35] and Emma et al. [11] propose to use ECC to extend refresh time. Venkatesan et al. propose retention-aware DRAM page allocation [32] to set the refresh period to be the shortest period only among the populated DRAM pages instead of all pages. Liu et al. exploit variable distribution of cell retention times to extend refresh period [22]. Ghosh et al. save refresh power and bandwidth overhead by skipping refreshes for recently accessed rows [12]. However, all of these proposals do not exploit different criticalities and error tolerance of data elements and are complementary to TRM.

**Approximate computing:** Approximate computing has recently drawn attention from the research community as a means to reduce power consumption by exploiting accuracy-energy tradeoffs. EnerJ [27] introduces a language extension to Java to allow the programmer to annotate which variables can be computed approximately. Also, there are several proposals to save refresh power by increasing refresh periods for non-critical data in commodity DRAM [22, 23]. However, their proposals target general-purpose computing platforms with commodity DRAMs and cannot exploit bit-level criticality, hence suboptimal for video applications. In contrast, TRM is targeted for video applications to achieve much higher power efficiency than these proposals.

**Hybrid on-chip memories:** There are proposals for integrating multiple types of SRAM cells to handle the higher-order bits of pixel data preferentially over the lower-order bits in video applications. Kwon et al. propose Heterogeneous SRAM with varying cell sizes to trade reliability for leakage power savings [20]. Gong et al. introduce an ultra-low voltage split-data-aware 10T and 8T SRAM for the same goal [13]. Chang et al. [8] propose to mix 6T and 8T cells and save powers by aggressive voltage scaling while maintain a high signal-to-noise ratio (SNR). However, those techniques are only applicable to expensive SRAM-based frame buffers but not to eDRAM-based ones. Also, there are proposals to use non-volatile memory (NVM) for frame buffers, including DRAM-PRAM hybrid [14] and DRAM-MRAM-PRAM hybrid [29], but their main focuses are optimizing hybrid memories with different read/write characteristics and endurance.

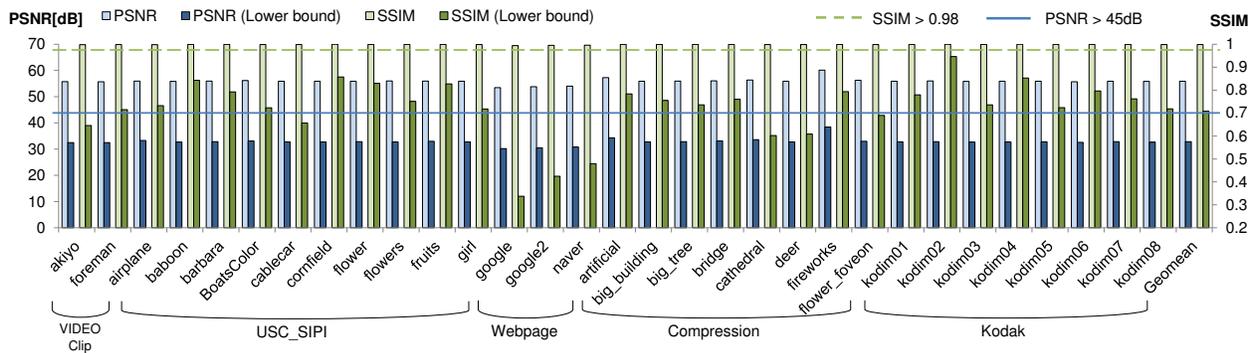


Figure 9: PSNR and SSIM for a wide variety of test images from multiple sources

## 7. CONCLUSION

High refresh power consumption is a serious concern for eDRAM-based on-chip memory, and the problem will get exacerbated with technology scaling, which reduces cell retention time. Furthermore, the screen resolution is expected to increase continuously in the foreseeable future, which will require even higher-capacity frame buffers, hence increasing refresh power. To save refresh power in the eDRAM-based frame buffer, we propose *Tiered-Reliability Memory (TRM)*, which enables the frame buffer to allocate refresh power to pixel data non-uniformly in proportional to their criticality. By judiciously trading data accuracies for power savings, the four-tier TRM achieves 48% savings of refresh power while keeping the video frame visually lossless.

## 8. REFERENCES

- [1] Image compression test images. <http://www.imagecompression.info>.
- [2] Kodak lossless true color image suite. <http://r0k.us/graphics/kodak/>.
- [3] Samsung Analyst Day 2013: Display trends. <http://www.samsung.com/>.
- [4] Xiph.org video test media. <http://xiph.org/video/derf/>.
- [5] C. Argyrides, C. A. Lisboa, L. Carro, and D. K. Pradhan. A soft error robust and power aware memory design. In *Proc. 20th Annu. Symp. Integr. Circuits Syst. Des. (SBCCI)*, pages 300–305. ACM, 2007.
- [6] J. Barth, W. Reohr, P. Parries, G. Fredeman, J. Golz, S. Schuster, R. Matick, H. Hunter, C. Tanner, J. Harig, et al. A 500MHz random cycle 1.5 ns-latency, soi embedded DRAM macro featuring a 3T micro sense amplifier. In *ISSCC, 2007*.
- [7] A. Carroll and G. Heiser. An analysis of power consumption in a smartphone. In *Proc. of USENIX, 2010*.
- [8] I. J. Chang, D. Mohapatra, and K. Roy. A priority-based 6T/8T hybrid SRAM architecture for aggressive voltage scaling in video applications. *IEEE Trans. on CSVT, 2011*.
- [9] M.-T. Chang, P. Rosenfeld, S.-L. Lu, and B. Jacob. Technology comparison for large last-level caches (l3cs): Low-leakage SRAM, low write-energy STT-RAM, and refresh-optimized eDRAM. In *Proc. of HPCA, 2013*.
- [10] M. Cho, J. Schlessman, W. Wolf, and S. Mukhopadhyay. Reconfigurable SRAM architecture with spatial voltage scaling for low power mobile multimedia applications. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, 19(1):161–165, 2011*.
- [11] P. G. Emma, W. R. Reohr, and M. Meterliyo. Rethinking refresh: Increasing availability and reducing power in DRAM for cache applications. In *Proc. of MICRO, 2008*.
- [12] M. Ghosh and H.-H. S. Lee. Smart refresh: An enhanced memory controller design for reducing energy in conventional and 3D die-stacked DRAMs. In *Proc. of MICRO, 2007*.
- [13] N. Gong, S. Jiang, A. Challapalli, S. Fernandes, and R. Sridhar. Ultra-low voltage split-data-aware embedded SRAM for mobile video applications. 2012.
- [14] K. Han, A. W. Min, N. S. Jeganathan, and P. S. Diefenbaugh. A hybrid display frame buffer architecture for energy efficient display subsystems. In *Proc. of ISLPED, 2013*.
- [15] IBM Corp. IBM Power Systems. <http://www-03.ibm.com/systems/power/>.
- [16] Intel Corp. 72-core Knights Landing CPU. <http://newsroom.intel.com/>.
- [17] N. Jing, H. Liu, Y. Lu, and X. Liang. Compiler assisted dynamic register file in gpgpu. In *Proc. of ISLPED, 2013*.
- [18] J. Kim and M. C. Papaefthymiou. Block-based multiperiod dynamic memory design for low data-retention power. *IEEE Trans. on VLSI Systems, 2003*.
- [19] K.-J. Kim, C. H. Kim, and K. Roy. TFT-LCD application specific low power SRAM using charge-recycling technique. In *Proc. of ISQED, 2005*.
- [20] J. Kwon, I. J. Chang, I. Lee, H. Park, and J. Park. Heterogeneous SRAM cell sizing for low-power H.264 applications. *Circuits and Systems I: Regular Papers, IEEE Transactions on, 59(10):2275–2284, 2012*.
- [21] X. Liang and et al. Process variation tolerant 3T1D-based cache architectures. In *MICRO, 2007*.
- [22] J. Liu, B. Jaiyen, R. Veras, and O. Mutlu. Rairdr: Retention-aware intelligent DRAM refresh. In *ISCA, 2012*.
- [23] S. Liu, K. Pattabiraman, T. Moscibroda, and B. G. Zorn. Flicker: saving DRAM refresh-power through critical data partitioning. *ACM SIGPLAN Notices, 47(4):213–224, 2012*.
- [24] W. K. Luk and et al. A 3-transistor DRAM cell with gated diode for enhanced speed and retention time. In *Symposium on VLSI Technology and Circuits, June 2006*.
- [25] H. Noguchi and et al. Which is the best dual-port SRAM in 45-nm process technology?—8T, 10T single end, and 10T differential—. In *Proc. of ICICDT, 2008*.
- [26] T. E. Oliphant. *A Guide to NumPy*, volume 1. Trelgol Publishing USA, 2006.
- [27] A. Sampson, W. Dietl, E. Fortuna, D. Gnanapragasam, L. Ceze, and D. Grossman. Enerj: Approximate data types for safe and general low-power computation. In *ACM SIGPLAN Notices*, volume 46, pages 164–174. ACM, 2011.
- [28] H. Shim, N. Chang, and M. Pedram. A compressed frame buffer to reduce display power consumption in mobile systems. In *Proc. of ASPDAC, 2004*.
- [29] L. C. Stancu and et al. Avid: Annotation driven video decoding for hybrid memories. In *Embedded Systems for Real-time Multimedia (ESTIMedia), 2012 IEEE 10th Symposium on*, pages 2–11. IEEE, 2012.
- [30] S. Thoziyoor, N. Muralimanohar, J. H. Ahn, and N. P. Jouppi. Cacti 5.1. *HP Laboratories, April, 2, 2008*.
- [31] L.-N. Tran and et al. Adjustable supply voltages and refresh cycle for process variations, temperature changes, and device degradation adaptation in 1T1C embedded DRAM. In *Design and Test Workshop (IDT), 2011 IEEE 6th International*, pages 124–129. IEEE, 2011.
- [32] R. K. Venkatesan, S. Herr, and E. Rotenberg. Retention-aware placement in DRAM(RAPID): software methods for quasi-non-volatile dram. In *Proc. of HPCA, 2006*.
- [33] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *Image Processing, IEEE Transactions on, 13(4):600–612, 2004*.
- [34] A. G. Weber. The usc-sipi image database version 5. *USC-SIPI Report*, 315:1–24, 1997.
- [35] C. Wilkerson, A. R. Alameldeen, Z. Chishti, W. Wu, D. Somasekhar, and S.-I. Lu. Reducing cache power with low-cost, multi-bit error-correcting codes. *ACM SIGARCH Computer Architecture News, 38(3):83–93, 2010*.