Content-Aware Low-Power Configurable Aging Mitigation for SRAM Memories

Muhammad Shafique, *Member, IEEE*, Muhammad Usman Karim Khan, and Jörg Henkel, *Fellow, IEEE*

Abstract—Aging through Negative Bias Temperature Instability (NBTI) significantly jeopardizes reliability of SRAM-based memories. We propose a content-aware microarchitectural-level technique for mitigating aging of these SRAM-based memories, by altering the input and output data. The goal is to achieve cost-effective lifetime improvement through low-power aging balancing of all memory cells. For a configurable design, we perform power, area, and aging analysis of different aging balancing circuits. This analysis is leveraged to design a novel aging resilient memory architecture. To curtail the power overhead while still achieving a balanced aging, our architecture employs an anti-aging controller that leverages the data characteristics to take spatio-temporal aging balancing decisions. It dynamically selects: (1) which aging balancing circuit to activate, (2) at what time instant the circuit should be activated, and (3) on which SRAM cells aging balancing should be applied. This is achieved by identifying different configuration parameters, which can be adjusted at run time to balance the aging of SRAM memories. Our experiments demonstrate significant aging improvements at a low power overhead. In addition, we perform sensitivity analysis of different parameters of our architecture to demonstrate power vs. reliability tradeoffs under different run-time scenarios.

Index Terms—Memory, reliability, SRAM, aging, image, video, camera, adaptive, static noise margin, low-power

1 Introduction

Memory intensive applications, e.g., image and video processing have proliferated into various critical domains like surveillance, automotive, satellite imaging, and sensor-based image/video processing over long durations. For these applications, reliable operation over lifetime or an extended lifetime is an important system requirement. To provide fast read/write accesses, application-specific architectures typically employ dedicated SRAM-based memories [1], [2], [3], [4]. However, in the nano-scale regime, these memories are subjected to various reliability threats, out of which aging is critical for lifetime [5], [6], [7], [8], [9].

To address the aging issues in a power-efficient way, we propose content-aware configurable aging optimization of SRAM-based memories deployed in application-specific architectures. It explores the spatio-temporal tradeoffs between aging resilience efficiency and the associated power overhead considering the content properties. In this paper, we first focus on Negative Bias Temperature Instability (NBTI) aging, which is a dominant aging threat in SRAM memories [10], [11], [12], [13], [14]. Towards the end, we will discuss aging due to Hot Carrier Injection (HCI) and impact of our proposed techniques.

- *M. Shafique and J. Henkel are with the Chair for Embedded Systems, Karlsruhe Institute of Technology, Karlsruhe, Germany. E-mail: {muhammad.shafique, henkel}@kit.edu.*
- *M.U.K. Khan is with IBM R&D Böblingen, Germany. E-mail: mkhan@de.ibm.com.*

Manuscript received 27 Aug. 2015; revised 10 Feb. 2016; accepted 14 Mar. 2016. Date of publication 10 Apr. 2016; date of current version 14 Nov. 2016. Recommended for acceptance by S.-Y. Huang.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TC.2016.2553025

Before presenting our novel contributions, we discuss the NBTI-aging in SRAMs in detail and highlight the limitations of state-of-the-art aging mitigation techniques.

1.1 Aging of SRAM-Based Memories

We consider a memory composed of numerous 6T SRAM cells. A 6T-cell is composed of two inverters (each consists of one PMOS and one NMOS transistor) to store complementary values of a bit; see Fig. 1b.[1] The NBTI aging occurs in PMOS transistors due to negative voltage at the gate (i.e., $V_{gs} = -V_{dd}$) that causes stress and breakdown of the Si-H bond at the Si-SiO_2 interface resulting in interface traps; see Fig. 1a. This manifests as an increase in threshold voltage and reduction in noise margin (i.e., short-term aging) that may lead to timing errors/delay faults and/or performance degradation at run time [13], [15], [16]. Once the stress is removed from the PMOS gate (i.e., at $V_{gs} = 0$), the Si-H bond may be reformed in a few cases that corresponds to the partial recovery mode. Such a situation occurs when a "one" stored in the SRAM cell is overwritten with a "zero" and vice versa. Since 100 percent recovery is not possible, NBTI results in continuous degradation over years (i.e., long-term aging), such that, the total aging throughout the lifetime depends upon the stress and recovery cycles; see Fig. 1c.

In case a "zero" or "one" value is stored in an SRAM cell, one of its PMOS transistors will be under stress and the other in the recovery phase. Since the aging of an SRAM cell is determined by the worst-case aging of one of the two PMOS transistors, the overall lowest aging is achieved when both PMOS transistors are stressed by the same amount of time during the whole lifetime. That

1. WL line is used to write a value. BL line is used to carry data to be stored in the cell. The data is retained in the cell by turning off access transistors N3, N4. To read data, the word line is set high and the bitline value is retrieved.

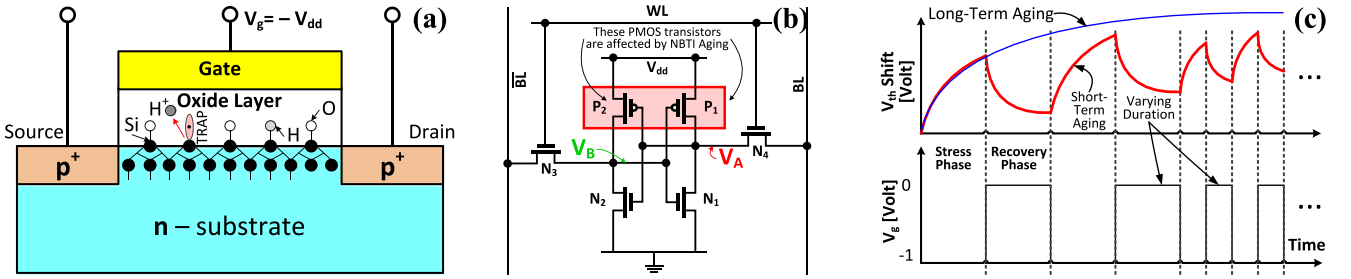


Fig. 1. (a) PMOS transistor under NBTI aging; (b) Standard 6T SRAM cell; (c) An abstract view of the stress and recovery phases for a PMOS transistor.

is, an SRAM cell contains "zero" value for 50 percent of its lifetime and "one" value for the remaining 50 percent of time.

Definition 1 (Duty Factor). *For ease of discussion, we define duty factor (Δ) as the percentage of a cell's lifetime when the stored value is "one". The above case corresponds to $\Delta = 50$ percent.*

The overarching objective of an NBTI-aging mitigation technique is to achieve a balanced duty factor, i.e., $\Delta = 50$ percent. This must be achieved at a minimal power and area overhead. However, achieving such a power-efficient aging balancing is a significant research challenge, especially under varying workloads with diverse data properties (e.g., distinct texture and motion properties in images/videos; see detailed analysis in Section 3). Note that data properties are not known a priori and run-time data diagnostics are expensive.

1.2 State-of-the-Art Techniques and Their Limitations

State-of-the-art techniques primarily target aging optimization for SRAM-based register files. However, these techniques do not target large-sized memories which have distinct access behavior and require different architectural support. The first category of works is based on the principle of "bit rotations" (i.e., moving the least significant bits by one position) to improve the duty factor of registers [17], [18]. These techniques are inefficient for registers with successive zeros and are only beneficial in case the bits inside a register are frequently modified, which is typically not the case for large-sized memories (see Fig. 3 in Section 3). Moreover, implementing bit rotations requires barrel shifters at the read and write ports of the memory. The total number of multiplexers required to implement an n-bit barrel shifter is $nlog_2n$. Therefore, both area and power consumption overhead of such techniques are high.

Another category of work is based on "bit flipping" at every write to the memory [18], [19], [20]. The register value inversion techniques result in significantly more read/writes and power. The recovery boosting technique [20] adds dedicated inverters in the SRAM cells to improve the recovery process. However, this incurs significant power overhead, which may be infeasible for large-sized video memories, for instance, targeting image buffers for high-definition (HD, 1920 $\times$ 1280 bytes) and Quad-HD (QHD, 4 $\times$ 1920 $\times$ 1280 bytes) resolutions. Additionally, it requires an alteration to the 6T SRAM cell circuitry. In [21], a redundancy based SRAM microarchitecture is used for extending the SRAM lifetime. Similar to [20], [22], [23], this also requires modification to the 6T SRAM cell. The work in [24] introduces techniques for balancing the duty factor of SRAM data caches by exploiting cache characteristics (i.e., tag bits). A similar technique is presented in [25], [26]. These techniques depend upon the inherent properties of caches (like flushing, cache hits, etc.) and are not directly applicable to general SRAM memories. Moreover, some of the mentioned balancing policies are designed for a certain bit pattern occurrence and are thus inefficient when considering different content properties with varying stress patterns. Many of the reported works for aging balancing require multiple read/write of the same data in the memory, rendering them to be power hungry.

In summary, state-of-the-art aging balancing techniques for memories employ bit flipping or rotation at every bit level, at every access time, and incur significant power and area overhead. These techniques do not explore the tradeoff between power consumption and aging balancing. Moreover, most of these techniques only provide elementary circuitry without exploring the benefits of different aging balancing techniques and lack a full (micro-)architectural solution with power-aware aging control and adaptations. This paper aims at bridging this gap by developing a full microarchitecture of an aging-resilient SRAM-based memory with a power-aware aging controller. It determines: (1) which aging balancing circuit to activate, (2) at what time instant the circuit should be activated, and (3) on which SRAM cells aging balancing should be applied. This enables a run-time tradeoff between power and aging, especially when different parts of SRAM experience different stress patterns due to varying content properties. In such scenarios, exploitation of application-specific data characteristics may provide a higher potential for duty factor balancing and power savings.

1.3 Our Novel Contributions and Concept Overview

In this work, we propose a microarchitecture-level technique for Application-Specific Configurable Aging Resilience (ASCAR, Section 4, Fig. 6) for SRAM-based memories that enables configurable aging resilience by adaptively exploring the optimization space of run-time aging balancing and associated power overhead.

This paper makes the following novel contributions:

Aging Analysis of Different Aging Balancing Circuits (Section 3): In order to design an efficient aging-resilient memory

architecture, we have implemented different circuit-level techniques to analyze their aging balancing efficiency. For this, we have performed a detailed case study on a memory for camera-based image/video processing architectures. The analysis is performed for different videos with diverse texture and motion properties. It provides guidelines for selecting an appropriate aging balancing circuit as a building block of our ASCAR architecture. We also evaluate the area and power overhead of these circuits.

Memory Read and Write Transducers (MRT, MWT; Section 4.1): The memory architecture is connected to a streaming FIFO for data input. Our architecture contains MRT and MWT that adapt the memory data at the read and write ports, respectively, to achieve a balanced duty factor. The MWT can be configured at run time to transform selective bits of the data written to the memory with minimum latency, area and power penalty. MRT performs the inverse function of MWT and supplies correct data to the application. By exploiting content properties, we have designed transducers and address generating units (AGUs) for read and write ports.

Memory Read and Write Address Generating Units (Section 4.1 and 4.2): These AGUs generate addresses for writing to and reading from the memory. The addresses generated by the Write AGU are used to write a data set of a particular size to the memory in a pipelined fashion to non-contiguous memory locations. The read AGU translates the address requested by the application to the correct location of the data written by the Write AGU.

Memory Aging Controller (Section 4.4): It performs dynamic control of MRT, MWT and Read/Write AGUs. It activates different settings or configurations of these units at different time instances in order to balance the duty factor of each SRAM cell under varying run-time scenarios. The controller determines two key decisions: (i) at what time instant an aging balancing circuit should be activated; and (ii) on which SRAM cells aging balancing should be applied. These configurations have distinct power and aging-resiliency features. The selection of a particular configuration depends upon the power budget allocated to the ASCAR architecture during the system design, i.e., whether a user desires a highly reliable system at high power, or may afford compromising some lifetime to achieve a low-power consumption.

To the best of our knowledge, ASCAR is the first work towards low-power configurable aging-resilience architecture for memories. An important feature of ASCAR is that it is transparent to the application and the memory, as it does not need software and hardware adjustment, and no information about the adapted data is required. Further, ASCAR does not incur additional reads and writes to the SRAM, rather performs on-the-fly adaptations.

We have implemented our ASCAR architecture using a 65 nm technology and Synopsys synthesis tools. Our experimental results (Section 5) illustrate the benefit of our architecture by giving aging balancing results of different MWTs/MRTs, along with their power and area overhead.

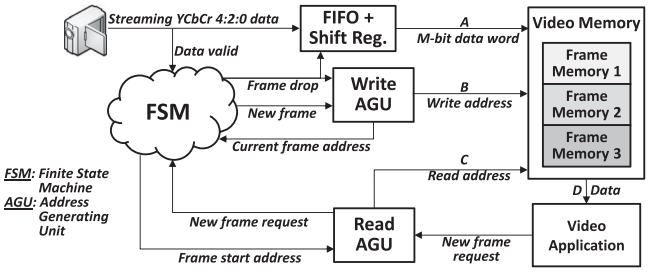


Fig. 2. Video memory management system for storing raw video samples.

2 Background and Preliminaries

2.1 Static Noise Margin

In this paper, we use the widely-used duty factor (Δ) and Static Noise Margin (SNM) to quantify the NBTI-aging for SRAM cells. The SNM denotes the resilience against noise, delay faults, and cell failures and directly affects the read stability [9], [15]. In case the SNM of an SRAM cell is reduced significantly as a result of NBTI, this cell is highly susceptible to read failures [9], [27]. According to the studies of [17], [18], the SNM degradation for the extreme cases (i.e., $\Delta = 0$ percent or 100 percent) is higher compared to that for the balanced case (i.e., $\Delta = 50$ percent). Considering this, we aim at reducing the SNM degradation through achieving a balanced duty factor (i.e., $\Delta \approx 40 - 60$ percent).

2.2 Video Terminology

A video is represented as a group of images (also called video frames) over time. An image is defined as a 2D-array of pixels with a width of W pixels and height of H pixels. In an 8-bit representation, a video sample requires one byte storage. In the RGB representation, each pixel contains 3 video samples, and an image requires $3 \times W \times H$ bytes storage. In YCbCr 4:2:0 representation, the size of one image is $1.5 \times W \times H$ bytes, for instance, one full-HD frame ($W = 1{,}920$ and $H = 1{,}080$ pixels) requires $\approx$2.97 Mbytes. The amount of memory required by a single video frame/image is called a "frame memory partition" throughout the text.

2.3 Baseline Video Memory Architecture

Fig. 2 illustrates our baseline memory architecture deployed in a camera-based image/video processing system. The camera captures videos in real time at a certain frame rate (typical values are 30 and 60 fps, frames per second). The video memory is composed of multiple frame memory partitions in order to simultaneously store multiple video frames. The streaming data, containing YCbCr 4:2:0 samples, is converted into words of M-bits ($M \geq 8$) using a combination of a FIFO and a shift register. The camera writes the video frame in a line-by-line order. The video memory can be a single port or a dual port SRAM. Dual port SRAMs are used to decrease the computational pressure on the application, by allowing parallel video frame writing (by the camera) and reading (by the application). Further, this requires more than one frame memory partition, such that, concurrent frame writing and reading is made possible. A frame memory partition is considered available for writing when the processing of a frame is completed and the application requests another frame to be processed.

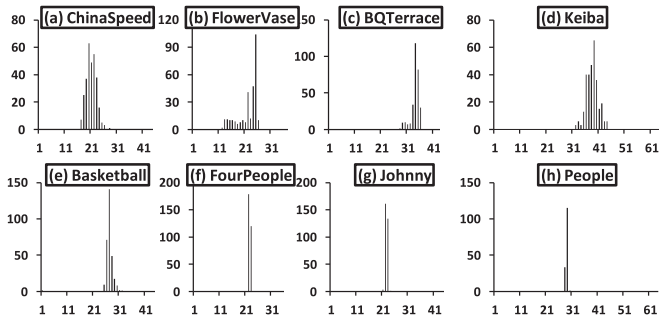


Fig. 3. Percentage histogram of SRAM memory overwritten with new bits by the video sequences given in Table 2. Here, x-axis presents the percentage of memory bits which are changed in the subsequent frame, while y-axis presents the number of times a certain percentage of change occurs for 300 continuous frames.

3 Aging Analysis for Different Circuits

In this section, we provide a detailed aging analysis, in terms of duty factor imbalance, for camera-based image/video processing architectures. Different test video sequences [28], [29] are used and issues related to image regions with distinct properties are highlighted (see details on video sequences and their properties in Section 5.1). This analysis provides guidelines for designing our configurable aging-resilient memory architecture.

3.1 Frame Memory Overwritten Bits

Fig. 3 shows the total percentage of bits (in form of a histogram) for a frame memory that are overwritten by a complementary bit of the new frame. As noticed, for some video sequences with low activity, this histogram is concentrated towards smaller percentages, which tells us that writing new frames will only marginally release stress on some 6T cells of the SRAM. Additionally, the histograms are not dispersed, showing that there is a significant correlation (in terms of texture and motion) between temporally neighboring frames. Therefore, the properties of the subsequent frames can be estimated from the history, such that it can be leveraged for efficient aging balancing. Specifically, we can predict the aging impact of the current and future video frame by analyzing the aging effects of the previous video frames.

3.2 Aging Imbalance in Least and Most Significant Bits

In Fig. 4, duty factor of a few selected bits of the luminance component of "FourPeople" test video sequence is plotted on the spatial scale in form of so-called "stressmaps". A balanced duty factor ($\Delta = 50$ percent or 0.5) is represented with a light greenish color (see the scale below the pictures). For duty factors heavily biased towards '0' and '1', we obtain a blue and a red colored distribution in the stressmap, respectively. It is noticed that the lower order bits (i.e., least significant bits) have a balanced duty factor, thus, the 6T SRAM cells storing these bits have regular relaxations and an extended lifetime. However, the higher order bits (i.e., most significant bits) have a highly biased duty factor, which causes an aging imbalance in the associated SRAM cells (i.e., one out of the two PMOS transistors is under increased stress). Different critical applications like surveillance and space exploration missions experience such long-duration static scenes. In summary, duty factors of different bits are not balanced and some bits age quicker than the others. In such cases, it becomes necessary to balance duty factor of each bit and to leverage the knowledge of bit location before applying an aging mitigation technique. Therefore, it is important to estimate the duty factor of each bit in addition to the balancing mechanism.

3.3 Aging of Static and Moving Regions

Less-frequently changing data (e.g., low complexity texture and large static backgrounds) will introduce the most amount of stress on the SRAM cells. This is also shown in Fig. 4 and Fig. 5 where the static background regions have a highly biased duty factor (see the stressmaps and boxplots for bit 6 and 7). However, the moving regions in the video frame have a marginally balanced duty factor. For example, the stressmaps for bit 6 and 7 have relatively balanced duty factor at the locations where people are moving. Therefore, the knowledge of less-frequently and more-frequently changing data can be exploited to distribute video samples in the memory, such that the transistors of each SRAM cell experience recovery effects.

3.4 Extending Baseline Architecture with Different Aging Balancing Circuits

In order to balance the duty factor, we extend the memory architecture of Fig. 2 with additional aging resiliency tools in the form of memory read transducer (MRT) and memory write transducer (MWT) connected to the memory read/write ports; see Fig. 5a. These transducers can be implemented using one of the three different aging balancing circuits as shown in Fig. 5b, c, d). As examples, we use inversion

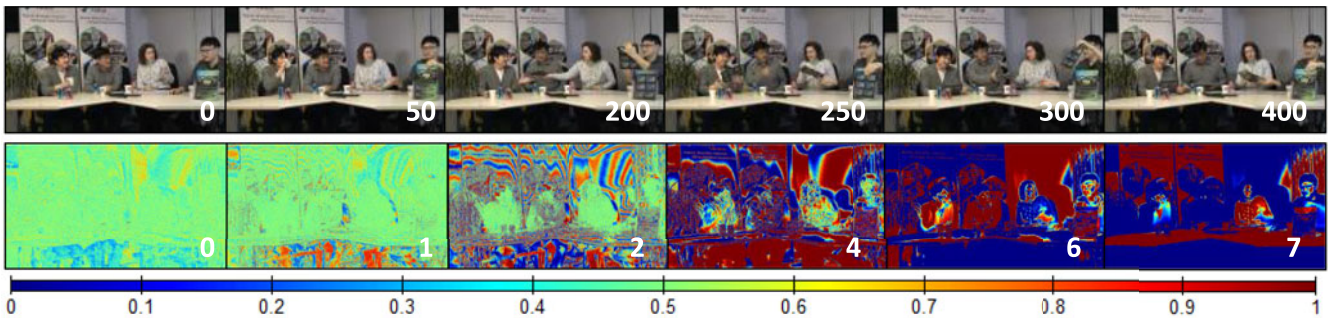


Fig. 4. (top) Video sequence "FourPeople" (1280 $\times$ 720) with frame numbers written and (bottom) Δ stressmaps for specific bits of the video sequence, by plotting duty factor (Δ) of the specific bit on spatial scale. The higher order bits have a biased Δ whereas the Δs of the lower order bits are self-balancing.

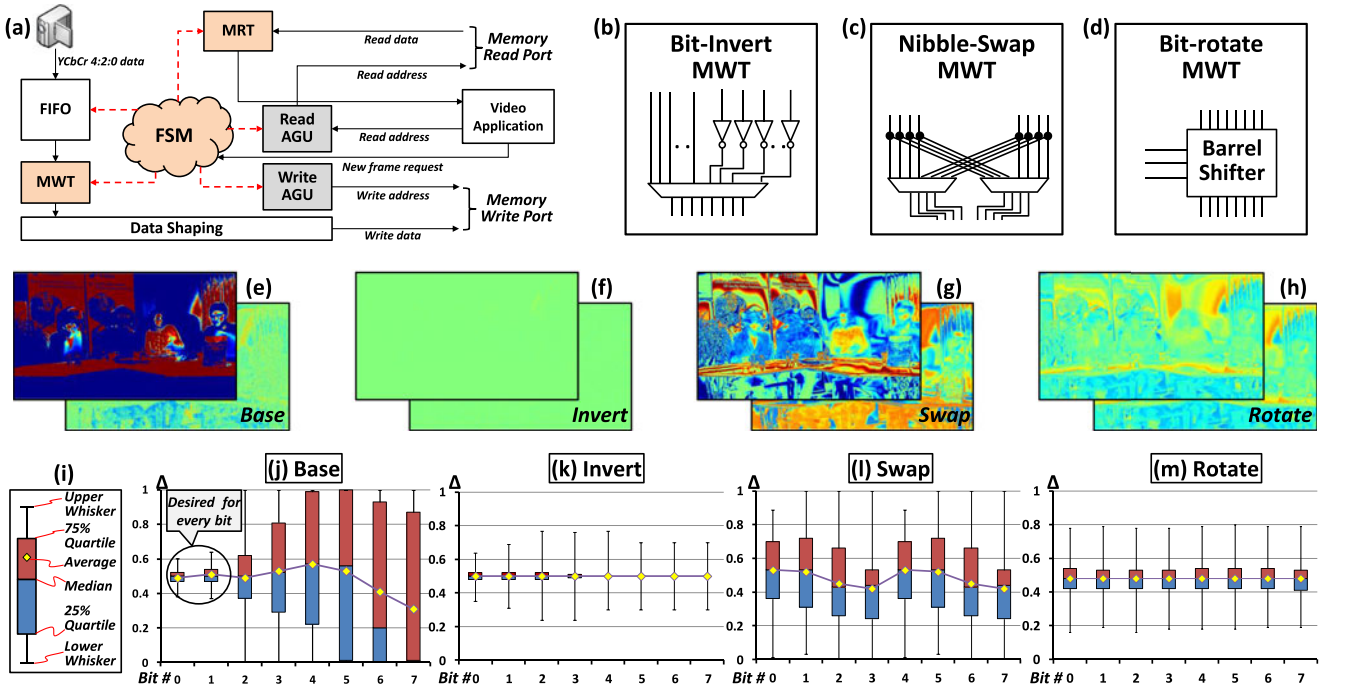


Fig. 5. (a) Insertion of aging resiliency components (i.e., Memory Read and Write Transducers, MWT and MRT) in video memory management of Fig. 2; (b) Bit-inversion MWT to invert all the bits of the video sample; (c) Nibble-swapping MWT to swap the most significant bits with the least significant bits; (d) Bit-rotation MWT to rotate bit locations with every frame; (e-h) Stressmaps for bit-7 (foreground) and bit-0 (background) for the above MWTs, with the bit-invert outperforming bit-swap and bit-rotate; (i) Box plot legend; (j-m) Box plots for the above MWTs, where all the bits are best-balanced for the bit-inversion.

(similar to [20]), nibble-swapping (similar to [18]) and bit-rotation (similar to [17]), which correspond to the state-of-the-art approaches to mitigate SRAM aging. For inversion and nibble-swapping, the bits of every second frame are inverted and swapped, respectively. In the bit-rotation circuit, the video sample bits are incrementally rotated by one with every frame, before writing to the frame memory.

3.5 Analysis of Different Aging Balancing Circuits

In Fig. 5e, f, g, h, the stressmaps of bit-7 for using no balancing (i.e., the base case) and the balancing circuits presented in Fig. 5b, c, d are plotted. As noticed, the duty factor for the inverter case is more balanced compared to the other balancing circuits. For convenience, the information about the duty factors for all bits is presented in the form of box plots, as shown in Fig. 5i, j, k, l. In the box plot, the distribution of duty factor of each bit is mapped to quartiles. In an ideal case, the spread of whiskers in the box plot should be minimal and the median of the box plot should be at '0.5'. The spread biased towards '0' indicates a higher number of "zeros" at the bit location and vice versa. As noticed for the base case, for the lower order bits (bits '0' and '1'), the spread of duty factor is limited and the median is closer to '0.5'. However, the spread of duty factor is large, and the median is not strictly '0.5' for the higher order bits (bits 3–7). This suggests that the higher order bits in a video sequence need more care and resiliency features embedded into the system must account for these bits. Furthermore, the lower order bits experience auto-balancing and the aging resiliency feature for these bits can be turned off to save power.

By using balancing circuits of Fig. 5b, c, d, the box plots significantly change. For inverters, we notice that the duty factor is nicely balanced for each bit and the spread is limited. The nibble-swapping introduces some improvement in balancing the duty factor, but it is not comparable to that of the inverter circuit. Moreover, the nibble-swapping also adversely impacts the aging of bit '0' and '1'. Bit-rotation fits in the middle of the inversion and nibble-swapping cases. Another important design parameter for MWTs is their energy and area overhead; refer to Fig. 12 in Section 5.2. As noticed, keeping the inverter ON for all bits all the time is not energy efficient. Therefore, the challenge is to design an adaptive, configurable controller to select the frame inversion rate and bits to invert at run time.

4 Memory Architecture with Configurable Aging Resilience

In this section, we detail our memory architecture with configurable aging resilience (ASCAR) composed of 6T SRAM cells. We assume that the memory capacity is sufficient enough to store a large chunk of data, e.g., multiple images/video frames. However, our configurable aging resiliency concepts are orthogonal to the number of memory ports and memory size. Fig. 6 illustrates the overall architecture of our aging-resilient memory. The basic operational steps are:

a. Data is written to a FIFO, controlled by the FIFO Controller, which also provides appropriate valid signals (e.g., data valid) to ASCAR.
b. The aging controller snoops the data from the data FIFO and generates appropriate control signals for the best aging resilience and power reduction trade-off (Section 4.4).
c. The signals generated by the aging control configure the memory write transducer (see details in

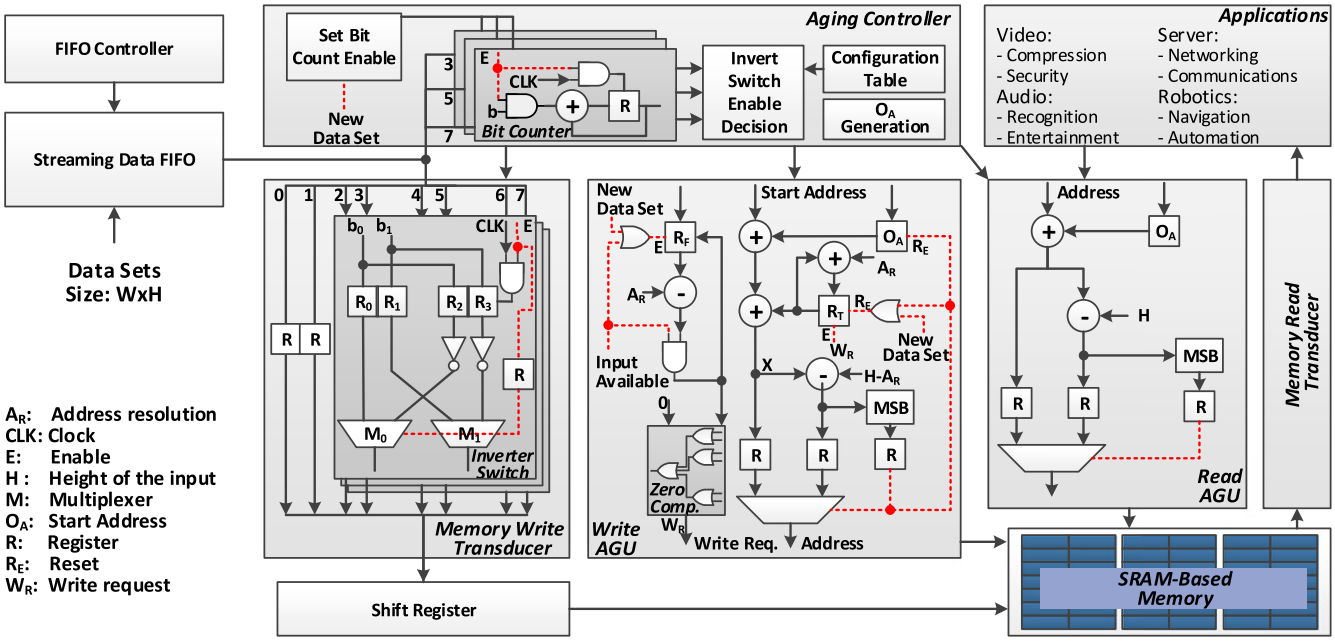


Fig. 6. Overview of ASCAR architecture. Notice the memory write transducer, Write AGU, Read AGU and memory read transducer connected to the SRAM. The aging controller configures these units by generating appropriate signals to adapt input data at read and write ports of the memory.

Section 4.1) to adapt input data samples before they are written to the memory. Specific bits of the data samples are selected for inversion by setting the appropriate enable signals of the Inverter Switches. In addition, the address of the data written to the memory is changed at run time using the Write AGU, to fully utilize the memory address space and introduce stress-relaxation at the SRAM cells holding less-frequently changing data samples (e.g., pixels of static background regions in an image).

d. Before the data is read by the application, it is re-adapted by the memory write transducer (an exact replica of MWT) and the logical address is appropriately converted to the physical address by the Read AGU.

In the following, we discuss the memory writing modules in detail. The working principles of the memory reading modules can easily be deduced as they perform the exact opposite operation of the writing modules.

4.1 Memory Write Transducer (MWT)

The MWT is used to invert specific bits of the raw data samples in order to toggle less-frequently changing data samples and release stress on the 6T SRAM cells storing these bits. The data bits are grouped in pairs and each bit-pair can be configured separately. The higher order bit-pairs (containing bits 2–7) are passed through controlled *Inverter Switches*. Fig. 6 shows that the first two bits (0 and 1) are not adapted. This is due to the fact that the two least significant bits always have the lowest degree of stress, due to a high degree of variation and hence a balanced duty factor (as shown in the case study of Fig. 5j). Therefore, ASCAR only specifies three Inverter Switches to control six most significant bits of a data sample. The input control lines (E) act as clock-gating signals to the registers R_2 and R_3 and as a "select" signal to the multiplexers M_0 and M_1. All the registers store the original bits (b_0 and b_1). The registers R_0 and R_1 are directly connected to the multiplexers, whereas R_2 and R_3 are inverted and fed to the multiplexers. For every bit-pair, five 1-bit registers, two inverters and two 1-bit multiplexers are required. For example, for 8-bit data samples, a total of 15 1-bit registers, six inverters and six 1-bit multiplexers are required.

If the control signal E of an invert switch is high, registers R_2 and R_3 latch the bits b_0 and b_1 and thus, the inputs and outputs of the inverters are updated. Both input bits are inverted and the inverted bits are generated at the output. If the "enable" signal is *low*, bits b_0 and b_1 will be forwarded to the shift register unaltered. Therefore, no dynamic energy will be consumed by R_2 and R_3 and the inverters. The signal E is controlled by the aging controller.

4.2 Memory Read Transducer

The MRT module is conceptually and architecturally same as the MWT. The MRT utilizes the principle of double inversion (i.e., $x = (x')'$) and inverts the data coming from the memory in the same manner as the MWT does. That is, the MWT will selectively invert the SRAM input, whereas MRT will selectively invert the SRAM output and therefore, generate the original data. This will result in the same image/video data being provided to the application as generated by the video camera. In short, the MRT construction is same as that of the MWT shown in Fig. 6.

4.3 Aging-Aware Address Generation Unit (AGU)

The Write AGU is responsible for selecting the appropriate memory partition (e.g., for writing an incoming video frame from a camera device) and generating addresses for writing the data words stored in the shift register. Selection of the appropriate frame memory partition for writing a complete data set follows a round-robin scheme. In addition, before overwriting a memory partition, it is required that the frame memory partition is no more required by the executing

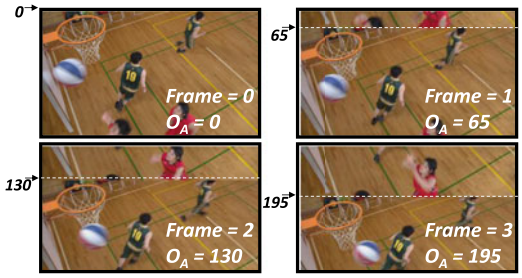


Fig. 7. Write AGU frame writing scheme with $O_A = 65$. The moving region (players and the basketball) are overwriting the static background (basketball court) with every frame.

application(s). An application signals this information by requesting the address of a new data set (see Section 2.3 for details). A signal ("New data-set request" in Fig. 6) prompts to deliver a new starting address to the Write AGU.

As discussed earlier, less-frequently changing data will introduce the most amount of stress on the 6T SRAM cells. If the most-frequently changing data words are identified, they can be distributed in memory in a spatial round-robin fashion. However, this requires further information from the application and additional analysis at run time, which is power- and area-wise inefficient. Therefore, ASCAR employs a simple approach for introducing spatial aging resiliency. For every new data set written to the memory partition, the data set is circularly shifted. This corresponds to changing the starting address of the data set. For every new data set, an offset in the starting address (O_A) is given (by the aging controller) to the Write AGU as the starting address of the data set. With every data set, O_A is accumulated and the starting address is shifted. This will ensure that the memory partitions containing less-frequently changing data are interchanged with the more-frequently changing data at regular intervals, thus relieving stress from the SRAM cells holding bits of the less-frequently changing data. Fig. 7 illustrates an example of a video frame written in the memory with $O_A = 65$. Similarly, the read AGU will add O_A to the logical address of the video frame from which data is requested by the application. In case the resultant physical address is greater than the height of the video frame, the physical address will be rolled back to the start of the frame-memory.

4.4 Aging Controller

As shown in Fig. 6, the aging controller configures the control signals of the MWT and supplies the start addresses of the data set (e.g., the starting address of a video frame) in the Write AGU. Fig. 8 shows the detailed flow of our aging controller for enabling Inverter Switches. Note that such a controller exists for each Inverter Switch, therefore, three such controllers are used in ASCAR. In order to generate the control signals for MWT, two decisions need to be made: (1) at what time instant the circuit should be activated, and (2) on which SRAM cells aging balancing should be applied.

4.4.1 Decision – 1: Activation of Aging Balancing Circuit at a Particular Time Instant

For this, the Inverter Switches are activated for a complete data set after a specific time period, i.e., a certain number of data sets are written without adaptation and are processed

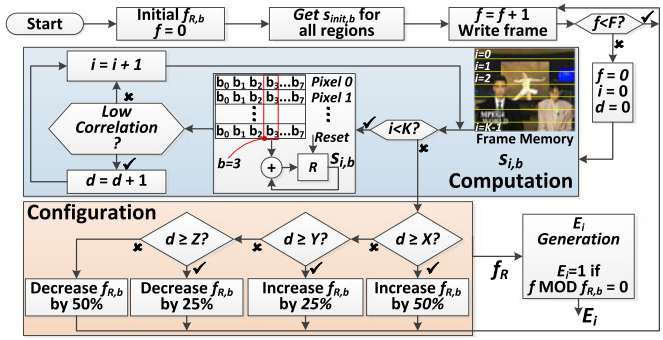


Fig. 8. Inverter Switch enabling decision logic. In ASCAR, there are three such circuits, one for each Inverter Switch.

by the application. For example, consider that a specific bit-plane of a data set $2i$ is stored without inversion ($E = 0$) and for the data set $2i + 1$, it is stored with its bits inverted ($E = 1$). This corresponds to the Data Adaptation Rate (f_R) equal to 1, i.e., every second data set is inverted. Formally, f_R denotes the number of data sets stored in the memory without adaptation for every inverted data set. In the definition above, a bit-plane is defined as the collection of the bits at the same bit location, in all the samples of a data set. In case two data sets are correlated (e.g., two neighboring video frames), it is expected that the inversion of data set $2i + 1$ will overwrite most of the bit locations of data set $2i$ with inverted bits. Thus, relieving stress on the SRAM cells and reducing the NBTI introduced aging. Note that there is a separate f_R for each Inverter Switch, and each Inverter Switch is independently activated. Additionally, f_R also plays an important role in deciding about the power consumption and aging rate of the SRAM memory. A high f_R will reduce the power consumption but will be less resilient to aging, and vice versa.

4.4.2 Decision – 2: Selecting SRAM Cells for Aging Balancing

At run time, the MWT enable signals can be turned ON or OFF, depending upon the expected aging and the power constraint of the system, which can be generated based upon f_R. When all the enable signals are inactive ($N = 0$), power penalty of ASCAR is the lowest because no inversion takes place. This is because no toggling activity occurs at the inputs of the inverters as the associated registers are unchanged. However, the rate of SRAM aging is the highest because we allow f_R number of data sets to be written with adaptation. Similarly, when all the enable signals are active ($N = 3$), SRAM aging rate is the lowest at the cost of the highest power penalty. When only one enable signal is active ($N = 1$), ASCAR inverts the two most significant bits (bit 6 and 7) as the SRAM cells storing these bits encounter the most stress (or aging). If the power configuration allows for two enable signals to be active (i.e., $N = 2$), the four most significant bits (bits 4–7) are inverted.

The parameter N depends upon f_R of all Inverter Switches. If f_R of all Inverter Switches is selected such that adaptation is active for all of them for the same data set, then $N = 3$. Otherwise, if two Inverter Switches are active for the same data set, then $N = 2$ etc. Selection of appropriate f_R is a control problem, which must be computed at run

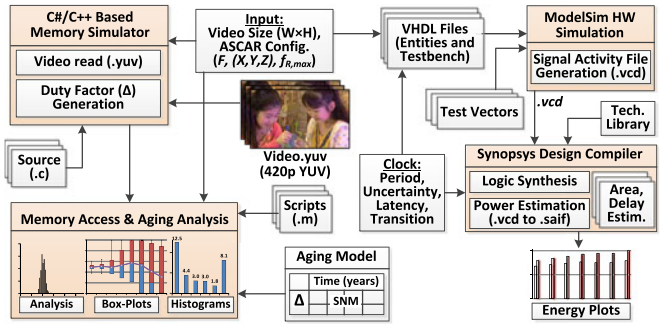


Fig. 9. Experimental setup showing different hardware/software components.

time by analyzing the characteristics of the input data (i.e., computation of duty factor for different bits). In ASCAR, we specify a simple and efficient microarchitectural technique to estimate duty factor online, as shown in Fig. 8. This circuit is used to determine f_R for a single bit-plane. A bit-plane is divided into K-parts, and the corresponding bits are accumulated for each part. For every part, if the number of 1s differs significantly from its previous stored value (depending upon the lower $s_{L,i,b}$ and upper $s_{H,i,b}$ threshold), the part has changed and a difference counter d is incremented. Afterwards, d is tested and f_R of the bit-plane is increased or decreased. The thresholds X, Y and Z ($X \geq Y \geq Z$) determine the change in f_R and can be set at design time.

However, computing the duty factor online incurs a power penalty. In ASCAR, a counter logic is used to activate the duty factor generation circuit. The set bit counter's register and the input bit to the adder are only updated if the enable signal is high. Thus, this will save dynamic power consumption. If online duty factor computations are more frequent (finer control with smaller F), the power consumption of ASCAR will be high and vice versa.

5 Experiments and Evaluation

5.1 Experimental Setup

Fig. 9 shows the details of the experimental setup used for obtaining the results, given in this paper.

5.1.1 Hardware Synthesis and Aging Estimation

We have implemented our ASCAR architecture in VHDL with different aging balancing circuits (inverter, bit swap, and bit rotate) as the basic building blocks and other architectural components like MWT, MRT, etc. The architecture is synthesized for a 65 nm TSMC technology [30] using Synopsys Design Compiler [31]. The gate-level simulations and functional verifications of the proposed architecture are performed using ModelSim [32], which is also used to generate the switching activity waveforms for power analysis. Similar to several state-of-the-art memory aging mitigation techniques, we can also report aging of SRAM using average stress (i.e., duty factor, Δ) instead of the SNM-degradation and that would be sufficient for the validation and comparisons. To report SNM, typically models like [17], [18] also rely on the duty factor to estimate SNM. In this paper, we use an example duty factor-to-SNM translation map based on the studies of [18] (see supplementary material, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TC.2016.2553025). However, our approach is orthogonal to any duty factor-to-SNM translation model/methodology because we basically target optimizing for the duty factor and switching activity. The studies of [17], [18] illustrate that the best SNM degradation for 6T SRAM cell after a lifetime of 7 years is at $\Delta = 0.5$, while the worst degradation is at $\Delta = \{0, 1\}$.

TABLE 1
Comparison Partners

MWT/MRT	Description
Base	No MWT or MRT used
Swap	Swap Lower and upper nibbles of the complete frame
Rotate	Rotate bits by 1 with every frame
$N = 1$	ASCAR, with only Inverter Switch 6–7 always ON
$N = 3$	ASCAR, with Inverter Switches 2–7 always ON
Controller	ASCAR, with Inverter Switches controlled
Invert	Invert all bits of the complete frame

5.1.2 Plotting Aging Results

For fast analysis and visualization of memory aging, we have developed a GUI-based tool in C# which is made open-source [33], [34]. This tool can accept user configurations like memory size, location of test data sets, total number of years the memory will be used etc. Using this tool, not only memory analysis and aging impacts can be visualized with ease, but also basic image and video processing applications (like filtering, color conversion, etc.) can be executed for aging analysis. This tool accepts any duty factor-to-SNM translation map, and automatically generates stressmaps, box plots, and SNM degradation histograms for different input data sets (examples are shown in our motivational analysis; Section 2).

5.1.3 Test Video Sequences

For our experiments, we employed various test video sequences recommended by the Joint Collaborative Team on Video Coding (JCT-VC) [35] which are available for download and testing [28], [29]. Some representative video sequences used in our experiments are presented in Table 2 along with their key attributes. These videos have diverse characteristics, and they can comprehensively represent various video capture scenarios.

5.2 Results and Comparison with State-of-the-Art

In Fig. 10, the percentage SNM degradation histograms are plotted for different MWTs (given in Table 1) with a single frame memory for different test video sequences. Except for the "base" and "controller" case, these histograms are generated with $f_R = 1$, i.e., every second frame is adapted. The base case (Fig. 10a) has a high distribution of SRAM cells with the worst SNM degradation. Majority of these cells are responsible for storing the higher order, low activity bits and thus exhibit the largest amount of stress. For the inverter MWT (Fig. 10g), almost all SRAM cells have the best possible degradation. Comparing with [21], [20], [22], the usage of bit-inverter MWT in the ASCAR architecture will have the same aging impact. However, the aging balancing in [21], [20], [22] are achieved by employing additional hardware and architectural changes to SRAM cells.

TABLE 2
Video Sequences and Their Attributes

Name	Basketball	Flowervase	Keiba	FourPeople	Johnny	ChinaSpeed	BQTerrace	Traffic	PeopleOnStreet
Attributes	832×480 Medium motion, no camera panning	832×480 Luminance changes, camera zooming in	832×480 Large motion, camera panning	1280×720 Very low motion, no camera panning	1280×720 Very low motion, no camera panning	1024×768 Large motion, no camera panning	1920×1080 Large static region, camera panning	2560×1600 Large static region, no camera panning	2560×1600 Medium motion, no camera panning
Resolution, Motion, Camera zooming/ panning, Frames									

This requires the designer to only use customized SRAM memories with added enhancements. Further, the leakage energy consumed and the area overhead of these SRAMs is much higher than ASCAR because each cell will have additional transistors associated with it.

The nibble-swap MWT (Fig. 5c) does not perform well as compared to the inverter and the rotator. For ASCAR without adaptive controller and Write AGU, and with only selected bits inverted ($N = 1$ and $N = 3$ in Fig. 10d, e), we notice that $N = 3$ has almost the same impact on aging as the inverter. This is because bits 0 and 1 are self-balancing themselves while bits 2–7 are adapted (see the box plot in Fig. 5 for bits 0 and 1). However, $N = 1$ will only invert bits 6 and 7, whereas from Fig. 5, we notice that bits 4 and 5 may also have highly biased Δ, which contributes to a high SNM degradation. Still, $N = 1$ considerably balances Δ, as compared to the base and swap case. For testing MWT with the adaptive controller (Fig. 10f), in this experiment we have chosen (X,Y,Z) = (0.75K,0.50K,0.25K) where K = 5 is the number of parts in which a bitplane is divided (see Fig. 8). The run-time adaptation of f_R by the proposed controller is shown in Fig. 11 for different test sequences. For highly static sequences like "FourPeople", f_R of each Inverter Switch is lowered to the minimum possible f_R, in order to adaptively counter the aging that such a sequence will induce. For high activity sequences or sequences with camera panning (like "BQTerrace"), f_R of each bit is increased, as it is not required to aggressively invert the frames. In addition, the aging impacts of the proposed techniques with multiple frame memories are almost identical.

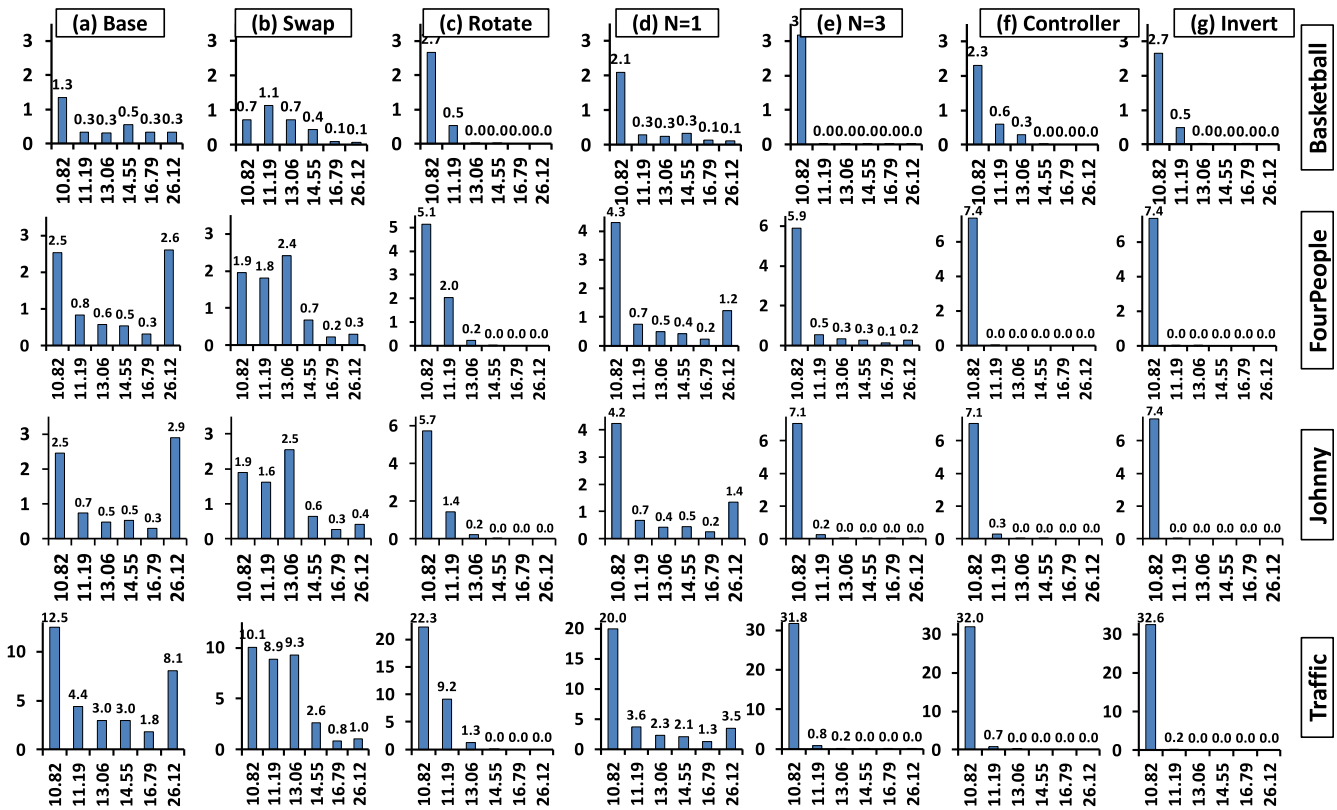


Fig. 10. Histogram of SNM degradation for a single frame memory partition with different comparison partners as given in Table 1. Values on x-axis denote the percentage SNM degradation. The y-axis denotes total number of bits in millions, and the actual value is written on top of each bar. The best aging balancing is achieved when the histogram is crowded towards left (i.e., near 10.82 percent). For all (except the base and adaptive controller case), every second frame is adapted ($f_R = 1$).

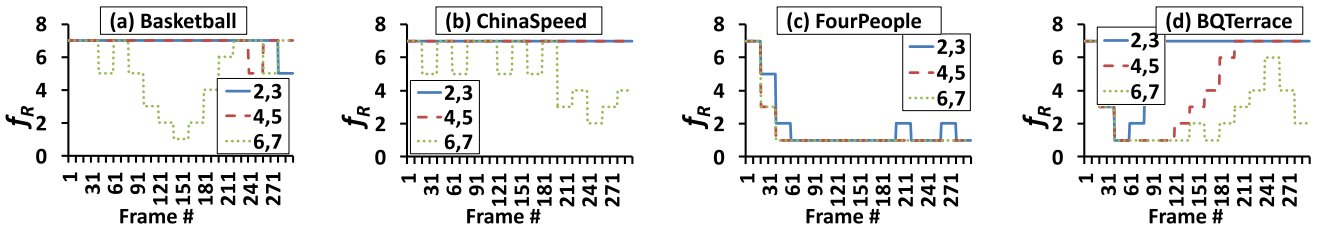


Fig. 11. Dynamic adaptation of f_R by the proposed aging controller with K = 5, (X,Y,Z) = (0.75K,0.50K,0.25K), F = 20. Minimum $f_R = 1$, maximum $f_R = 8.s_L = (1-\varepsilon) \times s_{init}$ and $s_H = (1+\varepsilon) \times s_{init}$ where ε for bits (3,5,7) = (1/64, 1/32, 1/8). Results are presented for four sequences: (a) Basketball, (b) ChinaSpeed, (c) FourPeople, and (d) BQTerrace.

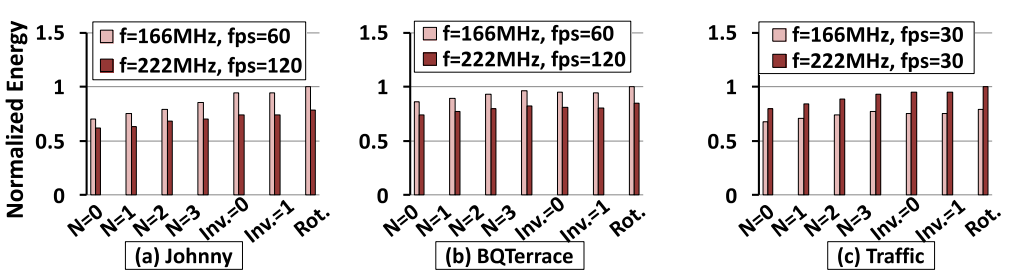
The energy consumption per frame and area of different MWTs, for different running frequencies and *fps* are given in Fig. 12. This data is generated by annotating the input to MWTs using ModelSim simulation. The signal annotations were then queried by the Synopsys Design Compiler to estimate average signal activity on each pin and generate the leakage and dynamic power. As noticed, the proposed MWT with no invert switch active ($N = 0$) consumes the smallest amount of energy, whereas the bit-rotate MWT consumes the largest amount of energy and area. From Fig. 10, we also notice that aging balancing achieved by the bit-inverter and our proposed adaptive bit-inversion can easily surpass the performance of the bit-rotate MWT. Therefore, it is reasonable to use inverters in MWTs for aging resiliency instead of bit-swapping and bit-rotation logic. Further, when the technique presented in [18] is applied to SRAM memories, it requires testing for leading zeros, read/write of infrequently accessed memory addresses and additional information storage. On the contrary, ASCAR does not require such tests because it adaptively generates addresses to span the whole memory space in a circular fashion to introduce activity in low-activity cells. Compared to [19], ASCAR does not require additional reads and writes to the SRAM memory, which itself consumes high dynamic energy.

Further, depending upon the application scenario and the allowable energy budget, ASCAR enables the application designer to select the best f_R and N configuration, suitable for the application. For example, from Fig. 12, a designer can achieve up to ~15 percent energy savings by turning off all the invert switches at the cost of SRAM aging. Therefore, a tradeoff between energy and SRAM aging can be established to select the best configuration.

5.3 Area, Energy, and Delay/Latency Overhead Analysis

The energy overhead comparison in Fig. 12a, b, c is pessimistic considering state-of-the-art approaches are integrated within the proposed architecture and MWT in a fixed way, thus showing generality and a wide-applicability of our architectural extensions. Originally, these state-of-the-art approaches consume additional power compared to our MRT/MWT approach due to the following factors:

1) *Customized SRAM Cells with Additional Transistors:* Employing inverter at each SRAM cell level (as done by the recovery boosting technique [20]) results in a significant area and power overhead. For instance, a full-HD YUV420 frame based memory design will require $1.5 \times 1920 \times 1280 \times 8 = 29.4912 \times 10^6$ additional inverters, while a Quad-HD design will require $4 \times 29.4912 \times 10^6 = 117.9648 \times 10^6$ additional inverters, which is several orders of magnitude more than the requirements of our anti-aging controller. An increased area also leads to a significant leakage and dynamic power. However, our technique requires only 15 1-bit registers, six inverters and six 1-bit multiplexers for 8-bit data samples in the MWT/MRT. This corresponds to an insignificant increase in the leakage and dynamic power. Besides this, when also considering the controller state machine and AGUs, the total area overhead of the complete aging-balancing circuity and controller is 10.273 K GEs for a 65 nm technology. Note that a baseline design and state-of-the-art will also require AGUs and some kind of state machine.
2) *Inversion through Re-Reading and Re-Writing:* The techniques in [17], [18], [19], [22] read the data from the SRAM, modify it, and then write it back to the SRAM. This will incur a significant latency and energy overhead as modification is done for each byte again after writing. These state-of-the-art techniques must wait until data is written to the SRAM (e.g., by the camera), then read and modify the data once again, and then write it back to the memory. Such architecture and processing model for the data written to the SRAM (at run time) introduce delay in acquiring read/write



MWT/MRT	Area (in Cells)
ASCAR	516.23
Bit-Invert	520.91
Bit-Rotate	685.4

(d)

Fig. 12. (a-c) Energy consumed per frame of MWTs and MRTs at different frequency and fps configurations. (d) Total area (in cells) for different MWTs/MRTs. Inv. = 0 denotes that the Invert MWT/MRT is inactive while Inv. = 1 denotes active Invert MWT/MRT. Rot. denotes Rotate MWT/MRT.

ports of the SRAM memory, therefore, delaying the content delivery to the application. Their overhead aggravates for increasing video resolutions. For instance, considering 1 cycle read/write latency of a 32-bit word, processing a full-HD YUV420 frame requires $2 \times 1.5 \times 1920 \times 1280/4 = 1.84 \times 10^6$ additional cycles (additional 0.44 mJ energy consumption per frame), while a Quad-HD design will require $4 \times 1.84 \times 10^6 = 7.36 \times 10^6$ additional cycles (additional 1.76 mJ per frame).

5.3.1 Reduced Delay and Latency of Our Technique

In contrast to state-of-the-art, our technique does not require any additional reads/writes, and performs aging-balancing on the fly using MRT/MWT and controller parameters. The slight latency increase due to MWT/MRT is even hidden by enabling the bypass mode controlled by the anti-aging controller. To further hide the latency overhead in aging-balancing, our ASCAR architecture is implemented in a pipelined design that embeds the MWTs as an additional pipeline stage between the streaming data FIFO and shift registers. There is a single cycle latency in delivering the data from the FIFO to the shift register due to the associated registers of the MWTs. However, since the video data is written in a pipelined fashion, this latency only occurs for the first output. That is, out of W × H samples (e.g., $1920 \times 1080 = 2073600$ samples), only the first sample will encounter a latency of 1 cycle when writing to the shift-registers (additional energy overhead is below 1 nJ per frame). This is a negligible overhead. Further, our synthesis showed that MWT does not form the critical path. Moreover, the read and write AGUs can result in a single cycle delay for each memory access. However, the control state-machine writes the addresses for reading and writing in a FIFO, which can deliver the access requests in pipelined fashion. This way, the problem of single cycle penalty per access is resolved.

In summary, unlike state-of-the-art, our architecture incurs a negligible performance and energy overhead due to on-the-fly aging control and our pipelined design. Moreover, our controller overhead is minimal compared to the frame memory area and power.

5.4 Discussion on HCI and Switching Activity Analysis

HCI-induced aging is caused by high energy (hot) carriers injected inside the gate oxide, resulting in interface traps that lead to a threshold voltage shift [36]:

$$\Delta V_{th} = A_{HCI} \times \alpha \times f \times e^{\frac{V_{dd}-V_{th}}{t_{ox}E_1}} \times t^{0.5},$$

where A_{HCI} is a constant depending on the aging rate, t is the time, α is the switching activity factor, f is the frequency, V_{th} is the threshold voltage and V_{dd} is the supply voltage, t_{ox} is the oxide thickness and E_1 is a constant. It should be noted that the degradation mainly affects NMOS transistors and that HCI is directly proportional to the switching activity [36]. Therefore, in the following, the duty factor (relevant for NBTI) and the switching activity (relevant for HCI) are discussed for two different video sequences for different aging-balancing circuits integrated in our architecture.

Different aging balancing circuits have different impact on the duty factor and toggling characteristics of the 6T SRAM cells. The distributions are shown in Fig. 13 as per-bit box-plots. The baseline video system incurs high NBTI-induced aging, but with limited HCI-induced aging. Same is the case with swap also due to high amount of switching. The Rotate MWT balances the duty factor but with increased maximum duty factor value in the box plot: However, it also elevates the toggling rate of the most significant bits (as shown by the concentration around 0.5–0.6 and a high maximum value), and hence the corresponding cells have a higher HCI-induced aging. The duty factor is considerably balanced by the inverter circuit thus encountering low NBTI-induced aging. However, it also increases the toggling rate due to aggressive switching of every bit for every data packet, which will result in a higher HCI-induced aging.

In contrast to the state-of-the-art fixed techniques, our anti-aging controller can adapt the spatial and temporal granularity of applying the aging-balancing techniques. Therefore, it can provide improved distribution profiles for duty factor and switching activity across different bits.

5.5 Applicability and Generalization

In general, the overall design methodology and architecture are orthogonal to the type of application and the low-level aging models. For instance, different audio processing and data-parallel applications will also benefit from this. However, this may require several design optimizations to get the best power-efficient aging-mitigation design, like (1) in addition to inverter, some rotation or swapping hardware blocks can also be integrated in the Memory Read and Write Transducers; (2) The values of controller parameters to adapt the application of aging-balancing also need to be re-evaluated. This will require an application- and content-aware analysis, as we have performed for the camera-based application in Section 3. To illustrate the varying duty factor behavior of a non-frame-based application, we have tested audio samples storage, as discussed below.

In Fig. 14, a 16 KHz, 16-bit Linear Pulse Code Modulated (LPCM) audio signal and the duty factor box-plot of the memory used for storing this signal are shown. Note, without using any aging balancing circuit, the duty factor of all the bits will already be balanced for most of the cases, which is visible from the concentrated spread of the boxplot. This is due to the fact that the audio signal swings around 0, and the number of negatives (with most significant bits storing 1s) and the number of positives (with most significant bits storing 0s) is similar, unlike the video data. Moreover, the temporal correlation of audio data is low, meaning that it is highly probable that the new audio data which overwrites the previous one will have different characteristics. However, this is not the case with video data, which will have high temporal correlation (for example, the background region, which will be static in many subsequent frames), leading to biased duty factors and higher stress on the 6T SRAM cells.

5.6 Sensitivity Analysis

As discussed in Section 4.4, the aging balancing achieved by ASCAR can be controlled by f_R, N and O_A configurable

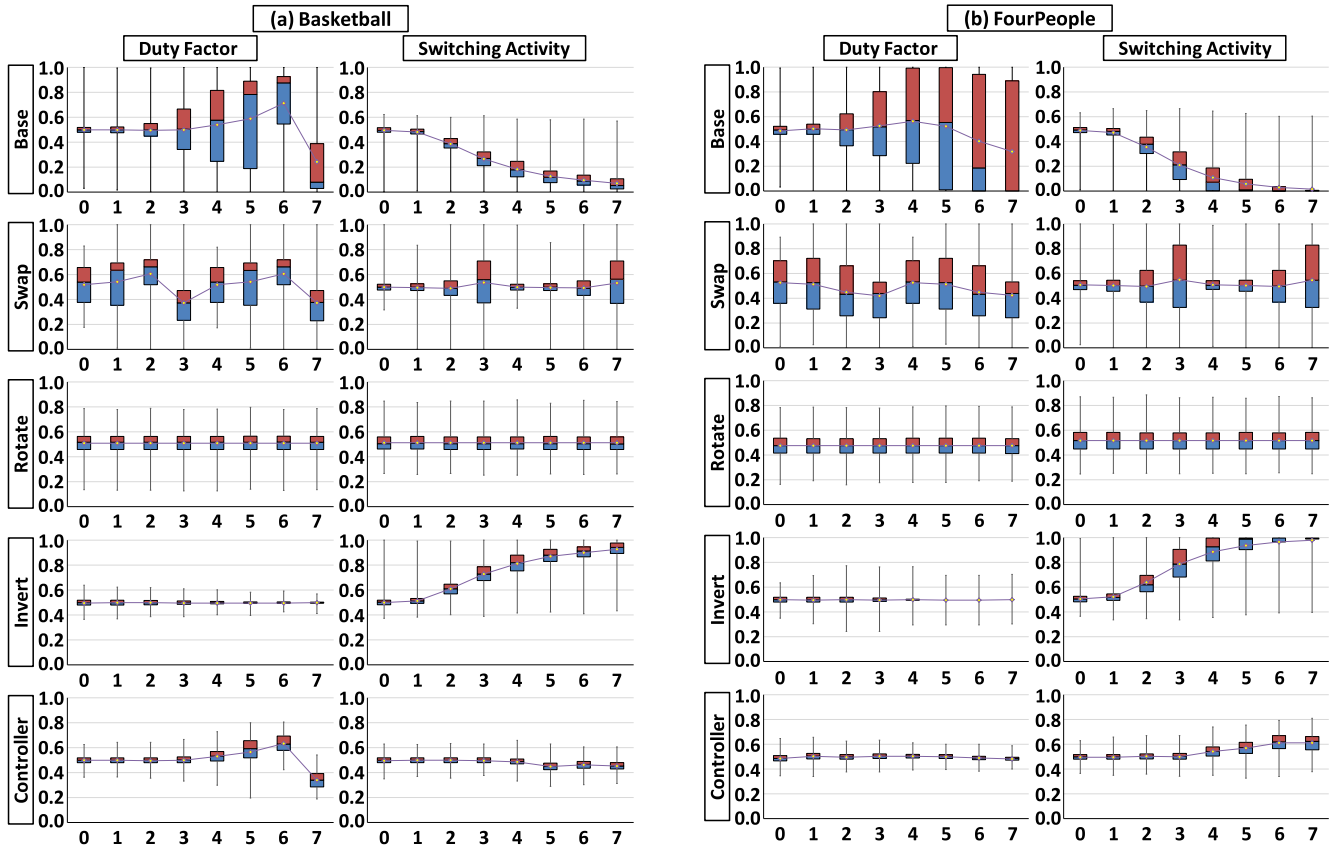


Fig. 13. Duty factor and toggling statistics for (a) "Basketball" video sequence and (b) "Four People" video sequence, for different MWTs. The x-axis on all the graphs presents the bit-planes of the memory. The y-axis on the duty factor plots shows the average duty factor per bit-cell. The y-axis for the toggle graphs shows the box-plot of average toggling rate for each bit-cell, i.e., the average number of times a write to a cell results in a bit-flip. For best results, the duty factor box-plots should be crowded around 0.5, while the toggle box-plots should be crowded towards 0.

parameters. In this section, we evaluate the impact of f_R, N and O_A on aging of video memories. For concise representation of the SNM degradation histograms in Fig. 10, we devise an aggregate memory aging metric (μ).

$$\mu = \frac{\sum_{\forall bins}\left[\left(V_{bin} - V_{bin,\min}\right) \times N_{bin}\right]}{\left(V_{bin,\max} - V_{bin,\min}\right) \times N_{samples}}. \quad (1)$$

In this equation, V_{bin} is the value of a bin of the histogram (on x-axis, SNM degradation), N_{bin} is the number of values in the bin (on y-axis, total number of bits for the particular SNM degradation), and $V_{bin,min}$ and $V_{bin,max}$ are the minimum and maximum values of the bins, respectively. $N_{samples}$ corresponds to the total number of samples in the histogram (in our case, this equals $W \times H \times 8$, i.e., the size of the frame memory). Thus, if the number of values in the bins is closer to $V_{bin,min}$ (10.82 in our case), we would expect μ closer to 0. Otherwise, μ will be closer to 1 (with $V_{bin,max} = 26.12$). Therefore, an aging resiliency technique should reduce μ as much as possible. Further, in our calculation of μ, we consider the distance between the local degradation and the least possible degradation.

The aging analysis of SRAM cells for different video sequences is given in Table 3. The column "Base" denotes the amount of aging without using any MWT, i.e., without applying any aging balancing techniques. Since different video sequences lead to different amount of stress as a result of varying distribution of "zeros" and "ones", the aging imbalance results in undesirable varying degradation of different SRAM cells. Sequences with large static structures in video frames (e.g., "Johnny") introduce the most amount of stress on the SRAM cells because of static sample values. These sequences are common for video security- and communication-applications. Camera panning and zooming sequences (like "BQTerrace", "FlowerVase" and "Keiba") usually have a low aging impact on SRAM cells. Largely static video sequences exhibit high aging due to less frequently changing data values (e.g., "Basketball", "ChinaSpeed", "FourPeople", and "Johnny").

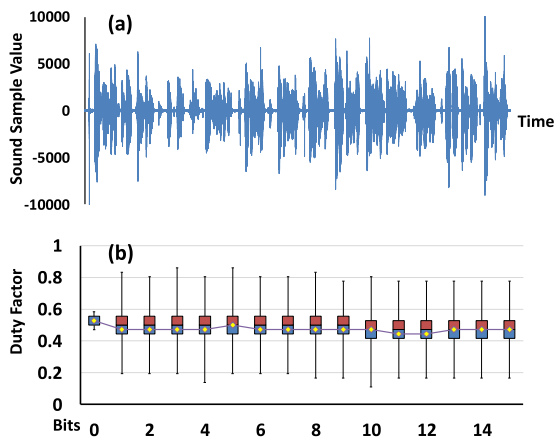


Fig. 14. (a) A 16-bit/sample 16 KHz audio signal; (b) Aging profile as box plot.

TABLE 3
Aging Parameter (μ in 10^{-2}) for Different Video Sequences, with No Inversion, No Controller

Seq.	Base	$f_R = 1$			$N = 0$, $f_R = \infty$		
		Invert	Swap	Rotate	$O_A = 17$	$O_A = 41$	$O_A = 61$
Basketball	20.46	0.50	10.97	0.43	4.55	4.47	4.45
BQTerrace	5.44	0.00	1.75	0.06	1.13	1.11	1.11
ChinaSpeed	32.57	0.00	17.01	9.34	8.96	8.91	8.92
FlowerVase	5.80	0.00	2.45	0.91	3.64	3.64	3.63
FourPeople	40.13	0.00	12.92	1.08	2.02	2.02	2.04
Johnny	43.70	0.00	14.66	0.89	16.51	16.51	16.52
Keiba	4.85	0.00	1.65	0.00	2.12	2.02	1.99
People	12.45	0.01	4.83	0.30	1.46	1.37	1.36
Traffic	30.66	0.01	10.88	1.28	2.70	2.71	2.63

The aging parameter after employing the MWTs given in Fig. 10b, c, d are also tabulated. For a comparison with these MWTs, results for using $N = 0$ (no Inverter Switch active) and only using the Write AGU to circularly write the frames in the frame memory in ASCAR are also tabulated. This is achieved by having $O_A \neq 0$. Using three different values for O_A, we notice considerable aging balancing achieved by only adapting the start addresses of the video frames. Specifically, largely static sequences get the most benefit. However, an interesting observation for the "Johnny" sequence can be made where we notice low aging improvement as compared to a sequence with similar aging profile (i.e., "FourPeople"). This is due to the fact that the static regions in the video frames are similar throughout the height of the frame and the video samples of the new frame which overwrite the video samples of the previous frame having similar values, thus, not contributing to stress relaxation. Hence, inversion is a better option in such a case. Further, O_A adaptation results in better aging resiliency as compared to nibble-swapping, with negligible power penalty. Note that O_A is chosen as a prime number to make the cycle of offset to be as large as possible.

The impact of parameters f_R and N on aging for different video sequences is given in Table 4. Note that increasing f_R causes more frames to be inserted between two adapted frames. However, for static sequences like "Johnny", aging is accelerated due to an increase in the duty factor bias. Sequences with camera panning, zooming and frequent scene changes exhibit lesser sensitivity to changing f_R, mainly because of the video memory overwritten continuously with changed video samples. For example, the sequence "Keiba" and "BQTerrace" exhibit lower sensitivity to increasing f_R. Similarly, introducing more inverters by enabling the control signals of the Inverter Switches in the MWT will largely balance the aging of video memory. For slow moving, static sequences like "Johnny" and "Traffic", $N = 3$ results in a considerably better aging profile compared to $N = 2$ or 1. Highly dynamic sequences can still achieve the same aging with $N = 2$ or $N = 1$.

Our experiments also reveal that using multiple frame memories result in almost the same aging balancing with the proposed technique. This is because frames are highly correlated, and instead of always overwriting the frame memory with the next frame, writing the second or third frame results in nearly the same statistics.

TABLE 4
Aging Parameter (μ in 10^{-2}) for Different Video Sequences, with $O_A = 0$, No Controller

Seq.	Base	$f_R = 1$			$f_R = 3$			$f_R = 7$		
		N = 1	N = 2	N = 3	N = 1	N = 2	N = 3	N = 1	N = 2	N=3
Keiba	4.85	0.12	0.00	0.00	0.88	0.78	0.77	1.64	1.57	1.57
Johnny	43.70	22.21	6.92	0.20	25.40	12.46	6.85	27.66	16.62	12.27
BQTerrace	5.44	0.95	0.04	0.00	1.72	0.94	0.90	2.95	2.48	2.46
Traffic	30.66	15.00	4.16	0.18	17.42	8.37	5.09	19.41	12.03	9.68

6 Conclusions

In order to mitigate the aging in SRAM memories, we propose an application-specific microarchitectural technique called ASCAR. It adapts the input and output video data of an SRAM-based memory at run time, in order to improve the duty factor and toggle rate of each SRAM cell. ASCAR performs spatio-temporal control on the toggling activity for data to be written in the SRAM cells. To enable this, we examined the aging profile and overhead (power and area) of different aging balancing circuits. Our ASCAR architecture can be used to mitigate both NBTI and HCI aging by adapting the controller parameters while reducing the power overhead of aging mitigation.

Acknowledgments

This work was carried out in Chair for Embedded Systems (CES), Karlsruhe Institute of Technology (KIT), and was partly supported by the German Research Foundation (DFG) as part of the priority program "Dependable Embedded Systems" (SPP 1500 - spp1500.itec.kit.edu, [8]).

References

[1] S. Udayakumaran, A. Dominguez, and R. Barua, "Dynamic allocation for scratch-pad memory using compile-time decisions," *ACM Trans. Embedded Comput. Syst.*, vol. 5, no. 2, pp. 472–511, 2006.

[2] M. Shafique, B. Zatt, F. L. Walter, S. Bampi, and J. Henkel, "Adaptive power management of on-chip video memory for multiview video coding," in *Proc. Des. Autom. Conf.*, 2012, pp. 866–875.

[3] F. Sampaio, M. Shafique, B. Zatt, S. Bampi, and J. Henkel, "dSVM: Energy-efficient distributed scratchpad video memory architecture for the next-generation high efficiency video coding," in *Proc. Des. Autom. Test Eur. Conf.*, 2014, pp. 1–6.

[4] F. Sampaio, M. Shafique, B. Zatt, S. Bampi, and J. Henkel, "Energy-efficient architecture for advanced video memory," in *Proc. Int. Conf. Comput.-Aided Des.*, 2014, pp. 132–139.

[5] J. Henkel, L. Bauer, N. Dutt, P. Gupta, S. Nassif, M. Shafique, M. Tahoori, and N. Wehn, "Reliable on-chip systems in the nano-era: Lessons learnt and future trends," presented at the Design Automation Conf., Austin, TX, USA, 2013.

[6] J. Henkel, L. Bauer, H. Zhang, S. Rehman, and M. Shafique, "Multi-layer dependability: From microarchitecture to application level," in *Proc. Des. Autom. Conf.*, 2014, pp. 1–6.

[7] S. Rehman, M. Shafique, F. Kriebel, and J. Henkel, "Reliable software for unreliable hardware: Embedded code generation aiming at reliability," in *Proc. 9th Int. Conf. Hardware-Softw. Codes. Syst. Synthesis*, 2011, pp. 237–246.

[8] J. Henkel, et al., "Design and architectures for dependable embedded systems," in *Proc. 9th Int. Conf. Hardware/Softw. Codes. Syst. Synthesis*, 2011, pp. 69–78.

[9] K. Kang, H. Kufluoglu, K. Roy, and M. A. Alam, "Impact of negative-bias temperature instability in nanoscale SRAM array: Modeling and analysis," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 26, no. 10, pp. 1770–1781, Oct. 2007.

[10] A. Ricketts, J. Singh, K. Ramakrishnan, N. Vijaykrishnan, and D. K. Pradhan, "Investigating the impact of NBTI on different power saving cache strategies," in *Proc. Des. Autom. Test Eur.*, 2010, pp. 592– 597.
[11] A. Calimera, E. Macii, and M. Poncino, "NBTI-aware power gating for concurrent leakage and aging optimization," in *Proc. Int. Symp. Low Power Electron. Des.*, 2009, pp. 127–132.
[12] J. B. Velamala, K. Sutaria, T. Sato, and Y. Cao, "Physics matters: Statistical aging prediction under trapping/detrapping," in *Proc. 49th ACM /EDAC/IEEE Des. Autom. Conf.*, 2012, pp. 139–144.
[13] J. Abella, X. Vera, and A. González, "Penelope: The NBTI-aware processor," in *Proc. Int. Symp. Microarchitecture*, 2007, pp. 85–96.
[14] S. Drapatz, "Parametric reliability of 6T-SRAM core cell arrays," Ph.D. dissertation, TU München, München, Germany, 2011.
[15] K. Kang, S. Gangwal, S. P. Park, and K. Roy, "NBTI induced performance degradation in logic and memory circuits: How effectively can we approach a reliability solution," in *Proc. Asia South Pacific Des. Autom. Conf.*, 2008, pp. 726–731.
[16] S. Kumar, C. Kim, and S. Sapatnekar, "Impact of NBTI on SRAM read stability and design for reliability," in *Proc. Int. Symp. Quality Electron. Des.*, 2006, pp. 210–218.
[17] S. Kothawade, K. Chakraborty, and S. Roy, "Analysis and mitigation of NBTI aging in register file: An end-to-end approach," in *Proc. 12th Int. Symp. Quality Electron. Des.*, 2011, pp. 1–7.
[18] H. Amrouch, T. Ebi, and J. Henkel, "Stress balancing to mitigate NBTI effects in register files," in *Proc. 43rd Ann. IEEE/IFIP Int. Conf. Dependable Syst. Netw.*, 2013, pp. 1–10.
[19] S. Wang, T. Jin, C. Zheng, and G. Duan, "Low power aging-aware register file design by duty cycle balancing," in *Proc. Des. Autom. Test Eur.*, 2012, pp. 546–549.
[20] T. Siddiqua and S. Gurumurthi, "Recovery boosting: A technique to enhance NBTI recovery in SRAM arrays," in *Proc. Annu. Symp. VLSI*, 2010, pp. 393–398.
[21] J. Shin, V. Zyuban, P. Bose, and T. Pinkston, "A proactive wearout recovery approach for exploiting microarchitectural redundancy to extend cache SRAM lifetime," in *Proc. Int. Symp. Comput. Archit.*, 2008, pp. 353–362.
[22] A. Sil, S. Ghosh, N. Gogineni, and M. Bayoumi, "A novel high write speed, low power, read-SNM-Free 6T SRAM Cell," in *Proc. 51st Midwest Symp. Circuits Syst.*, 2008, pp. 771–774.
[23] J. Abella, X. Vera, O. Unsal, and A. Gonzalez, "NBTI-resilient memory cells with NAND gates," US Patent US20080084732 A1, 2008.
[24] S. Wang, G. Duan, C. Zheng, and T. Jin, "Combating NBTI-induced aging in data caches," in *Proc. 23rd ACM Int. Conf. Great Lakes Symp. VLSI*, 2013, pp. 215–220.
[25] E. Gunadi, A. A. Sinkar, N. S. Kim, and M. H. Lipasti, "Combating aging with the colt duty cycle equalizer," in *Proc. 43rd Annu. IEEE/ACM Int. Symp. Microarchitecture*, 2010, pp. 103–114.
[26] A. Calimera, M. Loghi, E. Macii, and M. Poncino, "Partitioned cache architectures for reduced NBTI-induced aging," in *Proc. Des. Autom. Test Eur.*, 2011, pp. 1–6.
[27] K. Agarwal and S. R. Nassif, "Statistical analysis of SRAM cell stability," in *Proc. 43rd ACM/IEEE Des. Autom. Conf.*, 2006, pp. 57–62.
[28] F. Bossen, "Common test conditions," *Joint Collaborative Team Video Coding (JCT-VC) Doc. I1100*, 2012.
[29] "Common YUV 4:2:0 test sequences," [Online]. Available: https://media.xiph.org/video/derf. [Accessed 16 Aug. 2015]
[30] "Taiwan semiconductor manufacturing company limited," *TSMC*, [Online]. Available: http://www.tsmc.com./ [Accessed 7 Oct. 2015]
[31] "Design compiler," *Synopsys*, [Online]. Available: http://www.synopsys.com/Tools/Implementation/RTLSynthesis/DesignCompiler./ [Accessed 7 Oct. 2015]
[32] "ModelSim - leading simulation and debugging," *Mentor Graph.*, [Online]. Available: http://www.mentor.com/products/fpga/model./ [Accessed 7 Oct. 2015]
[33] M. Shafique, M. U. K. Khan, O. Tuefek, and J. Henkel, "CES Free Software - EnAAM," CES, KIT. Available: http://ces.itec.kit.edu/1023.php./ https://sourceforge.net/projects/enaam/ [Accessed 5 Oct. 2015]
[34] M. Shafique, M. U. K. Khan, O. Tuefek, and J. Henkel, "EnAAM: Energy-efficient anti-aging for on-chip video memories," in *Proc. 52nd Des. Autom. Conf.*, 2015, pp. 1–6.
[35] "Joint collaborative team on video coding (JCT-VC)," *ITU*, [Online]. Available: http://www.itu.int/en/ITU-T/studygroups/2013–2016/16/Pages/video/jctvc.aspx. [Accessed 7 Oct. 2015]
[36] A. Tiwari and J. Torrellas, "Facelift: Hiding and slowing down aging in multicores," in *Proc. Int. Symp. Microarchitecture*, 2008, pp. 129–140.

Muhammad Shafique received the PhD degree in computer science from the Karlsruhe Institute of Technology (KIT), Germany, in 2011. He is currently a Research Group Leader at the Chair for Embedded Systems, KIT. He has more than ten years of research and development experience in power-/performance-efficient embedded systems in leading industrial and research organizations. He holds one U.S. patent. His current research interests include design and architectures for embedded systems with focus on low power and reliability. He received 2015 ACM/SIGDA Outstanding New Faculty Award, six gold medals, the CODES + ISSS 2015, 2014 and 2011 Best Paper Awards, AHS 2011 Best Paper Award, DATE 2008 Best Paper Award, DAC 2014 Designer Track Poster Award, ICCAD 2010 Best Paper Nomination, several HiPEAC Paper Awards, Best Master Thesis Award, and SS'14 Best Lecturer Award. He is the TPC co-Chair of ESTIMedia 2015 and 2016, and has served on the TPC of several IEEE/ACM conferences like ICCAD, DATE, CASES, and ASPDAC. He is a member of the IEEE.

Muhammad Usman Karim Khan received the BSc degree in electrical engineering from the University of Engineering and Technology (UET), Peshawar, Pakistan in 2007. He received the MSc degree in electrical engineering from the Karlsruhe Institute of Technology (KIT), Germany and Politecnico di Torino (PdT), Italy as an Erasmus Mundus scholar in 2011. He is currently with IBM Research and Development, Böblingen, Germany. He has been actively involved in embedded architectures, DSPs and FPGAs, power and resource efficient designs, wireless transmission, and image/video processing. He was a recipient of a gold medal from UET, medals from the state for his academic successes, the best Ph.D. poster award in DATE'15 and the best designer track poster award in DAC'14. He has published several papers in prestigious conferences, and contributed as an external reviewer for major design automation conferences and transactions.

Jörg Henkel received the PhD degree from Braunschweig University with "Summa cum Laude". He is currently with the Karlsruhe Institute of Technology (KIT), Germany, where he is directing the Chair for Embedded Systems (CES). Before, he was a Senior Research Staff Member at NEC Laboratories in Princeton, NJ. He has/is organizing various embedded systems and low power ACM/IEEE conferences/symposia as General Chair and Program Chair and was a Guest Editor on these topics in various Journals like the IEEE Computer Magazine. He was Program Chair of CODES'01, RSP'02, ISLPED'06, SIPS'08, CASES'09, Estimedia'11, VLSI Design'12, ICCAD'12, PATMOS'13, NOCS'14 and served as General Chair for CODES'02, ISLPED'09, Estimedia'12, ICCAD'13 and ESWeek'16. He is/has been a steering committee member of major conferences in the embedded systems field like at ICCAD, ESWeek, ISLPED, Codes + ISSS, CASES and is/has been an editorial board member of various journals like the IEEE TVLSI, IEEE TCAD, IEEE TMSCS, ACM TCPS, JOLPE etc. In recent years, He has given around ten keynotes at various international conferences primarily with focus on embedded systems dependability. He has given full/half-day tutorials at leading conferences like DAC, ICCAD, DATE, etc. He received the 2008 DATE Best Paper Award, the 2009 IEEE/ACM William J. McCalla ICCAD Best Paper Award, the Codes + ISSS 2015, 2014, and 2011 Best Paper Awards, and the MaXentric Technologies AHS 2011 Best Paper Award as well as the DATE 2013 Best IP Award and the DAC 2014 Designer Track Best Poster Award. He is the Chairman of the IEEE Computer Society, Germany Section, and was the Editor-in-Chief of the ACM Transactions on Embedded Computing Systems (ACM TECS) for two consecutive terms. He is an initiator and the coordinator of the German Research Foundation's (DFG) program on "Dependable Embedded Systems" (SPP 1500). He is the site coordinator (Karlsruhe site) of the Three- University Collaborative Research Center on "Invasive Computing" (DFG TR89). He is the Editor-in-Chief of the IEEE Design & Test Magazine since January 2016. He holds ten US patent and is a Fellow of the IEEE.