

Buffering Implications for the Design Space of Streaming MEMS Storage

Mohammed G. Khatib and Leon Abelmann
University of Twente, The Netherlands
{M.G.Khatib, L.Abelmann}@utwente.nl

Abstract—Emerging nanotechnology-based systems encounter new non-functional requirements. This work addresses MEMS storage, an emerging technology that promises ultrahigh density and energy-efficient storage devices. We study the buffering requirement of MEMS storage in streaming applications. We show that capacity and lifetime of a MEMS device dictate the buffer size most of the time. Our study shows that trading off 10% of the optimal energy saving of a MEMS device reduces its buffer capacity by up to three orders of magnitude.

Index Terms—Secondary storage, energy efficiency, layout.

I. INTRODUCTION

The demand for energy-efficient systems is pressing in mobile and server environments alike. This work leverages MEMS storage to offer energy-efficient yet high-capacity mobile streaming systems. MEMS storage enjoys ultrahigh densities ($> 1 \text{ Tb/in}^2$) [1], a small footprint (41 mm^2), and high shock resistance (10 G m/s^2), to name a few. Based on the well-established MEMS fabrication techniques, MEMS storage also promises low-cost storage devices.

Problem – Buffering is typically used in mechanical devices to allow the moving medium to halt and thus save energy. MEMS storage requires a very small streaming buffer due to its short latency. The small buffer, however, conflicts with the formatted storage granularity and causes increased stress of mechanical parts. This can lead to a large loss in formatted capacity and short lifetime. We tackle the buffering problem to address these new non-functional requirements.

MEMS storage requires large capacity and long lifetime besides energy-efficiency. This work holistically studies these requirements. We make the following contributions:

- We model the energy consumption, capacity and lifetime of MEMS storage as a function of the buffer size.
- We implement inverse functions to map from design requirements to a design decision: buffer size.
- We analyse buffering implications and explore the design-space of MEMS storage.
- We show that, indeed, the emerging non-functional requirements dictate the buffer size most of the time.

Background on MEMS storage can be found in the literature [2] and is omitted for space reasons. Section II explains how to reduce MEMS energy in streaming systems. Section III details buffering implications for MEMS storage. Section IV explores the buffer design space. Section V concludes.

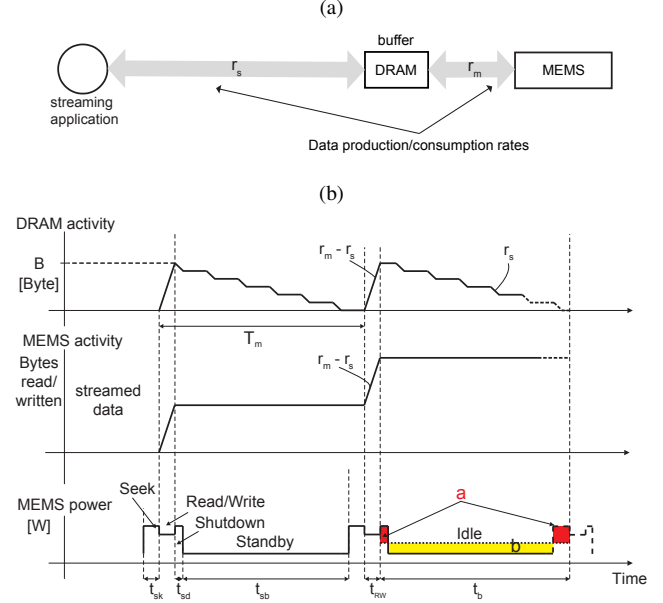


Fig. 1: A streaming architecture composed of a MEMS storage device and a DRAM buffer. The figure shows the activity in a refill cycle (T_m).

II. ENERGY-EFFICIENT STREAMING

A mechanical storage device must shut down to save energy. During shutdown data are staged in a buffer as in Figure 1a, which gets flushed periodically once the device is awakened. Buffering of data results in *shaping* streaming traffic from/to the storage device. Figure 1b shows the shaping activity in a MEMS–DRAM architecture. Here, the MEMS storage device serves as a backing store, whereas the DRAM is its buffer. The activity of MEMS and DRAM are presented for two consecutive *refill cycles*. To stream from it, MEMS starts every cycle of T_m time units filling the DRAM at a net rate $r_m - r_s$; While filling at r_m , the buffer empties at r_s . Every cycle MEMS seeks before the actual refill. After the refill, MEMS shuts down immediately and remains in standby to save energy. DRAM delivers data at a rate of r_s directly to the decoder of the streaming application.

III. BUFFERING IMPLICATIONS FOR MEMS

This section devises energy consumption, capacity, and lifetime as a function of the buffer size.

A. MEMS Energy Consumption

1) *Break-Even Buffer*: To save energy by shutting down, a MEMS storage device must stay in shutdown for a long

period. The employed buffer (B) must be proportionally large to make area “b” larger than area “a” in Figure 1b; Khatib [2] derives the buffer capacity for a mechanical device.

The break-even buffer, which makes area “b” equal to “a”, for MEMS storage is noticeably small. For streaming rates in the range 32 – 4096 kbps, the break-even buffer ranges from 0.07 kB to 8.87 kB. In contrast, the break-even buffer of a 1.8-inch disk drive for the same streaming range is 0.08–9.29 MB, a difference of three orders of magnitude.

2) *Energy consumption*: The per-bit energy consumption of a MEMS storage device in one refill cycle (T_m) is [2]:

$$E_m(B) = \frac{t_{oh}}{B} \cdot (P_{oh} - P_{sb}) + \frac{t_{rw}}{B} \cdot (P_{rw} - P_{sb}) + \frac{T_m}{B} \cdot P_{sb}, \quad (1)$$

where

$$T_m = \frac{B}{r_m - r_s} \cdot \frac{r_m}{r_s}, \quad t_{rw} = \frac{B}{r_m - r_s}$$

$$t_{oh} = t_{sk} + t_{sd}, \quad E_{oh} = E_{sk} + E_{sd}, \quad P_{oh} = \frac{E_{oh}}{t_{oh}}.$$

E_{oh} and t_{oh} are the overhead energy respectively time incurred due to shutdown. Figure 1b visualizes the previous parameters. From Equation (1) we find that the per-bit energy is inversely proportional to the buffer size. The first term, the overhead, scales inversely with the buffer size, whereas the second and third terms remain constant.

B. MEMS Capacity

Any storage device, such as MEMS, adds bookkeeping bits to user data for reliability and accessibility.

1) *Error-Correction Bits*: A storage device stores user data in sectors. In addition to user data, a sector stores error-correction code (ECC) to increase the reliability of user data. The amount of ECC data depends on, among others, the sector size and the type of errors the device is prone to. In disk drives, for example, ECC is one-tenth the size of the user data stored per sector [3]. In line with available figures from the IBM MEMS device, we assume that ECC data (S_{ecc}) in a MEMS storage device is one-eighth the user data (S_u):

$$S_{ecc} = \left\lceil \frac{S_u}{8} \right\rceil$$

2) *Synchronisation Bits*: Mechanical storage devices store a few synchronisation bits between each two consecutive subsectors on the media [2]. These bits (1) allow for data buffering before writing a subsector, and (2) keep the clock of the read channel running, so that the subsector can be fully read/written [4, Chapter 18, pages 650 – 652]. In the Disk drive, synchronisation bits have a small influence on the capacity, since they occur per sector. Contrarily, in a MEMS storage device synchronisation bits occur per subsector, thus potentially claiming a large amount of the available capacity. We assume three synchronisation bits per subsector, which amounts to a period of 30 μ s that is sufficient for processing.

From the above, striping a sector across K probes results in a subsector of size (s):

$$s = \left\lceil \frac{S_u + S_{ecc}}{K} \right\rceil + 3. \quad (2)$$

The effective sector size stored on the device (S) is:

$$S = K \cdot s, \quad (3)$$

and the capacity utilisation becomes:

$$u(S_u) = \frac{S_u}{S}. \quad (4)$$

For example, the capacity utilisation of our MEMS storage device (see Table I) tops with 88%, approximately 106 GB out of 120 GB effective user capacity.

From Equations (2) and (3), we find that the subsector size is crucial. As it increases fewer synchronisation bits are incurred per sector, which increases the effective capacity. We can increase the subsector size, if we format with large sectors (S), increasing user data (S_u) stored together. Thus, the streaming buffer must not be arbitrarily small:

$$B \geq S_u.$$

Phrased differently, if we allow large streaming buffer, then a MEMS storage device can be formatted with a large sector to utilise most of its capacity for user data. This is particularly crucial in streaming applications, where dealing with large files is the norm.

C. MEMS Lifetime

Saving energy in a streaming setting requires that a MEMS storage device seeks and shuts down repeatedly. This might affect the lifetime of its springs. Probes wear out when writing, which limits their lifetime. A failure of the springs or the probes claims the lifetime of the device. In the following, we delve into the lifetime of each component and their interplay with the streaming buffer size.

1) *Springs Lifetime*: The lifetime of the springs might be a concern in streaming applications. A MEMS device seeks and shuts down repeatedly. As a result, the springs are extended and compressed frequently for virtually their full range. Therefore, springs must be designed to sustain a large number of seek and shutdown cycles, called duty-cycle rating (D_{sp}). Mathematically, the springs lifetime of a streaming MEMS in years is:

$$L_{sp}(B) = \frac{D_{sp}}{\frac{T \cdot r_s}{B}} = \frac{D_{sp} \cdot B}{T \cdot r_s}, \quad (5)$$

where T is the total seconds played back per year, r_s is the streaming bit rate, and B is the buffer size. The term $\frac{T \cdot r_s}{B}$ is the number of refills per year.

Compared to the disk drive, MEMS lifetime is more crucial in a streaming setting, because the buffer of a MEMS storage device is three orders of magnitude smaller than that of a 1.8-inch disk drive. As a consequence, to meet a typical lifetime of today’s Disk-based systems, the springs must have a duty-cycle rating that is three orders of magnitude larger than that of the disk drive. That is, about 10^8 cycles compared to the 10^5 rating of the 1.8-inch disk drive.

Although a rating of 10^8 sounds large, it is very attainable in MEMS storage. Based on discussion with device technologists,

TABLE I: Settings of the modelled MEMS storage device and the exercised workload.

Parameter	Setting	Unit
Probe-array size	64×64	probe
Active probes	1024	probe
Probe-field area	100×100	μm^2
Capacity	120	GB
Per-probe data rate	100	kbps
Fast/Slow seek time	2	ms
Shutdown time	1	ms
I/O overhead time	2	ms
Read/Write power	316	mW
Fast/Slow Seek power	672	mW
Standby power	5	mW
Idle power	120	mW
Shutdown power	672	mW
Probe write cycles	100 & 200	cycles
Springs duty cycles	10^8 & 10^{12}	cycles
Hours per day	8	hours
Writes percentage	40%	
Best-effort fraction	5%	
Stream bit rate	32 – 4096	kbps

the following two reasons can be provided. Firstly, unlike the disk drive, a MEMS storage device has no rubbing surfaces that can wear each other [2], so that the motion components exhibit no tribological issues. Secondly, springs produced from silicon exhibit very small fatigue for stresses under 1 GPa [5], resulting in a cycle rating higher than 10^{12} . We take 10^8 for electroplated nickel [6] and 10^{12} for silicon as low- and high-end springs, respectively.

2) *Probes Lifetime*: A MEMS storage device is susceptible to probe wear. A probe wears out over time due to high temperatures and mechanical stress taking place when writing bits. In a streaming setting with write traffic probes lifetime is a concern. Through communication with device technologists [7], probes can overwrite the device 100 times (probes write-cycle rating D_{sp}) before they start functioning unreliably. Finding wear-resistant materials for probes is under heavy investigation [8]. Assuming a perfect balance in writing across all probes, probes lifetime (L_{pb}) is:

$$L_{pb}(B) = \frac{C \cdot D_{pb}}{w \cdot S \cdot \frac{T \cdot r_s}{B}} = \frac{C \cdot D_{pb} \cdot B}{w \cdot S \cdot T \cdot r_s}, \quad (6)$$

where C is the device capacity, w is the fraction of writes, and S is the sector size (Equation (3)).

The lifetime of a MEMS storage device is limited by either the springs or the probes failing first:

$$L = \min(L_{sp}, L_{pb}).$$

IV. DESIGN-SPACE EXPLORATION

A. Experimental Methodology

We take IBM MEMS prototype as our reference point [1]. The relevant parameters are summarised in Table I. As for the streaming buffer, we assume a DRAM buffer. We include

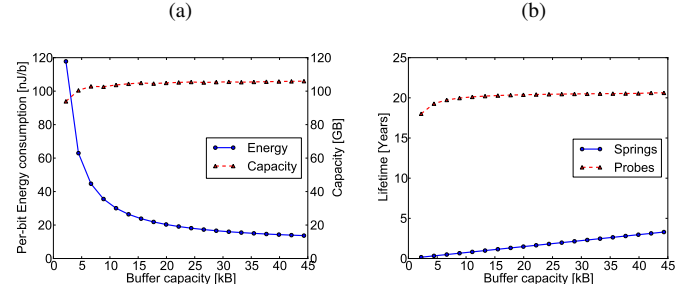


Fig. 2: Trends of the per-bit energy consumption, capacity, and lifetime for the modelled MEMS storage device.

energy to retain and to access data from the DRAM. The DRAM model is taken from Micron [9]. We found that DRAM energy consumption is negligible due to its tiny size, thanks to the small overheads of MEMS storage.

As for the workload, we assume a playback of eight hours every day all year round. Further, we assume that 40% of the streaming traffic writes to the device, for example recording video on the device. To account for Operating System and File System activities, we assume that 5% of the refill cycle (T_m) is spent in honouring best-effort requests, and take that into account in dimensioning the buffer.

B. Buffering Influence

This section quantifies the influence of buffer size on the energy consumption, capacity, and lifetime. For a streaming bit rate of 1024 kbps, we scale the buffer capacity in the range 1 – 20 times the break-even buffer and plot the resultant per-bit energy consumption and capacity utilisation in Figure 2a. We also plot the resultant springs (at 10^8 rating) and probes lifetime in Figure 2b.

Figure 2a confirms the merit of buffering to reduce the per-bit energy consumption (Equation (1)). The DRAM energy is present, but is negligible. The figure shows diminishing returns as the buffer increases beyond 20 kB, since the fraction of the overhead energy, saved by buffering, diminishes.

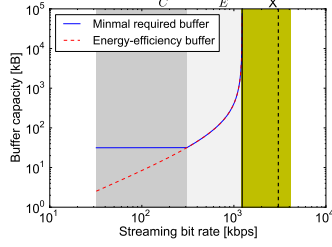
The capacity increases for a large buffer, because formatting with large granularity becomes possible. As a result, the device bits increasingly store user data instead of synchronisation data. Beyond 7 kB the capacity increase saturates.

Figure 2b shows that probes lifetime follows the capacity trend. This is explained by Equation (6), which shows that probes lifetime is inversely proportional to the storage granularity (S). In general, probes lifetime increases as the buffer increases, since probes write cycles are increasingly spent on writing user bits instead of synchronisation bits.

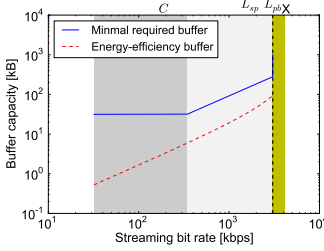
Springs lifetime is proportional to the buffer size; the larger the buffer, the longer the lifetime. Figure 2b shows that the springs at 10^8 limit the device lifetime to just 4 years, and more buffer is required for longer lifetime if the rating is 10^8 .

Observe from both figures that although energy efficiency is achieved with a 20 kB buffer, about 90 kB is required to attain a 7-year lifetime in order to extend the springs lifetime. In a higher streaming rate (>1024 kbps) the probes would limit the lifetime, whereas in a lower one the capacity loss dominates.

(a) ($E = 80\%$, $C = 88\%$, $L = 7$),
 $D_{pb} = 100$, $D_{sp} = 10^8$



(b) (70%, 88%, 7), 100, 10^8



(c) (70%, 88%, 7), 200, 10^{12}

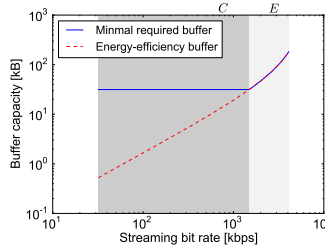


Fig. 3: Buffer size versus the streaming bit rate for our modelled MEMS storage device.

It is rarely were buffer requirement dominates due to energy-efficiency. We shall see this in the next section.

C. Buffer Dimensioning

This section asks a design question: *What is the buffer size required to achieve a design goal of certain energy saving (E), capacity (C), and lifetime (L)?* The answer could either be a quantitative result of the buffer size, or a statement of infeasible design point.

We implemented the inverse functions of Equations (1), (4) (assuming $S_u = B$), (5) and (6). This section studies the buffer size required to achieve design goals of different combinations of E, C, and L. E is provided as the energy saving relative to an always-on device. C is provided as capacity utilisation. L is provided as an absolute value of the (desired) lifetime of the device in years.

We vary the streaming rate in the range 32 – 4096 kbps and compute the required buffer size to achieve the design goal: ($E = 80\%$, $C = 88\%$, $L = 7$). These targets are the attainable maxima, and $L = 7$ is the typical lifetime of a mobile device. Figure 3a plots the buffer size versus the streaming rate for this goal. The figure also shows the ranges of the streaming rates where either energy (E), capacity (C), springs lifetime (L_{sp}), or probes lifetime (L_{pb}) dictates the buffer size.

From Figure 3a, we see that the capacity dominates for up to 300 kbps. After that the buffer size increases exponentially due to energy (see Figure 2a). At slightly above 1000 kbps the 80% energy-efficiency reaches its limit: no buffer size exists to achieve an 80% saving for higher bit rates. This is marked by the vertical solid line. In other words, the goal ($E = 80\%$, $C = 88\%$, $L = 7$) is infeasible for streaming rates above 1000 kbps, marked by the green range with an “X” on top.

Suppose the designer opts for ($E = 70\%$, $C = 88\%$, $L = 7$) then. Figure 3b plots the buffer scaling. Compared to the

previous goal, this goal is feasible for more streaming rates: up to 1500 kbps. Observe that energy has no word on buffer size for this goal. Capacity and then springs lifetime dominate for the entire range. Further, the figure shows a difference of 1 to 2 orders of magnitude between the required buffer and the energy-efficiency buffer. Also, the buffer size drops three orders of magnitude compared to Figure 3a. The probes lifetime is reached at around 1500 kbps, marked by the vertical dashed line.

To avoid the probes lifetime limit, designers need to increase probes durability. Suppose that technologists increase probes write cycles from 100 to 200 cycles. Further, silicon springs are used of a 10^{12} rating. Figure 3c shows an updated version of Figure 3b where capacity prevails followed by energy. With a 10^{12} rating, springs lifetime is not a concern and therefore disappears from the figure.

If the designer opts for lower capacity, say $C = 85\%$, the domination range of C decreases (no figure is shown for space reasons). Lifetime dominates temporarily before energy takes over as in Figure 3a. Note that a large buffer size has virtually no influence on probes lifetime as Figure 2b shows.

One noteworthy point is that the *system-wide* impact of the energy saving between 70% and 80% on the storage device might be negligible. On the contrary, the buffer size differs three orders of magnitude (Figure 3a versus Figure 3b), so that 70% might well be preferable.

V. CONCLUSIONS

This work investigated the buffer size of MEMS storage in streaming applications. Contrary to other storage devices, we showed that capacity and lifetime of a MEMS device are crucial when dimensioning the buffer and must be concerned. Energy-efficiency has a lesser influence. We explored MEMS design space and showed that enhancement in probes lifetime is essentially needed.

ACKNOWLEDGEMENT

This research is supported by the Technology Foundation STW, applied science division of NWO and the technology programme of the Ministry of Economic Affairs under project number TES.06369.

REFERENCES

- [1] M. A. Lantz *et al.*, “A Vibration Resistant Nanopositioner for Mobile Parallel-Probe Storage Applications,” *Journal of Microelectromechanical Systems*, vol. 16, no. 1, pp. 130–139, February 2007.
- [2] M. G. Khatib, “MEMS-Based Storage Devices - Integration in Energy-Constrained Mobile Systems,” Ph.D. dissertation, University of Twente, June 2009.
- [3] S. McCarthy *et al.*, “Larger Disk Blocks or Not?” Tech. Rep., 2002.
- [4] B. Jacob *et al.*, *Memory Systems (Cache, DRAM, Disk)*. Morgan Kaufmann, 2008.
- [5] C. L. Muhlstein *et al.*, “High-cycle fatigue of single-crystal silicon thin films,” *J. Microelectromech. Syst.*, vol. 10, no. 4, pp. 593–600, 2001.
- [6] Y. Yang *et al.*, “Fatigue of liga ni micro-electro-mechanical system thin films,” *Metallurgical and Materials Trans. A*, vol. 38, pp. 2340–2348, 2007.
- [7] H. Pozidis, “Wear in probe storage,” Personal Communication, July 2008.
- [8] H. Bhaskaran *et al.*, “Nanoscale PtSi tips for conducting probe technologies,” *IEEE Trans. on Nanotechnology*, vol. 8, no. 1, pp. 128–131, January 2009.
- [9] J. W. Janzen, *TN-46-03 - Calculating Memory System Power for DDR*, Micron Technology, Inc., October 2003.