Handheld System Energy Reduction by OS-Driven Refresh

Vasily G. Moshnyaga1, Hoa Vo2, Glenn Reinman2, and Miodrag Potkonjak2

1 Department of Electronics Engineering and Computer Science, Fukuoka University
8-19-1, Nanakuma, Jonan-ku, Fukuoka 814-0180, Japan
2 Computer Science Department, University of California, Los Angeles,
3532 Boelter Hall, Los Angeles, CA 90095-1596

Abstract. Emerging portable devices relay on DRAM/flash memory system to satisfy requirements on fast and large data storage and low-energy consumption. This paper presents a novel approach to reduce energy of memory system, which unlike others, lowers energy of refresh operation in DRAM. The approach is based on two key ideas: (1) DRAM-based flash cache that keeps dirty pages to reduce the number of accesses to flash memory; and (2) OS-controlled page allocation/aging to stop the refresh operations in banks, whose pages are clean and not accessed for a long time. Simulations show that by using this technique we can decrease the overall energy consumption of DRAM/flash memory on video applications by 8-26% while reducing the DRAM refresh energy by 59-74%.

1 Introduction

1.1 Motivation

Modern battery-operated handheld devices, such as mobile phones, incorporate 128Mb-512Mb SDRAM as main storage and 256MB-1GB solid-state flash memory as non-volatile secondary storage to satisfy performance and memory demands of data-intensive multimedia applications. In 2.5-3G cell phones [1], flash (mainly NAND) memory stores OS, application programs, and data. During boot time, the OS and application programs are copied from the flash memory to the main memory in a process referred to as “memory shadowing”. To reduce both the program download delay and the DRAM size, recent systems employ “demand paging” which swaps pages of code/data between memories according to processor’s requests. This memory organization leads to a smaller DRAM, less loading time, but requires memory management unit that ensures both high performance and low energy page swapping.

In cell-phones, energy is consumed during active operation as well as on idling, i.e. periods of inactivity. The memory system takes almost 30% of total device power during program execution [2], and more than half of total energy consumed by phone over a day [3] with almost 20% of the energy spent for data retention. Clearly, reducing of energy consumed by memory can significantly extend mobile phone battery life.
The main goal of this paper is to develop a memory management technique capable of lowering energy consumed not only by data accesses, but also by DRAM refresh and data retention.

1.2 Background

(a) Flash Memory: Flash memory is a non-volatile device. It has higher storage density than DRAM. However, its content is not randomly accessed. Usually, a NAND flash contains a fixed number of blocks each of which consists of 16-64 pages. A page normally includes 512B of main data and 16B of spare data. So, a typical 32MByte NAND flash has 4K blocks of 16 pages each [4].

Using flash memory has two main limitations. The first one is a potential durability problem. The flash memory cells have a limited number of write/erase accesses for which performance is guaranteed. After about 10,000 cycles, subsequent accesses begin to take longer. After 100,000 write/erase cycles, flash cells begin fail and become unusable. The second limitation of flash memory is the need to erase data before it can be overwritten. The flash memory manufacturer determines how much memory is erased in a single operation. Usually, erase is performed on a block basis, while read and write are conducted based on a page basis.

There are three important aspects to erasure: flash cleaning, performance and power. When the size of data block is larger than transfer unit, any block data that are still needed must be copied elsewhere. Cleaning flash memory is thus analogous to segment cleaning in a log-structured file system. The cost and frequency of segment cleaning is related in part to cost of erasure and in part to segment size. The larger the segment, the more data will likely to be moved before erasure can take place.

The second aspect to erasure is performance. In NAND flash memory, the time to erase and write a page is 8 times longer than the time of read (see Table 1) [4]. Because the erasure time is independent of the amount of data being erased, the cost of erasure is amortized over large erasure units. To avoid delaying writes for erasure it is important to keep a pool of erased memory available.

The third aspect to erasure is power. Erasing a page in NAND memory consumes as twice as more power than reading the page. Since a write with erase takes almost 10 times more power than a read, page swapping policy must minimize the number of flash memory writes, even though it might incur additional read operations. One such policy is Clean-First-LRU [5], which swaps clean (i.e. unmodified pages first) while keeping dirty pages in the DRAM as long as possible. If there are no clean pages in a predefined time window, a standard LRU is used. To further reduce the number of flash memory writes, the Clean-first-LRU can be combined with selective compression and caching [6]. Because of compression and decompression overhead (248us and 216us respectively), the compression is applied only to those pages, whose compression ratio exceeds a predefined threshold. The other pages are stored in un-compressed form.

(b) DRAM: Dynamic Random Access Memory (DRAM) contains a number of components: cell array, decoders, sense amplifiers and controller. Each DRAM cell consists of one transistor and one capacitor. A write operation of a DRAM cell is performed by charging the capacitor via an on-state cell transistor, while the cell transistor is in an off-state during the charge retention period. DRAM cell retention time is limited by charge leakage from the capacitor through an off-state transistor channel and/or