HIGH PERFORMANCE NAND FLASH MEMORY SYSTEM DESIGN

By

Guiqiang Dong

A Thesis Submitted to the Graduate

Faculty of Rensselaer Polytechnic Institute

in Partial Fulfillment of the

Requirements for the Degree of

DOCTOR OF PHILOSOPHY

Major Subject: ELECTRICAL ENGINEERING

Approved by the
Examining Committee:

Tong Zhang, Thesis Adviser

Gary J. Saulnier, Member

Koushik Kar, Member

Saroj Nayak, Member

Rensselaer Polytechnic Institute
Troy, New York

August 2012

ABSTRACT

The steady bit cost reduction over the past decade enabled NAND flash memory enter increasingly diverse applications, from consumer electronics to personal and enterprise computers. The continuous bit cost reduction of NAND flash memory mainly relies on aggressive technology scaling. Besides technology scaling, multi-level per cell (MLC) technique, i.e., to store more than 1 bit in each memory cell, has been widely used to further improve effective storage density and hence reduce bit cost of NAND flash memory.

The reliability of NAND flash memory relies on several factors, such as program/erase (P/E) cycling count, storage period, etc. For better understanding the behavior of NAND flash memory, it would be of great help if there exists one practical NAND flash memory device model. This thesis presents a NAND flash memory device model, which can emulate various major noise sources in NAND flash memory including the erase/program operations, P/E cycling effect, retention effect, and cell-to-cell interference.

Cell-to-cell interference has been well recognized as a major noise source responsible for raw memory storage reliability degradation. Leveraging the fact that cell-to-cell interference is a deterministic data-dependent process and can be mathematically described with simple formula, this thesis presents two simple yet effective data processing techniques that can well tolerate significant cell-to-cell interference at the system level. These two techniques essentially originate from two signal processing techniques being widely used in digital communication systems to compensate communication channel inter-symbol interference.

Bits stored in each MLC memory cell are subject to different bit error rates. In current practice, bits stored in each cell belong to different pages and all the pages are protected using the same error correction code (ECC) tuned for the worst-case scenario. This results in over-protection for other pages and hence reduced storage capacity. This thesis first demonstrates the significant intra-cell unbalanced bit error characteristics for MLC NAND flash memory, and further develops two techniques that can better address this issue to minimize the overall redundancy overhead and hence improve effective capacity.

As the technology continues to scale down, NAND flash memory has been increasingly relying on more powerful ECCs to ensure the overall data storage integrity. Although advanced ECCs such as low-density parity-check (LDPC) codes can provide significantly stronger error correction capability over BCH codes being used in current practice, their decoding requires soft-decision log-likelihood ratio (LLR) information. This results in a critical issue: Accurate calculation of LLR demands fine-grained memory cell sensing, which nevertheless tends to incur implementation overhead and access latency penalty. Hence, it is critical to minimize the fine-grained memory sensing precision. This thesis proposes a non-uniform memory sensing strategy to reduce the memory sensing precision and thus sensing latency, while still maintaining good error correction performance.

The application of soft decision sensing increases not only the sensing latency, but also the data transfer latency. Powerful LDPC coding solution demands soft-decision memory sensing, which results in longer on-chip memory sensing latency and memory-to-controller data transfer latency. This thesis presents two simple design techniques that can reduce the memory-to-controller data transfer latency. The key is to appropriately apply entropy coding to compress the memory sensing results.

The raw error rate of NAND flash increases with P/E cycling, as well as with storage period. Larger endurance or larger retention will require more powerful ECC with lower coding rate, which decreases the effective storage capacity; with ECC solution fixed, increasing endurance requirement will result in reduction of retention capability. This implies the design trade-off issue among endurance, retention and effective storage capacity. Regardless to specific codes and signal processing schemes, it is of great practical importance to know the theoretical limit on the achievable cell storage efficiency. This thesis develops strategies for estimating the information-theoretical bounds on cell storage efficiency. It can readily reveal the trade-offs among cell storage efficiency, P/E cycling endurance, and retention limit, which can provide important insights for system designers. Motivated by the dynamics of P/E cycling effect revealed by the information-theoretical study, this thesis further develops two memory system design techniques that can improve the average NAND flash memory programming speed and increase the total amount of user data that can be stored in the memory over the memory lifetime.

NAND flash memory has dynamic behavior, i.e., the noise margin decreases with P/E cycling. In the early life time, the noise margin is relatively large. NAND flash memory P/E cycling gradually degrades memory cell storage noise margin, and sufficiently strong fault tolerance must be used to ensure the memory P/E cycling endurance. As a result, the relatively large cell storage noise margin in early memory lifetime is essentially wasted in conventional design practice. To explore the available noise margin in early life time, this thesis advocates a lifetime-aware progressive programming concept to improve single-level per cell (SLC) NAND flash memory write endurance. This thesis proposes to always fully utilize the available cell storage noise margin by adaptively adjusting the number of storage levels per cell, and progressively use these levels to realize multiple 1-bit programming operations between two consecutive erase operations. This simple progressive programming design concept can be realized by two different implementation strategies, which are discussed and compared in details.