NAND Flash Memory With Multiple Page Sizes for High-Performance Storage Devices

Abstract:

In recent years, the demand for NAND flash-based storage devices has rapidly increased because of the popularization of various portable devices. NAND flash memory (NFM) offers many advantages, such as non-volatility, high performance, the small form factor, and low-power consumption, while achieving high chip integration with a specialized architecture for bulk data access. A unit of NFM’s read and program operations, the page, has continuously grown. Although increasing page size reduces costs, it adversely affects performance because of the resultant side effects, such as fragmentation and wasted space, caused by the incongruity of data and page sizes. To address this issue, we propose a multiple-page-size NFM architecture and its management. Our method dramatically improves write performance through adopting multiple page sizes without requiring additional area overhead or manufacturing processes. Based on the experimental results, the proposed NFM improves write latency and NFM lifetime by up to 65% and 62%, respectively, compared with the single-page-size NFM. The proposed architecture of this paper analysis the logic size, area and power consumption using Xilinx 14.2.

Enhancement of the project:

Existing System:

NFM’s higher degree of chip integration compared with other types such as NOR flash memory is achieved by its specialized architecture for bulk data access. In this architecture, the page—the unit for read and program operations—includes numerous memory cells. The unit for erase operations, the block, is composed of tens of pages. Page size has continuously increased since early NFMs, when a single page was smaller than 1 kB.

Increasing the page size enlarges the portion of cell areas in a die and reduces the number of NFM operations that service a given amount of data. Thus, the larger page reduces the cost-per-
bit of NFMs and improves the throughput of NFSDs. However, the larger page cannot guarantee better performance in all cases. In particular, the larger page causes higher fragmentation within NFMs, which causes inefficient utilization of NFM space [called false capacity (FC)] and increases the number of garbage collections (GCs). Moreover, when requests with a small amount of data are given, NFM write performance deteriorates because of frequent read-and-modify procedures. Despite the limitations, increasing the page size is inevitable because of its cost reduction and the performance bottlenecks caused by large requests. We, therefore, propose an NFM architecture that offers the advantages of a small page without sacrificing the benefits of a large one. Although Wang et al. discussed the necessity of such architecture; they did not propose specific and realistic solutions for the implementation.

Disadvantages:

- write latency is high
- The life-time of NFM is less

Proposed System:

Design Concept and Cost

An NFM consists of two parts: 1) the core and 2) the peripheral. The peripheral consists of numerous logic and analog circuits to assist the core, which includes cell arrays, a row/column decoder, and page buffers. A page is defined as the cells activated by one word line within a cell array; therefore, SPNFM, as shown in Fig. 1(a), includes identical pages across multiple symmetric cell arrays. The proposed MPNFM as shown in Fig. 1(b), on the other hand, has asymmetric cell arrays, which result in different page sizes for different cell arrays. In fact, the difference between MPNFM and SPNFM is not significant with respect to the architecture, except for the multiple page sizes. Hence, the design and fabrication of MPNFM do not cause obvious changes in terms of cost and the manufacturing process.
The total area of cell arrays for the two NFMs is identical, because the total number of cells connected to the left and right word lines and the bit line are the same; the only difference between them is the size of the row decoders. The row decoder for a 28 kB page must adopt a larger driver size, but the increased driver size can be compensated by the reduction of the driver size for the 4 kB page. Thus, the total area of row decoders becomes approximately similar. In addition, the cell structure and operation conditions of MPNFM are the same as that of SPNFM; only the number of cells connected to a word line differs.

FLASH TRANSLATION LAYER FOR MPNFM

To support differential pages within MPNFM, we exploit existing FTL with minimal modification. Fig. 2 shows the overall FTL diagram for MPNFM. It includes two independent page-level FTLs that can be employed to leverage an attractive features in modern NFSDs.
Address Translation

The request distributor and region bit table are modules relating to address translation. To classify write requests, the request distributor forwards write requests smaller than \( \theta \) to the small-page FTL, where \( \theta \) is the size of one small page. By forwarding small writes to small pages, large-page fragmentation incurred by small writes can be reduced.

Garbage Collection

GC of each page type is separately performed, because there are two independent FTLs for each page type. However, owing to the limited capacity of small pages and migration overhead, it is better for data to be eventually moved to large pages; when the data are moved from large to small pages, more program operations must be executed.

The GC modification is intended to maximize the effects of MPNFM. That is, the modified GC reduces fragmentation by transforming small write requests into large ones that are compatible with large pages. Moreover, the modification on the FTL does not require significant design efforts, because only reads of adjacent data are to be added and small-page programs are to be substituted for large ones.
Advantages:

- Improves write latency
- The life-time of NFM is increased

Software implementation:

- Modelsim
- Xilinx ISE