KIOXIA

Technical Brief

Managed Flash Background Operations Series

Part 5: Understanding Logical Block Address to Physical Block Address Translation in NAND Flash Memory

NAND flash memory enables content mobility that is relied upon daily and is considered an effective way of scaling data storage capacity and performance, as long as the proper management is applied. Logical block address to physical block address translation provides a fundamental management operation that enables efficient data storage in e-MMC[1] and UFS[2] devices and is the focus of this technical brief. This brief also represents the final installment in the Managed Flash Background Operations series that has covered error correction coding, bad block management, wear leveling and garbage collection.

Logical Block Address to Physical Block Address Translation Overview

Logical block addresses (LBAs) and physical block addresses (PBAs) in storage systems are used for data storage and retrieval in storage devices (such as hard drives, SSDs and managed flash[3] devices). Logical block addresses provide an abstraction for the software to interact with storage devices, while physical block addresses represent the actual hardware locations.

Hard drives have typically been the traditional way to store data and use fixed data sizes called sectors at 512 bytes in size. Sectors are located via logical block addresses through a linear addressing methodology that use an integer represented by a hexadecimal number. For example, the first sector is LBA0x0, the second is LBA0x1, and so on. As SSDs and managed flash devices replaced hard drive storage, they needed to support the same traditional addressing schemes for compatibility. NAND flash memory organizes data by rows and columns comprised of blocks, and within a block are several pages. The blocks and pages access the data written to NAND flash memory.

Accessing data from a logical block address to a physical block address requires the creation of an entry or record that assigns a logical block address to the NAND block address (Figure 1).

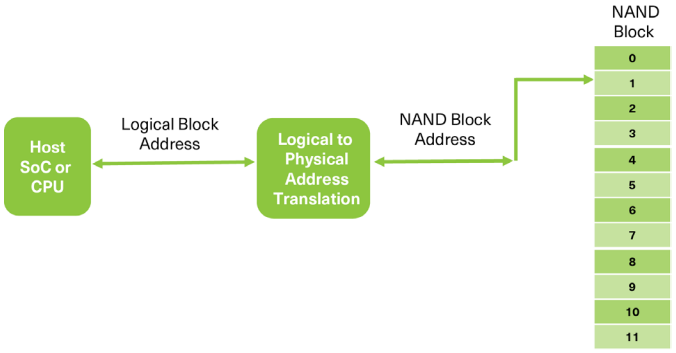

Figure 1: A created entry assigns a logical block address to a physical block address

To keep track of all logical block address to physical block address assignments, a logical to physical (L2P) mapping table is used. The table (Figure 2) is stored in the system data of e-MMC or UFS devices and updated whenever a write operation occurs on a logical block address.

	Physical Block Address	
Logical Block Address	Block Number	Page Number
0	0	0
1	0	1

Figure 2: Example of logical block address to physical block address mapping table

How Logical Block Address to Physical Block Address Translation Works

Process 1: Logical Block Addressing:

The host system accesses the e-MMC/UFS device using logical block addresses to locate sectors. These addresses are virtual representations of the data stored on the e-MMC/UFS device and are managed by the file system.

Process 2: Metadata Mapping:

The internal controller of an e-MMC/UFS device utilizes mapping metadata, such as translation tables or mapping tables, to convert a logical block address into physical block addresses. The metadata structures store the necessary information to identify the physical location of data within NAND flash memory (Figure 3).

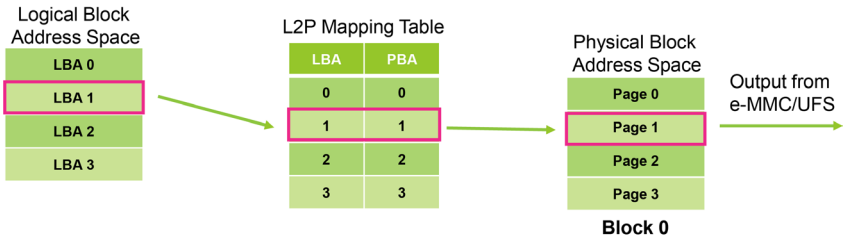

Figure 3: Logical block addressing and metadata mapping within NAND flash memory

Process 3: Address Translation:

Upon a data request, the e-MMC/UFS controller references the mapped metadata to determine the corresponding physical block addresses translated from logical block addresses. This translation process searches mapping tables and performs necessary calculations to determine correct data locations within NAND flash memory (Figure 3).

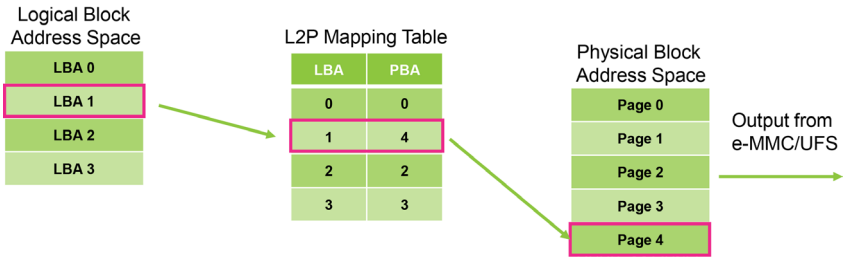

Figure 4: Logical block addresses seen by the host do not change even after the physical block address space changed

The e-MMC/UFS device undergoes background operations (i.e., garbage collection and wear leveling), moving data to different physical locations in the NAND memory array, versus from where the data was originally written. Though the physical block addresses change, the logical block addresses seen by the host do not (Figure 4). In this example, LBA1 was originally mapped to PBA1, but over time, the data moved to PBA4, yet LBA1 remained visible to the host.

Process 4: Data Access:

Once a physical block address is determined, the e-MMC/UFS controller accesses the appropriate NAND flash memory cells to read or write the data. This operation involves activating the desired memory cells, applying voltage levels, and managing data movements within NAND flash memory to perform read or write operations.

Effects of Logical Block Address to Physical Block Address Translation in e-MMC/UFS Devices

Logical block address to physical block address translation is a fundamental operation in e-MMC/UFS devices that enables efficient data storage and is necessary due to the inherent characteristics of the NAND flash memory architecture. For example, new data written to NAND flash memory requires erasing the old data first. By utilizing logical block address to physical block address translation, e-MMC/UFS devices can simply change the physical block address without having to erase the old data, which helps to improve NAND flash memory endurance. This type of address translation also helps manage NAND bad block counts and wear leveling.

When implementing logical block address to physical block address translation in e-MMC/UFS devices, some design challenges may become apparent, and include:

Performance Impact:

Logical block address to physical block address translation introduces overhead and latency that can negatively affect the performance of e-MMC/UFS devices. This occurs from the constant updating of metadata and frequent data movements that can result in increased access times and decreased overall performance. The additional operations required for this address translation can slow down write and read operations that can lead to reduced responsiveness and low throughput.

Performance Optimization:

Logical block address to physical block address translation plays a critical role in optimizing e-MMC/UFS device performance. By utilizing efficient translation algorithms and caching mechanisms, latency and access times reduce resulting in improved read and write speeds. One example is that the L2P table does not specify old data for deletion every time, which would reduce latency and result in fast performance during data erasures.

Data Integrity:

Accurate logical block address to physical block address translation is crucial for maintaining data integrity. Errors that occur in the translation process can lead to corrupted data or data loss. It is important to store the metadata and L2P table in reliable NAND flash memory, such as in single-level cell (SLC) mode.

Write Amplification:

Logical block address to physical block address translation can contribute to write amplification, a phenomenon where the number of physical writes exceeds the number of logical writes (Figure 5). This occurs because the translation process often involves data movements and updates to metadata that lead to additional write operations. Higher write amplification reduces the lifespan of the e-MMC/UFS devices, as each NAND flash memory cell has specified endurance relating to program and erase cycles.

Write Amplification Factor	
WAF =	$\frac{\textit{Data written to NAND flash memory}}{\textit{Data written by the host}}$

Figure 5: WAF formula

Summary

Logical block address to physical block address translation is a vital process that delivers significant benefits to e-MMC/UFS devices. By converting logical block addresses to physical block addresses within NAND flash memory, these managed flash devices can optimize performance, effectively manage data and extend flash memory lifecycles. As logical block address to physical block address translation tables update over the lifetime of e-MMC/UFS devices, the performance impacts can be evident. With applications and usage of e-MMC/UFS devices continuing to evolve, the ability to address the challenges associated with logical block address to physical block address translation will remain critical in developing products that further enhance the reliability and performance of these storage devices.

General information for KIOXIA memory products is available here.

FOOTNOTES:

[1] Embedded MultiMediaCard (e-MMC) is a specification developed by JEDEC® for mobile applications. The current release is v5.1, published in February 2015.

[2] Universal Flash Storage (UFS) devices are based on the UFS specification, of which, the v4.0 specification is the current release issued by JEDEC and published in August 2022.

[3] A managed flash device combines raw NAND flash memory and an intelligent controller in one integrated package, enabling internal memory management.

TRADEMARKS:

JEDEC is a registered trademark of JEDEC Solid State Technology Association. All other company names, product names and service names may be trademarks or registered trademarks of their respective companies.

DISCLAIMERS:

 Information in this tech brief, including product specifications, tested content, and assessments are current and believed to be accurate as of the publication date of the document, but is subject to change without prior notice. Technical and application information contained here is subject to the most recent applicable KIOXIA product specifications. Images within are for illustration purposes only.