

Accurately judging endurance for solid-state storage

DETERMINING THE APPROPRIATE SOLID-STATE-STORAGE APPROACH FOR OEM APPLICATIONS CAN BE DIFFICULT, ESPECIALLY WHEN SUPPLIERS OFTEN RELY ON VARYING METHODS TO MEASURE THE PERFORMANCE OF THEIR PRODUCTS. IT IS NO WONDER THE SELECTION PROCESS IS DAUNTING.

Designers require a common measure for accurately evaluating storage options and their reliability. Endurance can provide such a measure, and understanding how to evaluate it correctly can help designers effectively select the right approach. “Endurance” refers to the number of write/erase cycles that a product can perform before it wears out. However, endurance is more than just a function of the storage media you are evaluating. The media and the associated controller technology combine to define a product’s endurance. For example, magnetic media is an order of magnitude less reliable than NAND flash, yet the controller technology that rotating hard drives employ can compensate reasonably well for this deficiency.

Vendors specify write/erase-cycle endurance for solid-state storage in many ways. Some specify the endurance at the physical-block level, others specify it at the logical-block level, and still others specify it at the card or drive level. Because endurance also relates to data retention, you can calculate endurance at a higher level if the data-retention requirement is lower. For these reasons, it is often difficult to make an “apples-to-apples” comparison of write/erase endurance by solely relying on these numbers in a data sheet.

To most appropriately measure endurance, you must break specification data into three criteria: storage-media type, wear-leveling algorithms, and error-correction capabilities. Additional factors that affect endurance include the amount of spare sectors available and whether the storage device performs a write using a file system or direct logical-block addressing. Although these issues can contribute to the overall endurance calculation, their effects on the resulting number are much lower than those of the first three criteria.

STORAGE-MEDIA TYPE

The first of three critical steps in selecting the right solid-state-storage media is to compare media types, including both NOR and NAND flash. Current NOR-technology implementations generally find use only in cell-phone and other chip-on-board applications. For these applications, NOR provides execute-in-

place, boot, and data-storage functions in one chip. The economies of scale and component densities of NAND make it a better approach than NOR for nonvolatile, solid-state-storage systems.

The two dominant NAND technologies are SLC (single-level cell), or binary, and MLC (multilevel cell). SLC technology stores one bit per cell, and MLC stores two bits. SLC NAND is generally specified at 100,000 write/erase cycles per block with 1 bit of ECC (error-correction code), whereas MLC is specified at 10,000 cycles with ECC. Although a data sheet for an MLC device may not specify the level of required ECC, MLC manufacturers recommend 4-bit ECC when using this technology. So, when using the same controller, a storage system using SLC has an endurance value roughly an order of magnitude higher than that of a similar MLC-based product.

WEAR-LEVELING ALGORITHMS

A wear-leveling algorithm allows storage media to have evenly distributed data writes and allows the controller in the storage media to remap logical-block addresses to different physical-block addresses in the solid-state memory array. The frequency of this remap, the algorithm to find the least worn area on which to write, and any data-swapping capabilities are generally proprietary intellectual property of the controller vendor. An integrated controller in the solid-state-storage system, which is independent of the host system, performs wear leveling. The host system performs its reads and writes only to logical-block addresses. So, as far as the host is concerned, the data never moves.

Three scenarios to consider when discussing wear leveling are no wear leveling, dynamic wear leveling, and static wear leveling. Flash cards for the retail consumer market and those for solid-state-storage systems for industrial applications differ greatly in endurance specs. A flash card that uses no wear leveling stops operating once the physical blocks wear out and all the spare blocks are exhausted, regardless of how much storage space remains unused. Early flash cards used no wear leveling and thus failed in write-intensive applications. For this reason, vendors do not recommend flash without wear leveling for use in enter-

prise-system OEM applications requiring industrial-grade, solid-state storage.

Dynamic wear leveling wears levels only over dynamic, or “free,” areas. Systems using dynamic wear leveling do not touch static data. In a system using 75% of storage for static data, only 25% is available for wear leveling. So, you calculate the endurance of this approach as 25 times greater than a card with no wear leveling, but only one-fourth that of an approach that wear-levels over the entire storage space.

A static-wear-leveling algorithm evenly distributes data over an entire system and searches for the least used physical blocks. Once it finds them, it writes the data to those locations. If blocks are empty, the write occurs normally. If they contain static data, it moves that data to a more heavily used location before it moves the newly written data. You calculate the endurance of a storage system using static wear leveling to be 100 times better than an approach without wear leveling. A system with 75% of the storage containing static data is four times better than a card that implements dynamic wear leveling. Static wear leveling provides better endurance because it writes data to all blocks of the storage system.

ERROR-CORRECTION CAPABILITIES

Part of the solid-state memory-component specification relates to error correction, an equally important consideration for measuring endurance. For example, SLC NAND components perform 100,000 write/erase cycles with 1-bit ECC. It stands to reason that the specification would increase with a better error-correction algorithm. Most flash cards employ ECC algorithms with 2- to 4-bit correction, and a few industrial-grade, solid-state-storage systems employ as much as 6-bit correction. The terms “2-bit,” “4-bit,” and “6-bit” correction can be slightly confusing, because they refer to bytes in a 512-byte sector. For example, 6-bit correction defines the ability to correct as many as 6 bytes of data in a 512-byte sector. Because a byte equals 8 bits, a system with 6-bit ECC can correct 48 bits, as long as those bits stay within 6 bytes in the sector. The same definition holds true for 2- and 4-bit ECC.

The relationship between the number of bytes per sector that a controller can correct is not directly proportional to the overall endurance of a storage system, because the bit-error rate of NAND flash is not linear. Actually, 6-bit error correction should be much more than three times better than 2-bit ECC, because the probability of a 3-bit error is significantly greater than the probability of a 7-bit error.

THE ENDURANCE CALCULATOR

You can use the following equations to get an idea of how long a solid-state-storage system will last in an application. These calculations are valid only for products that use either dynamic or static wear leveling. You use solid-state-memory-component specifications for products that use no wear leveling. To calculate the expected life in years a product will last, use the following equation:

$$\text{YEARS} = \frac{(\alpha - \beta) \times \lambda \times (1 - \phi)}{(\omega \times \xi) \times k}$$

MORE AT EDN.COM



+ We encourage your comments!
Go to www.edn.com/ms4152 and click on Feedback Loop to post a comment on this article.

where α is the capacity in megabytes, β is the amount of static data in megabytes, λ is the block-level endurance specification, ϕ is the safety margin, ω is the file size in megabytes, ξ is the number of writes of file size ω per minute, and k is the number of minutes per year. For the α calculation, when

converting from gigabytes to megabytes, megabytes equal the gigabytes times 1024. Note that you need to calculate β , the static data, only for cards that use a dynamic-wear-leveling algorithm. This value should be zero for static wear leveling.

When calculating λ , the block-level endurance, the point is to determine the true endurance of a product—not the length of time a vendor claims it will last. Rather than use the specification on the component vendor’s data sheet, obtain block-level-endurance specifications directly from the storage provider and ensure that you have the endurance specification at the block level and not for the card or system. Many vendors specify endurance at the card level, so be careful to use the proper value for this variable. Base the user-defined ϕ , or safety margin, on the confidence of the data-transaction numbers that follow. For ω , the file size, when converting from megabytes to kilobytes, megabytes equal kilobytes divided by 1024. When calculating k , the number of minutes per year, the formula is: 60 minutes/hour \times 24 hours/day \times 365 days/year = 525,600.

To calculate the number of data transactions, use the following equation:

$$\text{TRANSACTIONS} = \frac{(\alpha - \beta) \times \lambda \times (1 - \phi)}{\omega}$$

where all of the symbols carry the same meaning as they do in the expected-life calculation.

Designers will continue to face challenges in their selections as solid-state-storage manufacturers and flash-card providers vie for their attention. As part of their evaluation, designers will note differing requirements for industrial-OEM applications and for consumer applications, so it is important to distinguish between a flash-card supplier and a developer of solid-state storage.

Although it is valuable to evaluate solid-state-storage systems based on the type of media, it should never be your only consideration. The controller’s ability to compensate for the media is an even more significant issue. Additionally, the use of wear leveling and ECC can dramatically affect the reliability and enhance the usable life of the solid-state-storage system in an application. Taking all of these parameters into consideration should help designers more easily wade through the selection process and ultimately select the best approach for their application. **EDN**

AUTHOR’S BIOGRAPHY

Gary Drossel is director of product marketing at SiliconSystems (Aliso Viejo, CA), which he joined in 2004. He is responsible for managing marketing and business development for the company’s product line. A 15-year industry veteran, Drossel has also played a leading role in developing the company’s marketing strategy, including product roll-outs and customer introductions. He received a bachelor’s degree in electrical and computer engineering from the University of Wisconsin (Madison).